

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 637 490**

51 Int. Cl.:

H04N 19/13 (2014.01)

H03M 7/40 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **15.04.2013 PCT/US2013/036646**

87 Fecha y número de publicación internacional: **24.10.2013 WO13158566**

96 Fecha de presentación y número de la solicitud europea: **15.04.2013 E 13718475 (0)**

97 Fecha y número de publicación de la concesión europea: **17.05.2017 EP 2839645**

54 Título: **Grupos de coeficientes y codificación de coeficientes para barridos de coeficientes**

30 Prioridad:

16.04.2012 US 201261625039 P

02.07.2012 US 201261667382 P

15.03.2013 US 201313834006

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

13.10.2017

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

5775 Morehouse Drive

San Diego, CA 92121, US

72 Inventor/es:

SOLE ROJALS, JOEL;

JOSHI, RAJAN LAXMAN y

KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 637 490 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

Grupos de coeficientes y codificación de coeficientes para barridos de coeficientes

5 CAMPO TÉCNICO

La presente divulgación se refiere a la codificación de vídeo y, de forma más particular, a técnicas para codificar elementos sintácticos en la codificación de vídeo.

10 ANTECEDENTES

Las capacidades de vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de radiodifusión digital directa, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de emisión de vídeo en continuo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo definidas de acuerdo con las normas de codificación de vídeo. Los dispositivos de vídeo digital pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de compresión de vídeo. Entre las normas de codificación de vídeo se incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluidas sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC). Además, la codificación de vídeo de alta eficiencia (HEVC) es una norma de codificación de vídeo que está en proceso de elaboración por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC.

Las técnicas de compresión de vídeo llevan a cabo la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo de un segmento intracodificado (I) de una imagen se codifican mediante predicción espacial con respecto a unas muestras de referencia de unos bloques contiguos de la misma imagen. Los bloques de vídeo de un segmento intercodificado (P o B) de una imagen pueden utilizar predicción espacial con respecto a unas muestras de referencia de unos bloques vecinos de la misma imagen, o predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

La predicción espacial o temporal dan como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de codificación intra y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de transformada, dando como resultado unos coeficientes de transformada residual, que posteriormente se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden leerse mediante barrido con el fin de generar un vector unidimensional de coeficientes de transformada, y puede aplicarse codificación de entropía para lograr aún más compresión.

El documento US 2012/0082233 A1 de Sze *et al.* divulga un procedimiento y un aparato para codificar un código de bits mediante simplificación de dependencia de contexto para reducir los barridos dependientes. El procedimiento incluye recuperar al menos una matriz de 2 dimensiones de coeficientes de transformada, transformar la al menos una matriz de 2 dimensiones del mapa de significancia de los coeficientes de transformada en un barrido de coeficientes de 1 dimensión y determinar al menos uno de entre una dirección de barrido y un tipo de unidad de codificación y un tipo de segmento asignados al coeficiente de transformada, seleccionar vecinos basándose en al menos uno de entre una dirección de barrido y un tipo de unidad de codificación y un tipo de segmento, calcular un índice de contexto basándose en los valores de los vecinos seleccionados para selección de contexto y realizar una codificación aritmética para generar un bit codificado mediante el índice de contexto y la binarización calculados.

60 RESUMEN

En general, esta divulgación describe técnicas para codificar y decodificar datos que representan la significancia de los coeficientes de transformada, tales como los indicadores de coeficiente de significancia y los indicadores de grupos de coeficientes, en la codificación de coeficientes de transformada. Se describen varias técnicas para determinar un contexto que se va a utilizar para realizar la codificación CABAC (codificación aritmética binaria

adaptativa según el contexto) de los elementos sintácticos asociados con los coeficientes de transformada.

Por ejemplo, en algunas técnicas, un codificador de vídeo y un decodificador de vídeo seleccionan un patrón de contextos de una misma pluralidad de patrones de contextos para un tipo de barrido de un barrido horizontal, un barrido vertical y un barrido diagonal. En otras palabras, independientemente de si un subbloque se lee mediante barrido vertical, horizontal o diagonal, el codificador de vídeo y el decodificador de vídeo pueden seleccionar el patrón de contextos a partir de los mismos patrones de contextos para los tres tipos de barrido. El codificador de vídeo y el decodificador de vídeo utilizan los contextos del patrón de contextos seleccionado para realizar la codificación CABAC o la decodificación CABAC, respectivamente, de los elementos sintácticos de significancia de un bloque de transformada.

En otro ejemplo, en algunas técnicas, un codificador de vídeo y un decodificador de vídeo pueden seleccionar un patrón de contextos. En algunos ejemplos, el codificador de vídeo y el decodificador de vídeo utilizan el patrón de contextos seleccionado para dos o más tipos de barrido del subbloque. Por ejemplo, si el subbloque se lee mediante barrido horizontal, el codificador de vídeo y el decodificador de vídeo utilizan el patrón de contextos seleccionado y si el subbloque se lee mediante barrido vertical, el codificador de vídeo y el decodificador de vídeo utilizan el patrón de contextos seleccionado.

En un ejemplo, la divulgación describe un procedimiento para decodificar datos de vídeo. El procedimiento comprende recibir, en un flujo de bits, elementos sintácticos de significancia para coeficientes de transformada de un subbloque actual de un bloque, seleccionar un patrón de contextos a partir de una pluralidad de patrones de contextos bidimensionales para una pluralidad de tipos de barrido para los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual, en el que la pluralidad de patrones de contextos bidimensionales es la misma para cada uno de la pluralidad de tipos de barrido, y en el que cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más subbloques vecinos, asignar contextos a cada uno de los elementos sintácticos de significancia de los coeficientes de transformada basándose en el patrón de contextos seleccionado y decodificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados.

En otro ejemplo, la divulgación describe un dispositivo para decodificar datos de vídeo, comprendiendo el dispositivo un decodificador de vídeo configurado para recibir, en un flujo de bits, elementos sintácticos de significancia para coeficientes de transformada de un subbloque actual de un bloque, seleccionar un patrón de contextos de una pluralidad de patrones de contextos bidimensionales para una pluralidad de tipos de barrido para los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual, en el que la pluralidad de patrones de contextos bidimensionales es la misma para cada uno de la pluralidad de tipos de barrido, y en el que cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más subbloques vecinos, asignar contextos a cada uno de los elementos sintácticos de significancia de los coeficientes de transformada basándose en el patrón de contextos seleccionado, y decodificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados.

En otro ejemplo, la divulgación describe unos medios de almacenamiento legibles por ordenador que tienen instrucciones almacenadas que, al ejecutarse, hacen que uno o más procesadores de un dispositivo para decodificar datos de vídeo reciban, en un flujo de bits, elementos sintácticos de significancia para coeficientes de transformada de un subbloque de un bloque, seleccionen un patrón de contextos de una pluralidad de patrones de contextos bidimensionales para una pluralidad de tipos de barrido para los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual, en el que la pluralidad de patrones de contextos bidimensionales es la misma para cada uno de la pluralidad de tipos de barrido, y en el que cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más subbloques vecinos, asignen contextos a cada uno de los elementos sintácticos de significancia de los coeficientes de transformada basándose en el patrón de contextos seleccionado y decodifiquen mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados.

En otro ejemplo, la divulgación describe un procedimiento para codificar datos de vídeo. El procedimiento comprende generar elementos sintácticos de significancia para coeficientes de transformada de un subbloque actual de un bloque, seleccionar un patrón de contextos de una pluralidad de patrones de contextos bidimensionales para una pluralidad de tipos de barrido para los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual, en el que la pluralidad de patrones de contextos bidimensionales es la misma para cada uno de la pluralidad de tipos de barrido, y en el que cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más subbloques vecinos, asignar contextos a cada uno de los elementos sintácticos de significancia de los coeficientes de transformada basándose en el patrón de contextos seleccionado, codificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados, y facilitar los elementos sintácticos de significancia

codificados.

5 En otro ejemplo, la divulgación describe un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo un codificador de vídeo configurado para generar elementos sintácticos de significancia para coeficientes de transformada de un subbloque actual de un bloque, seleccionar un patrón de contextos a partir de una pluralidad de patrones de contextos para una pluralidad de tipos de barrido para los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual, en el que la pluralidad de patrones de contextos bidimensionales es la misma para cada uno de la pluralidad de tipos de barrido, y en el que cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más subbloques, asignar contextos a cada uno de los elementos sintácticos de significancia de los coeficientes de transformada basándose en el patrón de contextos seleccionado, codificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados, y facilitar los elementos sintácticos de significancia codificados.

15 En otro ejemplo, la divulgación describe un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo medios para generar elementos sintácticos de significancia para coeficientes de transformada de un subbloque actual de un bloque, medios para seleccionar un patrón de contextos a partir de una pluralidad de patrones de contextos bidimensionales para una pluralidad de tipos de barrido para los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual, en el que la pluralidad de patrones de contextos bidimensionales es la misma para cada uno de la pluralidad de tipos de barrido y en el que cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más subbloques vecinos, medios para asignar contextos a cada uno de los elementos sintácticos de significancia de los coeficientes de transformada basándose en el patrón de contextos seleccionado, medios para codificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados, y medios para facilitar los elementos sintácticos de significancia codificados.

30 En otro ejemplo, la divulgación describe un procedimiento para decodificar datos de vídeo. El procedimiento comprende recibir, en un flujo de bits, elementos sintácticos de significancia de coeficientes de transformada para un subbloque actual de un bloque, seleccionar un patrón de contextos, en el que el patrón de contextos identifica contextos para dos o más tipos de barrido del subbloque actual, asignar contextos a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en el patrón de contextos seleccionado y decodificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados.

40 En otro ejemplo, la divulgación describe un dispositivo para decodificar datos de vídeo, comprendiendo el dispositivo un decodificador de vídeo configurado para recibir, en un flujo de bits, elementos sintácticos de significancia de coeficientes de transformada para un subbloque actual de un bloque, seleccionar un patrón de contextos, en el que el patrón de contextos identifica contextos para dos o más tipos de barrido del subbloque actual, asignar contextos a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en el patrón de contextos seleccionado y decodificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados.

50 En otro ejemplo, la divulgación describe unos medios de almacenamiento legibles por ordenador que tienen instrucciones almacenadas que, al ejecutarse, hacen que uno o más procesadores de un dispositivo para decodificar datos de vídeo reciban, en un flujo de bits, elementos sintácticos de significancia de coeficientes de transformada para un subbloque de un bloque, seleccionen un patrón de contextos, en el que el patrón de contextos identifica contextos para dos o más tipos de barrido del subbloque actual, asignen contextos a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en el patrón de contextos seleccionado, y decodifiquen mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados.

60 En otro ejemplo, la divulgación describe un procedimiento para codificar datos de vídeo. El procedimiento comprende generar elementos sintácticos de significancia de coeficientes de transformada para un subbloque actual de un bloque, seleccionar un patrón de contextos, en el que el patrón de contextos identifica contextos para dos o más tipos de barrido del subbloque actual, asignar contextos a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en el patrón de contextos seleccionado, codificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados, y facilitar los elementos sintácticos de significancia codificados.

65 En otro ejemplo, la divulgación describe un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo

un codificador de vídeo configurado para generar elementos sintácticos de significancia de coeficientes de transformada para un subbloque actual de un bloque, seleccionar un patrón de contextos, en el que el patrón de contextos identifica contextos para dos o más tipos de barrido del subbloque actual, asignar contextos a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en el patrón de contextos seleccionado, codificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados, y facilitar los elementos sintácticos de significancia codificados.

En otro ejemplo, la divulgación describe un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo medios para generar elementos sintácticos de significancia de coeficientes de transformada para un subbloque actual de un bloque, medios para seleccionar un patrón de contextos, en el que el patrón de contextos identifica contextos para dos o más tipos de barrido del subbloque actual, medios para asignar contextos a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en el patrón de contextos seleccionado, medios para codificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual basándose en los contextos asignados, y medios para facilitar los elementos sintácticos de significancia codificados.

Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La FIG. 1 es un diagrama conceptual que ilustra un ejemplo de grupos de coeficientes y barridos para un bloque de vídeo.

Las FIGS. 2A-2B son diagramas conceptuales que ilustran ejemplos de grupos de coeficientes y barridos para un bloque de vídeo.

Las FIGS. 3A-3B son diagramas conceptuales que ilustran ejemplos de grupos de coeficientes y barridos para un bloque de vídeo.

La FIG. 4 es un diagrama conceptual que ilustra una relación entre los coeficientes de transformada de un bloque de vídeo y un mapa de significancia asociado con el bloque de vídeo.

Las FIGS. 5A-5D son diagramas conceptuales que ilustran un ejemplo de patrones para la asignación de contexto para los coeficientes de un subbloque.

La FIG. 6 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede utilizar las técnicas de predicción inter descritas en esta divulgación.

La FIG. 7 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas de predicción inter descritas en esta divulgación.

La FIG. 8 es un diagrama de bloques que ilustra un ejemplo de codificador de entropía que puede implementar técnicas para realizar la codificación de entropía de unos elementos sintácticos predictivos de acuerdo con esta divulgación.

Las FIGS. 9A-9D son diagramas conceptuales que ilustran ejemplos de patrones para la asignación de contexto para los coeficientes de un subbloque.

La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de patrón para la asignación de contexto para los coeficientes de un subbloque.

La FIG. 11 es un diagrama de flujo que ilustra la codificación de elementos sintácticos predictivos de acuerdo con las técnicas de esta divulgación.

FIG. 12 es un diagrama de flujo que ilustra la codificación de elementos sintácticos predictivos de acuerdo con las técnicas de esta divulgación.

La FIG. 13 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo que puede implementar las técnicas de predicción inter descritas en esta divulgación.

La FIG. 14 es un diagrama de bloques que ilustra un ejemplo de codificador de entropía que puede implementar técnicas para decodificar elementos sintácticos predictivos de acuerdo con esta divulgación.

La FIG. 15 es un diagrama de flujo que ilustra la decodificación de elementos sintácticos predictivos de acuerdo con las técnicas de esta divulgación.

La FIG. 16 es un diagrama de flujo que ilustra la decodificación de elementos sintácticos predictivos de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

Un codificador de vídeo diseñado de acuerdo con algunos ejemplos, tal como el propuesto en el borrador de trabajo 7 (WD7) de la norma de codificación de vídeo de alta eficiencia (HEVC), denominado en el presente documento HEVC WD7 y disponible en http://phenix.itsudparis.eu/jct/doc_end_user/documents/9_Geneva/wgll/JCTVC-I1003-v5.zip, puede requerir un acceso de datos para los grupos de coeficientes no cuadrados diferente al de un grupo de coeficientes de subbloque 4x4. Esto puede imponer una complejidad de hardware y software adicional durante la implementación del codificador de vídeo. La complejidad adicional de hardware y software puede reducirse si se eliminan los grupos de coeficientes no cuadrados y se leen mediante barrido los coeficientes de subbloque 4x4 de

acuerdo con uno de un tipo de barrido diagonal, vertical u horizontal. Sin embargo, esta modificación puede reducir la eficiencia de codificación cuando las obtenciones de contexto definidas de acuerdo con HEVC WD7 se utilizan para asignar contextos a elementos sintácticos que indican si un subbloque incluye coeficientes significativos. Por lo tanto, esta divulgación describe una técnica para asignar contextos para elementos sintácticos que indican si un subbloque incluye coeficientes significativos que pueden proporcionar una eficiencia de codificación mejorada.

Por ejemplo, en algunas de estas otras técnicas (es decir, aquellas que no están necesariamente de acuerdo con las técnicas descritas en esta divulgación), se utilizan patrones de contextos para un subconjunto de los tamaños posibles para unidades de transformada (TU o bloques de transformada) para la codificación de elementos sintácticos que indican la significancia de los coeficientes de transformada de las unidades de transformada (conocidos como elementos sintácticos de significancia). Además, estas otras técnicas utilizaban los patrones de contextos para tipos de barrido limitados. En consecuencia, se desperdician recursos computacionales al tener que determinar el tamaño de la TU para poder determinar si se pueden utilizar patrones de contextos.

En las técnicas descritas en esta divulgación, se utilizan los mismos patrones de contextos para una pluralidad de tipos de barrido (por ejemplo, un barrido horizontal, un barrido vertical y un barrido diagonal) para una variedad de TU de diferentes tamaños. Por ejemplo, un codificador de vídeo o un decodificador de vídeo pueden seleccionar un patrón de contextos a partir de la misma pluralidad de patrones de contextos para un subbloque 4x4 de una TU 8x8, independientemente del tipo de barrido para el subbloque 4x4 (por ejemplo, independientemente de si el subbloque 4x4 se lee mediante barrido horizontal, se lee mediante barrido vertical o se lee mediante barrido diagonal). Como se describe con más detalle, cada uno de la pluralidad de patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada significativo (por ejemplo, algún coeficiente de transformada distinto de cero) en uno o más subbloques vecinos. Como también se describe en mayor detalle, esta divulgación describe características de la pluralidad de patrones de contextos a partir de los cuales el codificador de vídeo o decodificador de vídeo selecciona el patrón de contexto. De esta manera, pueden obtenerse eficiencias computacionales, puesto que el codificador de vídeo y el decodificador de vídeo pueden utilizar los mismos patrones de contextos para determinar contextos para unos elementos sintácticos de significancia para una pluralidad de tipos de barrido (por ejemplo, un barrido horizontal, un barrido vertical y un barrido diagonal) para los elementos sintácticos de significancia del subbloque, incluidos un subbloque 4x4 de un bloque 8x8 (es decir, una TU 8x8).

En los ejemplos anteriores, los patrones de contextos pueden ser patrones de contextos bidimensionales. Sin embargo, los aspectos de esta divulgación no están limitados a lo anterior. En algunos ejemplos, el codificador de vídeo y el decodificador de vídeo seleccionan un patrón de contextos (por ejemplo, uno que se almacena como un patrón de contextos unidimensional). Por ejemplo, algunos patrones de contextos pueden definirse como patrones de contextos bidimensionales. Tal vez sería posible precalcular un patrón de contextos unidimensional a partir del patrón de contextos bidimensional. El precálculo puede agilizar el proceso de codificación y decodificación. Por ejemplo, los coeficientes de transformada de un bloque bidimensional pueden convertirse en un bloque unidimensional. Con los patrones unidimensionales precalculados, pueden obtenerse eficiencias de codificación y decodificación si se utiliza un patrón unidimensional para codificar o decodificar los elementos sintácticos de significancia, debido a que los coeficientes de transformada se convierten en un bloque unidimensional, en comparación con la utilización del patrón de contextos bidimensional en un bloque unidimensional. Debe entenderse que no se requiere precalcular el patrón de contextos unidimensional a partir del patrón de contextos bidimensional en cada ejemplo, y que eso no debe considerarse como una forma limitada de determinar el patrón de contextos unidimensional.

Puede haber varias maneras en las que se puede calcular el patrón de contextos unidimensional. En un ejemplo se lee un patrón de contextos bidimensional mediante barrido diagonal, barrido horizontal y barrido vertical para generar tres patrones de contextos unidimensionales (por ejemplo, uno para cada tipo de barrido). En las técnicas descritas en esta divulgación, los patrones de contextos bidimensionales pueden comprender características que reducen el número total de patrones de contextos unidimensionales que se generan.

Por ejemplo, si hay cuatro patrones de contextos bidimensionales cada uno de los cuales se lee mediante barrido horizontal, vertical y diagonal para generar patrones de contextos unidimensionales, entonces habría un total de 12 patrones de contextos unidimensionales. En algunos ejemplos, los cuatro patrones de contextos bidimensionales pueden incluir contextos dispuestos de tal manera que dos barridos diferentes dan como resultado el mismo patrón de contextos unidimensional.

Por ejemplo, uno de los patrones de contextos bidimensionales puede incluir contextos que si se leen mediante barrido horizontal o vertical dan como resultado el mismo patrón de contextos unidimensional. En otro ejemplo, uno de los patrones de contextos bidimensionales puede incluir contextos que cuando se leen mediante barrido horizontal dan como resultado un patrón de contextos unidimensional que es el mismo vector unidimensional que resultaría si se leyera mediante barrido vertical otro de los patrones de contextos bidimensionales. En otro ejemplo, uno de los patrones de contextos bidimensionales puede incluir contextos que cuando se leen mediante barrido horizontal, vertical y diagonal dan como resultado el mismo patrón de contextos unidimensional.

De esta manera, puede haber superposición en los patrones de contextos unidimensionales resultantes del diferente

barrido de los patrones de contexto, lo que reduce el número total de patrones de contextos unidimensionales que es necesario almacenar. Esto permite utilizar un patrón de contextos para dos o más tipos de barrido del subbloque.

5 Por ejemplo, tal como se ha descrito anteriormente, uno de los patrones de contextos bidimensionales incluye contextos que cuando se leen mediante barrido horizontal, vertical y diagonal dan como resultado el mismo patrón de contextos unidimensional. Por consiguiente, para este patrón de contextos solo se almacena un patrón de contextos unidimensional, porque el patrón de contextos unidimensional es el mismo para los tres tipos de barridos de patrones de contexto.

10 En otro ejemplo, uno de los patrones de contextos bidimensionales incluye contextos que cuando se leen mediante barrido horizontal o vertical dan como resultado el mismo patrón de contextos unidimensional. En este caso, se almacena el patrón de contextos unidimensional para el barrido diagonal y se almacena el patrón de contextos unidimensional para el barrido horizontal o el barrido vertical, pero no ambos, porque el patrón de contextos unidimensional que resulta del barrido horizontal y vertical es el mismo. En estos ejemplos, estos patrones de contextos unidimensionales calculados a partir de los patrones de contextos bidimensionales pueden precalcularse y almacenarse, lo que puede agilizar los procesos de codificación y decodificación.

15 Además, en algunos ejemplos, el patrón de contextos unidimensional no tiene necesariamente que calcularse a partir de un patrón de contextos bidimensional. Más bien, el patrón de contextos unidimensional puede preseleccionarse y almacenarse como un patrón de contextos unidimensional. Incluso en estos ejemplos, el patrón de contextos unidimensional puede identificar contextos para dos o más tipos de barrido del subbloque actual.

20 Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo para codificar y decodificar información de vídeo digital con más eficacia. Las técnicas de compresión de vídeo pueden definirse de acuerdo con una norma de codificación de vídeo, tal como la norma HEVC actualmente en elaboración por el JCT-VC. Los trabajos de normalización HEVC se basan en un modelo de dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM supone mejoras en las capacidades de los dispositivos de codificación de vídeo con respecto a los dispositivos de codificación de vídeo disponibles durante la elaboración de normas de codificación de vídeo anteriores, por ejemplo, la ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación de predicción intra, la norma HEVC proporciona hasta treinta y cinco modos de codificación mediante predicción intra. Además, como parte de los trabajos de normalización HEVC, el JCT-VC ha definido condiciones de prueba que pueden utilizarse para evaluar cómo las modificaciones individuales a los borradores de la norma HEVC pueden afectar al rendimiento general de codificación. Uno de los criterios utilizados para evaluar el rendimiento de la codificación es la denominada velocidad en baudios.

35 Un borrador de trabajo (WD) reciente de la norma HEVC, denominado "Borrador de trabajo HEVC 7" o "WD7", se describe en el documento JCTVC-I1003_d4, de Bross *et al.*, "High efficiency video coding (HEVC) text specification draft 7, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 9th Meeting: Ginebra, Suiza, abril-mayo de 2012. Además, otro borrador de trabajo reciente de la norma HEVC, el Borrador 9 (WD9), se describe en el documento HCTVC-J1003_d7, de Bross, *et al.*, "High Efficiency Video Coding (HEVC) Text Specification Draft 9," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 11th Meeting: Shanghai, CN, octubre de 2012. La última versión del WD9 se encuentra en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v29.zip.

40 Aunque las técnicas de esta divulgación se describen con respecto a la norma ITU-T H.264 y la próxima norma HEVC, las técnicas de esta divulgación son aplicables, en general, a cualquier norma de codificación de vídeo. La codificación, de acuerdo con algunos de los aspectos propuestos actualmente de la norma HEVC en proceso de elaboración, se describirá en esta solicitud con fines ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser también útiles y aplicarse a otros procesos de codificación de vídeo, tales como los que se definen de acuerdo con la norma H.264 u otra norma, o procesos de codificación de vídeo patentados.

50 Una secuencia de vídeo incluye típicamente una serie de tramas de vídeo, también denominadas imágenes. Un grupo de imágenes (GOP) comprende en general una serie de una o más de las tramas de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, en una cabecera de una o más de las imágenes o en otras ubicaciones, que describen un número de imágenes incluidas en el GOP. Cada trama puede incluir una pluralidad de segmentos. Cada segmento de una imagen puede incluir datos sintácticos de segmento que describen un modo de codificación para el segmento respectivo. Cada segmento puede incluir una pluralidad de bloques de vídeo o unidades de codificación. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

60 Los bloques de vídeo pueden codificarse aplicando técnicas de predicción espacial (intratrama) y/o predicción temporal (intertrama) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Una predicción espacial puede denominarse "modo intra" (modo I), y una predicción temporal puede denominarse "modo inter" (modo P o modo B). Las técnicas de predicción generan un bloque predictivo de datos de vídeo, que también puede denominarse bloque de muestras de referencia. Un bloque de datos de vídeo original que se va a codificar se compara con el bloque predictivo. La diferencia entre el bloque de vídeo original y el bloque predictivo puede

denominarse datos residuales. Los datos residuales son típicamente una matriz de la diferencia entre los valores de píxeles de un bloque predictivo y el bloque original de datos de vídeo.

5 Se puede aplicar una transformada, por ejemplo, una transformada de coseno discreta (DCT) o una transformada conceptualmente similar, una transformada entera, una transformada de wavelet, u otro tipo de transformada, a los datos residuales durante el proceso de codificación para generar un conjunto correspondiente de coeficientes de transformada. Por lo tanto, el bloque original de vídeo puede reconstruirse aplicando una transformada inversa a los coeficientes de transformada y añadiendo los datos residuales al bloque predictivo. Los coeficientes de transformada también pueden cuantificarse. La cuantificación se refiere en general a un proceso en el que unos coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes, proporcionando compresión adicional. Es decir, los valores de los coeficientes de transformada pueden representarse como una cadena de bits de acuerdo con una profundidad de bits definida. Por ejemplo, un valor de n bits puede redondearse a la baja hasta un valor de m bits durante la cuantificación, donde m es menor que n . En algunos casos, la cuantificación puede tener como resultado que los coeficientes de transformada de bajo valor se representen como cero. Los coeficientes de transformada cuantificados pueden denominarse niveles de coeficientes de transformada.

20 Después de la cuantificación, los coeficientes de transformada cuantificados pueden someterse a codificación de entropía de acuerdo con una metodología de codificación de entropía, tal como, por ejemplo, codificación de longitud variable adaptativa según el contenido (CAVLC), codificación aritmética binaria adaptativa según el contexto (CABAC) o codificación de entropía por división de intervalos de probabilidad (PIPE). También puede llevarse a cabo la codificación de entropía de elementos sintácticos, tales como un elemento sintáctico que define un modo de predicción. Para realizar la CAVLC, un codificador de vídeo puede seleccionar un código de longitud variable para un símbolo que se va a transmitir. Las palabras de código en la VLC pueden construirse de forma que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. Para realizar la CABAC, un codificador de vídeo puede asignar un contexto de un modelo de contextos a un símbolo que se va a transmitir.

30 Para algunas técnicas de codificación de entropía se puede utilizar un orden de barrido predefinido para leer mediante barrido los coeficientes de transformada cuantificados a fin de generar un vector en serie de coeficientes de transformada cuantificados que se puede someter a codificación de entropía. Por tanto, de acuerdo con la codificación de vídeo predictiva, los valores residuales que comprenden valores de diferencias de píxeles se pueden transformar en coeficientes de transformada, cuantificar y leer mediante barrido para generar coeficientes de transformada en serie para la codificación de entropía.

35 Para la codificación de vídeo, por ejemplo, una trama de vídeo se puede dividir en uno o más segmentos, donde un segmento incluye un número entero consecutivo de unidades de codificación. Una unidad de codificación (CU) en general se refiere a una zona de imagen rectangular que sirve de unidad básica a la cual se aplican diversas herramientas de codificación para la compresión de vídeo. En general, las técnicas de esta divulgación se refieren a la transformación, cuantificación, lectura mediante barrido y codificación de entropía de los datos de una CU. Una CU es cuadrada típicamente y puede considerarse similar al denominado "macrobloque" que se describe en otras normas de codificación de vídeo, tales como, por ejemplo, la ITU-T H.264. Una CU puede considerarse una matriz de valores de muestra de vídeo. Los valores de muestra de vídeo también pueden denominarse elementos de imagen, píxeles o pels. Una CU habitualmente tiene un componente de luminancia, denominado Y , y dos componentes de croma, denominados U y V . Los dos componentes de croma también pueden denominarse componentes C_b y C_r , respectivamente. El tamaño de una CU puede definirse de acuerdo con un número de muestras horizontales y verticales. Por lo tanto, una CU puede describirse como una CU $N \times N$ o $N \times M$. En esta divulgación, " $N \times N$ " y " N por N " pueden utilizarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de las dimensiones vertical y horizontal, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en la dirección horizontal ($x = 16$). Asimismo, un bloque $N \times N$ presenta en general N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no tienen que presentar necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

60 Para lograr una mejor eficiencia de codificación, una CU puede tener tamaños variables dependiendo del contenido de vídeo. De acuerdo con la HEVC, los datos sintácticos de un flujo de bits pueden definir la unidad de codificación más grande (LCU), que es la CU más grande para una trama o una imagen en términos del número de muestras. Típicamente, una LCU incluye muestras de luma 64×64 , pero el tamaño de una LCU puede variar dependiendo de la aplicación de codificación. Las LCU también pueden denominarse "unidades de árbol de codificación". Las CU de otras dimensiones pueden generarse dividiendo recursivamente una LCU en varias sub-CU. La división de las LCU en sub-CU se puede realizar mediante una estructura de árbol cuaternario conocido como "árbol cuaternario residual" (RQT). Por tanto, las LCU también pueden denominarse bloques de árbol. De acuerdo con la división de árbol cuaternario, un nodo raíz del árbol cuaternario, tal como una LCU, puede dividirse en cuatro nodos más pequeños, y cada nodo hijo puede a su vez dividirse en otros cuatro nodos más pequeños. Los datos sintácticos

para un flujo de bits pueden definir un número máximo de veces en que puede dividirse una LCU, que se denomina profundidad de CU. Por consiguiente, un flujo de bits también puede definir una unidad que es la unidad de codificación más pequeña (SCU). Típicamente, una SCU incluye muestras de luma 8x8. Por tanto, en un ejemplo, pueden generarse cuatro CU 32x32 dividiendo una LCU 64x64 en cuatro sub-CU, y cada una de las CU 32x32 puede dividirse además en dieciséis CU 8x8.

Una CU puede incluir una o más unidades de predicción (PU) asociadas y/o unidades de transformada (TU). En general, una PU incluye datos que se utilizan para generar un bloque predictivo de datos de vídeo para una CU. Las PU también pueden denominarse "divisiones de predicción". Los datos sintácticos asociados a una CU pueden describir la división de una CU en una o más PU. Una TU puede tener una forma cuadrada o no cuadrada. Los datos de tipo incluidos en una PU pueden diferir dependiendo de si la CU está codificada en modo de salto o directo, codificada en modo de predicción intra o codificada en modo de predicción inter. Por ejemplo, cuando la CU va a codificarse en modo intra, una PU puede incluir datos que describen un modo de predicción intra y cuando la CU va a codificarse en modo inter, una PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, una precisión de un cuarto de píxel o una precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia para el vector de movimiento. Después de la predicción mediante las PU de una CU, un codificador de vídeo puede calcular los datos residuales para una CU.

El HM admite la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular sea 2Nx2N, el HM admite la predicción intra en tamaños de PU de 2Nx2N o NxN y la predicción inter en tamaños de PU simétricos de 2Nx2N, 2NxN, Nx2N o NxN. El HM también admite la división asimétrica para la predicción inter en tamaños de PU de 2Nx n U, 2Nx n D, n Lx2N y n Rx2N. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida en 25 % y 75 %. La parte de la CU correspondiente a la división de 25 % está indicada por una "n" seguida de una indicación "arriba", "abajo", "izquierda" o "derecha". Así, por ejemplo, "2Nx n U" se refiere a una CU 2Nx2N que está dividida horizontalmente con una PU 2Nx0,5N encima y una PU 2Nx1,5N debajo.

Como se ha descrito anteriormente, puede aplicarse una transformada a unos datos residuales para transformar los datos residuales de un dominio de píxel en un dominio de transformada. La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU típicamente se basa en el tamaño de las PU de una CU dada, aunque puede que no siempre sea así. El tamaño de una TU puede ser igual al tamaño de una CU, o una CU puede dividirse en una pluralidad de TU. Las TU presentan típicamente el mismo tamaño o un tamaño más pequeño que las PU. Los datos sintácticos asociados a una CU también pueden describir la división de la CU en una o más TU de acuerdo con un árbol cuaternario.

En general, se utiliza una TU en el proceso de transformar datos residuales en coeficientes de transformada. Una TU puede tener forma cuadrada o no cuadrada. Por ejemplo, un bloque de 8x8 valores residuales puede transformarse en un conjunto de 8x8 coeficientes de transformada. Este conjunto de coeficientes de transformada se puede denominar más generalmente como bloque de transformada. Por ejemplo, se puede aplicar una transformada a los valores residuales asociados con la matriz de muestras 16x16 o puede aplicarse una transformada a cada una de las cuatro matrices de muestras 8x8. Las TU más grandes en general proporcionan más compresión con un efecto pixelado más perceptible en una imagen reconstruida, mientras que las TU más pequeñas proporcionan en general menos compresión con menos efecto pixelado perceptible. La selección de tamaños de TU puede basarse en un análisis de optimización velocidad-distorsión. Los tamaños de TU disponibles pueden incluir TU 32x32, 16x16 y 8x8. Debe observarse que esta divulgación utiliza típicamente el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede utilizar el término "bloque de vídeo" para referirse a un bloque de árbol, es decir, una LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

Conceptualmente, un bloque de transformada o TU puede ser una matriz bidimensional (2D) de coeficientes de transformada. Como se ha descrito anteriormente, un codificador de vídeo puede realizar una operación de cuantificación en un bloque de transformada. Puede utilizarse un orden de barrido predefinido para leer mediante barrido los coeficientes de transformada cuantificados a fin de generar un vector en serie de coeficientes de transformada cuantificados. El vector en serie de coeficientes de transformada cuantificados puede entonces someterse a codificación de entropía para proporcionar compresión adicional.

En algunos ejemplos, para TU 16x16 y 32x32, se utiliza un barrido de subbloque 4x4 a fin de generar un vector en serie de coeficientes de transformada cuantificados. Por ejemplo, los subbloques se leen mediante barrido en la dirección inversa, desde la parte superior derecha hasta la parte inferior izquierda. Dentro de un subbloque, los coeficientes de transformada también se leen mediante barrido en la dirección inversa, desde la parte inferior derecha hasta la parte superior izquierda. Este tipo de barrido puede denominarse barrido diagonal de subbloque 4x4. En algunos ejemplos, las TU 8x8 también pueden utilizar el barrido diagonal de subbloque 4x4 como uno de los posibles barridos. La FIG. 1 es un diagrama conceptual que ilustra un ejemplo de grupos de coeficientes y barridos

para un bloque de vídeo. La FIG.1 ilustra un bloque de vídeo 8x8 dividido en cuatro subbloques 4x4, donde se realiza un barrido diagonal en cada uno de los subbloques. Un subbloque también puede denominarse grupo de coeficientes. En la FIG. 1, cada grupo de coeficientes se identifica y separa mediante líneas interiores más gruesas. Como se ilustra en la FIG. 1, hay cuatro grupos de coeficientes y cada grupo de coeficientes incluye 16 coeficientes. Los barridos realizados dentro de los subbloques de la FIG. 1 se muestran mediante las flechas de dirección.

Además del barrido diagonal de subbloque 4x4, los barridos de coeficientes dependientes del modo permiten barridos horizontales y verticales no cuadrados para algunos modos de predicción intra 8x8. Para los barridos horizontales y verticales no cuadrados de una TU 8x8, los grupos de coeficientes se definen como rectángulos 8x2 para el barrido horizontal no cuadrado (es decir, 16 coeficientes consecutivos en el orden de barrido). De manera similar, los grupos de coeficientes se definen como rectángulos 2x8 para el barrido vertical no cuadrado. Las FIGS. 2A-2B ilustran grupos de coeficientes no cuadrados con barridos horizontales y verticales de una TU 8x8, respectivamente. En las FIGS. 2A-2B, cada grupo de coeficientes se identifica y separa mediante líneas interiores más gruesas. Como se ilustra en las FIG. 2A-2B, hay cuatro grupos de coeficientes y cada grupo de coeficientes incluye 16 coeficientes. En la FIG. 2A, los subbloques se leen mediante barrido de derecha a izquierda. Como se ilustra en la FIG. 2B, los subbloques se leen mediante barrido de abajo arriba. Los barridos que se realizan dentro de los subbloques de las FIGS. 2A-2B se muestran mediante las flechas de dirección. Debe observarse que un codificador de vídeo diseñado de acuerdo con la norma HEVC WD7 puede requerir un acceso de datos para los grupos de coeficientes no cuadrados diferente al de los grupos de coeficientes de subbloque 4x4 habituales. Esto puede imponer una complejidad de hardware y software adicional durante la implementación del codificador de vídeo.

Por tanto, además del barrido diagonal de subbloque 4x4 y los barridos horizontal y vertical no cuadrados, se han propuesto también barridos horizontales y verticales de subbloque para los subbloques 4x4 de una TU 8x8 como alternativa a los barridos horizontales y verticales no cuadrados. Se describen ejemplos de barridos horizontales y verticales de subbloque en las publicaciones: (1) Rosewarne, C., Maeda, "M. Non-CE11: Harmonisation of 8x8 TU residual scan" JCT-VC Contribution JCTVC-H0145;(2) Yu, Y., Panusopone, K., Lou, J., Wang, L. "Adaptive Scan for Large Blocks for HEVC; JCT-VC Contribution JCTVCF569; y (3) Solicitud de patente de Estados Unidos n.º 13/551.458, presentada el 17 de julio de 2012, cada una de las cuales se incorpora al presente documento a título de referencia. Dos ejemplos de barridos horizontales y verticales de subbloque 4x4 que pueden utilizarse para una TU 8x8 se ilustran en las FIGS. 3A-3B. En las FIGS. 3A-3B, cada grupo de coeficientes se identifica y separa mediante líneas interiores más gruesas. De forma similar al ejemplo mostrado en la FIG. 1, en las FIGS. 3A-3B hay cuatro grupos de coeficientes 4x4 que se leen mediante barrido en la dirección inversa, desde la parte superior derecha hasta la parte inferior izquierda. Los barridos que se realizan dentro de los subbloques se ilustran mediante flechas de dirección.

Debe tenerse en cuenta que mientras se utiliza el barrido horizontal de subbloque 4x4 y el barrido vertical de subbloque 4x4 como alternativas a los barridos horizontales y verticales no cuadrados, de acuerdo con la propuesta JCTVC-H0145, la utilización del conjunto del barrido diagonal de subbloque 4x4, el barrido horizontal de subbloque 4x4 y el barrido vertical de subbloque 4x4 como barridos posibles de una TU 8x8 dio como resultado una pérdida de rendimiento de la velocidad en baudios del 0,3 % para la configuración intra en comparación con la utilización de un conjunto del barrido diagonal de subbloque 4x4, el barrido horizontal rectangular 8x2 y el barrido vertical rectangular 2x8 como barridos posibles para una TU 8x8. Es decir, la sustitución de los barridos ilustrados en las FIGS. 2A-2B por los barridos ilustrados en las FIGS. 3A-3B disminuyó el rendimiento de codificación en un 0.3 % en algunos casos de prueba.

En los ejemplos de codificación de vídeo, cada subbloque se codifica mediante cinco pases de codificación, a saber: (1) un pase de significancia, (2) un pase de mayor que uno, (3) un pase de mayor que dos, (4) un pase de signo y (5) un pase de nivel de coeficiente restante. La codificación de significancia se refiere a la generación de elementos sintácticos para indicar si alguno de los coeficientes de un subbloque tiene un valor de uno o superior. Es decir, un coeficiente con un valor de uno o superior se considera significativo. La codificación de la significancia incluye dos partes. Para la primera parte de la codificación de significancia, se codifica o deduce un elemento sintáctico para cada grupo de coeficientes (es decir, un subbloque 4x4) que indica si hay algún coeficiente distinto de cero en el subbloque. Un ejemplo de dicho elemento sintáctico se denomina indicador de grupo de coeficientes (CGF). En HEVC WD7, el CGF puede representarse mediante el elemento sintáctico **significant_coeff_group_flag**. En HEVC WD9, el nombre del elemento sintáctico del indicador de grupo de coeficientes se ha cambiado de **significant_coeff_group_flag** a **coded_sub_block_flag** (que también puede denominarse CSBF, ya que los grupos de coeficientes son subbloques 4x4). Esta divulgación se refiere a un indicador de grupo de coeficientes como CGF, que puede corresponder a cualquiera de los elementos sintácticos **significant_coeff_group_flag** o **coded_sub_block_flag**.

Para la segunda parte de la codificación significativa, si el CGF es 1 (es decir, si hay coeficientes distintos de cero en el subbloque), entonces se generan los elementos sintácticos para cada coeficiente de transformada del grupo de coeficientes, que indican si el coeficiente de transformada es significativo o no (es decir, un valor de uno o superior). Los ejemplos de dichos elementos sintácticos se denominan elementos sintácticos de significancia, ejemplos de los cuales son los indicadores de coeficientes significativos. En HEVC WD7 y WD9, los indicadores de coeficientes

significativos están representados por el elemento sintáctico **significant_coefficient_flag**.

En otras palabras, para codificar la significancia de los coeficientes, se codifican dos tipos de elementos sintácticos. Un primer elemento sintáctico (por ejemplo, CGF) se codifica para cada grupo de codificación (es decir, un subbloque) que indica si hay coeficientes distintos de cero en el grupo de codificación. Si el primer elemento sintáctico indica que hay al menos un coeficiente distinto de cero en el grupo de codificación, se codifica un segundo elemento sintáctico (por ejemplo, un elemento sintáctico de significancia o **significant_coefficient_flag**) para cada coeficiente del grupo de codificación que indica si un coeficiente es cero o un coeficiente distinto de cero.

10 Un pase de mayor que uno genera elementos sintácticos para indicar si el valor absoluto de un coeficiente significativo es mayor que uno. En un ejemplo, un elemento sintáctico denominado **coeff_abs_level_greater1_flag** (abreviado "**gr1Flag**") proporciona una indicación sobre si un coeficiente significativo tiene un valor absoluto mayor que uno. De manera similar, el pase de mayor que dos genera elementos sintácticos para indicar si el valor absoluto de un coeficiente mayor que uno es mayor que dos. En un ejemplo, un elemento sintáctico denominado **coeff_abs_level_greater2_flag** (abreviado "**gr2Flag**") proporciona una indicación sobre si un coeficiente mayor que uno tiene un valor absoluto mayor que dos.

20 Un pase de signo genera elementos sintácticos para indicar la información de signo para los coeficientes significativos. En un ejemplo, un elemento sintáctico denominado **coeff_sign_flag** (abreviado "**signFlag**") puede indicar la información de signo para un coeficiente significativo. Por ejemplo, un valor de 0 para **signFlag** puede indicar un signo positivo, mientras que un valor de 1 puede indicar un signo negativo. Un pase de nivel de coeficiente restante genera elementos sintácticos que indican el valor absoluto restante de un nivel de coeficiente de transformada (por ejemplo, el valor restante). En un ejemplo, un elemento sintáctico denominado **coeff_abs_level_remain** (abreviado "**levelRem**") puede proporcionar esta indicación. El elemento sintáctico **levelRem** no puede señalarse a menos que **gr2Flag** esté presente para cualquier coeficiente dado, por ejemplo, aunque dicha limitación no siempre es necesaria. En un ejemplo, un coeficiente con un valor *level* puede codificarse como $(\text{abs}(\text{level}) - x)$, donde el valor de *x* depende de la presencia de **gr1Flag** y **gr2Flag**. Por ejemplo, *x* puede ser igual a 3 si **gr2Flag** está presente. En algunos ejemplos, el valor *level* se puede codificar como $(\text{abs}(\text{level}) - 3)$ para cualquier coeficiente para el cual haya un resto. Debe observarse que el sistema de cinco pases es solo un ejemplo de técnica que se puede utilizar para codificar el coeficiente de transformada y que las técnicas descritas en el presente documento pueden ser igualmente aplicables a otras técnicas.

35 Asimismo, aparte de los elementos sintácticos descritos anteriormente, puede indicarse una posición de un último coeficiente significativo de una TU en el flujo de bits. La posición del último coeficiente significativo en la TU depende de un orden de barrido asociado a la TU. El orden de barrido para el propósito de identificar un último coeficiente significativo puede ser cualquiera de los órdenes de barrido descritos anteriormente u otro orden de barrido predeterminado. En HEVC WD7, la posición del último coeficiente significativo de un bloque se indica especificando un valor de coordenada *x* y un valor de coordenada *y*. El valor de la coordenada *x* puede indicarse mediante los elementos sintácticos **last_significant_coeff_x_prefix** y **last_significant_coeff_x_suffix**. El valor de coordenadas *y* puede indicarse mediante los elementos sintácticos **last_significant_coeff_y_prefix** y **last_significant_coeff_y_suffix**.

45 De esta manera, los elementos sintácticos descritos anteriormente se pueden utilizar para señalar un denominado mapa de significancia de coeficientes de transformada, donde un mapa de significancia ilustra la posición de los coeficientes significativos con una TU. La FIG. 4 es un diagrama conceptual que ilustra una relación entre los coeficientes de transformada de un bloque de transformada y un mapa de significancia asociado con el mapa de transformada. Como se ilustra en la FIG. 4, el mapa de significancia incluye un "1" para indicar cada aparición de un valor de coeficiente significativo, es decir, un valor mayor que cero, en el bloque de transformada. Además, en este ejemplo, el valor del CFG es "1" para indicar que hay al menos un coeficiente distinto de cero en el grupo de codificación (es decir, el subbloque).

55 Por ejemplo, el subbloque de la izquierda, en la FIG. 4, ilustra ejemplos de coeficientes de transformada (por ejemplo, coeficientes de transformada cuantificados). Como se ilustra, hay al menos un coeficiente distinto de cero en el subbloque, por lo tanto el CFG es 1. Además, el mapa de significancia situado a la derecha en la FIG. 4 incluye el elemento sintáctico de significancia (por ejemplo, el indicador de coeficiente significativo) para cada coeficiente de transformada del subbloque. Por ejemplo, el valor 1 del indicador de coeficiente significativo para todos los coeficientes de transformada correspondientes indica que el valor de estos coeficientes de transformada no es cero (es decir, un coeficiente de transformada distinto de cero) y el valor 0 para todos los coeficientes de transformada correspondientes indica que el valor de estos coeficientes de transformada es cero.

60 En HEVC, los elementos sintácticos relacionados con los coeficientes de transformada cuantificados, como los **significant_coeff_group_flag** y **significant_coefficient_flag** descritos anteriormente y otros elementos sintácticos pueden someterse a codificación de entropía mediante CABAC (codificación aritmética binaria adaptativa según el contexto). Para aplicar la codificación CABAC a un elemento sintáctico, la binarización se puede aplicar a un elemento sintáctico a fin de formar una serie de uno o más bits, que se denominan *bins*. Además, un contexto de codificación puede estar asociado con un *bin* del elemento sintáctico. El contexto de codificación puede identificar

probabilidades de codificación de *bins* que tienen valores particulares. Por ejemplo, un contexto de codificación puede indicar una probabilidad de 0,7 de codificar un *bin* de valor 0 (que representa un ejemplo de "símbolo más probable" en este caso) y una probabilidad de 0,3 de codificar un *bin* de valor 1. Después de identificar el contexto de codificación, se puede codificar el *bin* aritméticamente basándose en el contexto. En algunos casos, los contextos asociados con un elemento sintáctico particular o sus *bins* pueden depender de otros elementos sintácticos o parámetros de codificación.

Por ejemplo, la obtención del contexto del CGF depende del orden de barrido del correspondiente grupo de coeficientes. Por ejemplo, para un grupo de coeficientes leídos mediante barrido diagonal de subbloque 4x4 (por ejemplo, en los casos de las TU 16x16 y 32x32 y algunas TU 8x8), el contexto del CGF (es decir, el contexto del elemento sintáctico **significant_coeff_group_flag**) depende del CGF del subbloque situado a la derecha (CGF_R) y debajo (CGF_B) del grupo de coeficientes. Para los barridos horizontales y verticales no cuadrados ilustrados en la FIG. 2A-2B (por ejemplo, en ese caso de una TU 8x8), el contexto del CGF del grupo de coeficientes depende solo del CGF del grupo de coeficientes codificados previamente. Para el barrido horizontal, el grupo de coeficientes codificados previamente se refiere al grupo de coeficientes codificados situado debajo del grupo de coeficientes. Para el barrido vertical, el grupo de coeficientes anterior se refiere al grupo de coeficientes situado a la derecha del grupo de coeficientes.

Debe tenerse en cuenta que, debido a que en HEVC WD7 la obtención del contexto de un CGF es diferente para el barrido diagonal de subbloque 4x4 (es decir, depende de CGF_R y CGF_B) y los barridos horizontal y vertical no cuadrados (es decir, depende solo del CGF anterior), se requiere una ruta lógica diferente para la obtención del contexto del CGF para el grupo de coeficientes de subbloque diagonal 4x4 y la obtención del contexto del CGF para los barridos horizontal y vertical no cuadrados, lo que también puede imponer complejidad de hardware y software.

En HEVC WD7, los contextos asignados al elemento sintáctico **significant_coefficient_flag** dependen de (1) la posición del coeficiente de transformada dentro del subbloque 4x4, (2) los CGF del subbloque situado a la derecha (CGF_R) y debajo (CGF_B) del subbloque actual y (3) la presencia o no del coeficiente DC en el subbloque. La publicación de Kumakura, T., Fukushima, S. "Non-CE3: Simplified context derivation for significant map" JCT-VC Contribution JCTVC-I0296, que se incorpora en su totalidad al presente documento a título de referencia, proporciona un ejemplo en el que se asignan contextos a los coeficientes de un subbloque 4x4, dependiendo de los valores de CGF_R y CGF_B y la posición de un coeficiente de un subbloque.

Las FIGS. 5A-5D ilustran cuatro patrones diferentes para asignaciones de contexto de indicadores de coeficiente significativo de un subbloque 4x4 dependientes de CGF_R y CGF_B. Cabe señalar que aunque en las FIGS. 5A-5D la numeración del contexto comienza con 0, esto tiene una finalidad ilustrativa y no refleja los números de contexto reales utilizados en HEVC WD7, sino solo una numeración relativa al contexto. Como se ilustra en las FIG. 5A a 5D, cada patrón de contextos incluye 16 valores de contexto, donde cada valor de contexto corresponde al coeficiente situado en la posición respectiva. Además, como se ilustra en las FIGS. 5A a 5D, un patrón de contextos se determina basándose en valores de CGF_R y CGF_B. De esta manera, los patrones de las FIGS. 5A a 5D ilustran un ejemplo en el que se asignan contextos a indicadores de coeficientes significativos basándose en la posición del coeficiente de transformada en el subbloque 4x4 y los valores de CGF_R y CGF_B. Cabe señalar que las asignaciones de contexto ilustradas en las FIGS. 5A-5D no son óptimas para los barridos horizontales y verticales de subbloque ilustrados en las FIGS. 3A-3B basados en la localización probable de los coeficientes significativos dentro de un vector generado mediante los barridos.

Además, en un ejemplo, los valores de contextos de un patrón de contextos se pueden modificar basándose en si el subbloque correspondiente incluye o no el coeficiente DC. El coeficiente DC puede ser el primer coeficiente de la transformada y puede indicar en general la cantidad media de energía de todo el bloque, por ejemplo. Para los coeficientes de transformada luma, si un subbloque 4x4 no contiene un coeficiente DC, puede aplicarse un desplazamiento de contexto. En algunos ejemplos, se aplica un desplazamiento de contexto de 3. En un ejemplo, si un subbloque 4x4 no contiene ningún coeficiente DC, y una asignación de contexto obtenida a partir de un patrón de contextos es 2, el contexto real utilizado puede ser 5. En otras palabras, el proceso de obtención del contexto puede ser exactamente el mismo en ambos casos (es decir, se selecciona un patrón a partir de un conjunto de patrones basándose en los valores de CGF_R y CGF_B), pero se utilizan diferentes conjuntos de contextos para subbloques DC y no DC. Es decir, los bloques DC y no DC no comparten los mismos contextos.

Esta divulgación utiliza el término "subbloque DC" para referirse a un subbloque de un bloque (por ejemplo, una TU) que incluye un coeficiente DC del bloque. Por ejemplo, suponiendo que el coeficiente DC de una TU es un coeficiente situado en el extremo superior izquierdo, un subbloque situado en el extremo superior izquierdo de la TU que incluye el coeficiente DC se puede denominar subbloque DC. Además, en un ejemplo, para los coeficientes de transformada de croma, la determinación de desplazamiento de contexto basada en la presencia o no del coeficiente DC en el subbloque 4x4 no se aplica. Es decir, los contextos se comparten para subbloques DC y subbloques no DC para coeficientes de transformada de croma. Por tanto, en algunos casos solo se utilizan tres contextos para los coeficientes de transformada asociados con los componentes de croma. Además, en algunos casos, un coeficiente DC siempre puede utilizar un contexto separado, que se comparte para todos los tamaños de TU. Además, en HEVC WD7, la obtención de contexto de mapa de significancia para una TU 8x8 utiliza una tabla 8x8 escalada para

la asignación de contexto y, así pues, la codificación de mapa de significancia para una TU 8x8 no se unifica con la obtención de contexto de mapa de significancia para las TU 16x16 y 32x32.

5 Esta divulgación describe varias técnicas para codificar los elementos sintácticos asociados a los coeficientes de transformada incluidos en un bloque de transformada, tales como el elemento sintáctico de indicador de grupo de codificación (es decir, **significant_coeff_group_flag** o **coded_sub_block_flag**) y los elementos sintácticos de coeficiente significativo (es decir, **significant_coefficient_flag**). En particular, esta divulgación describe técnicas en las que los órdenes de barrido en las FIGS. 3A-3B pueden utilizarse como alternativa a los grupos de coeficientes no cuadrados ilustrados en las FIGS. 2A-2B. Además, esta divulgación describe técnicas de obtención de contexto para los elementos sintácticos asociados con coeficientes de transformada, en las que las técnicas se basan en las características de los barridos de subbloque ilustrados en las FIGS. 3A-3B. En un ejemplo, las técnicas de obtención de contexto pueden mitigar la pérdida de rendimiento de velocidad en baudios, como se ha descrito anteriormente, cuando se utilizan los barridos ilustrados en las FIGS. 3A-3B en lugar de los barridos ilustrados en las FIGS. 2A-2B.

15 La FIG. 6 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo 10 que puede configurarse para asignar contextos mediante las técnicas descritas en esta divulgación. Como se muestra en la FIG. 6, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificado que un dispositivo de destino 14 va a decodificar en un momento posterior. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluidos ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo en continuo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica. No obstante, las técnicas de esta divulgación no están limitadas necesariamente a las aplicaciones o los entornos inalámbricos. Las técnicas pueden aplicarse a la codificación de vídeo, como apoyo a cualquiera de entre una diversidad de aplicaciones de multimedia, tales como radiodifusiones de televisión a través del aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, por medio de Internet, codificación de vídeo digital para almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para admitir la transmisión de vídeo unidireccional o bidireccional a fin de ofrecer aplicaciones tales como la emisión de vídeo en continuo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

35 En el ejemplo de la FIG. 6, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captación de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo captado previamente, una interfaz de entrada de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo y/o un sistema de gráficos de ordenador para generar datos de gráficos de ordenador como el vídeo de origen, o una combinación de dichas fuentes. En un ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videotéfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. El vídeo captado, precaptado o generado por ordenador puede ser codificado por el codificador de vídeo 20. Los datos de vídeo codificado pueden transmitirse directamente al dispositivo de destino 14 a través de la interfaz de salida 22 del dispositivo de origen 20 por medio del enlace 16. Los datos de vídeo codificado pueden almacenarse, de forma adicional o alternativa, en el dispositivo de almacenamiento 32 para su posterior acceso por el dispositivo de destino 14 u otros dispositivos, con fines de decodificación y/o reproducción.

50 El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de transportar los datos de vídeo codificado desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir al dispositivo de origen 12 transmitir datos de vídeo codificado directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificado pueden modularse de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global, tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

65 El dispositivo de almacenamiento 32 puede incluir cualquiera de entre una diversidad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco fijo, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificado. En un ejemplo adicional, el dispositivo de almacenamiento 32 puede corresponder a un servidor

de archivos o a otro dispositivo de almacenamiento intermedio que pueda retener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento 32 a través de transmisión en continuo o descarga. Un servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificado y transmitir esos datos de vídeo codificado al dispositivo de destino 14. Entre los ejemplos de servidores de archivos se incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, unos dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificado a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esta puede incluir un canal inalámbrico (por ejemplo, una conexión wifi), una conexión por cable (por ejemplo, DSL, cable módem, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificado almacenados en un servidor de archivos.

En el ejemplo de la FIG. 6, el dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 34. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificado por el enlace 16 o desde el dispositivo de almacenamiento 32. Los datos de vídeo codificado transmitidos por el enlace 16, o proporcionados en el dispositivo de almacenamiento 32, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20, para su uso por un decodificador de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Dichos elementos sintácticos pueden incluirse con los datos de vídeo codificado transmitidos en un medio de comunicación, almacenarse en un medio de almacenamiento o almacenarse en un servidor de archivos.

El dispositivo de visualización 34 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también puede estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. El dispositivo de visualización 34 visualiza los datos de vídeo decodificado ante un usuario, y puede comprender cualquiera de entre una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma HEVC que actualmente se halla en fase de elaboración, como se ha indicado anteriormente, y en general pueden ajustarse al modelo de prueba HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma ITU-T H.264, o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Además, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo que se modifica para incorporar las técnicas descritas en el presente documento.

Aunque no se muestra en la FIG. 6, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrados, cada uno de ellos, en un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos diferentes. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse como cualquiera de entre una variedad de circuitos de codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de estos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en unos medios legibles por ordenador no transitorios adecuados, y ejecutar las instrucciones en hardware, mediante uno o más procesadores, para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o decodificadores, donde cualquiera de ambos puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

La FIG. 7 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas descritas en esta divulgación. En el ejemplo de la FIG. 8, el codificador de vídeo 20 incluye una unidad de selección de modo 35, una unidad de procesamiento de predicción 41, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformada 52, una unidad de procesamiento de cuantificación 54 y una unidad de codificación de entropía 56. La unidad de procesamiento de predicción 41 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44 y un módulo de predicción intra 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de procesamiento de cuantificación inversa 58, un módulo de transformada inversa 60 y un sumador 62. También puede incluirse un filtro de eliminación de bloques (no mostrado en la FIG. 7) para filtrar límites de bloque a fin de eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará típicamente la salida

del sumador 62. También pueden utilizarse filtros de bucle adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques. Debe observarse que la unidad de procesamiento de predicción 41 y la unidad de procesamiento de transformada 52 no deben confundirse con las PU y las TU descritas anteriormente.

5 Como se muestra en la FIG. 7, el codificador de vídeo 20 recibe datos de vídeo, y la unidad de selección de modo 35 divide los datos en bloques de vídeo. Esta división también puede incluir también la división en segmentos, mosaicos u otras unidades mayores, así como la división en bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de LCU y CU. El codificador de vídeo 20 ilustra en general los componentes que
10 codifican bloques de vídeo de un segmento de vídeo que se va a codificar. Un segmento puede dividirse en varios bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 41 puede seleccionar uno de una pluralidad de posibles modos de codificación, tal como uno de una pluralidad de modos de codificación intra o uno de una pluralidad de modos de codificación inter, para el bloque de vídeo actual, basándose en resultados de errores (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intracodificado o
15 intercodificado resultante al sumador 50 para generar datos de bloques residuales, y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

La unidad de predicción intra 46 de la unidad de procesamiento de predicción 41 puede realizar la codificación de predicción intra del bloque de vídeo actual en relación con uno o más bloques vecinos de la misma trama o
20 segmento que el bloque que va a codificarse, para proporcionar compresión espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 de la unidad de procesamiento de predicción 41 realizan la codificación de predicción inter del bloque de vídeo actual en relación con uno o más bloques predictivos de una o más imágenes de referencia, para proporcionar compresión temporal.

25 La unidad de estimación de movimiento 42 puede estar configurada para determinar el modo de predicción inter para un segmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar segmentos de vídeo de la secuencia como segmentos P o segmentos B. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la
30 unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento que estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual en relación con un bloque predictivo de una imagen de referencia.

35 Un bloque predictivo es un bloque que se averigua que se corresponde estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxeles, lo cual puede determinarse mediante una suma de una diferencia absoluta (SAD), una suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel subentero de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede
40 interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionario de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxel completo y las posiciones de píxel fraccionario, y facilitar un vector de movimiento con una precisión de píxel fraccionario.

45 La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo de un segmento sometido a codificación inter, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse a partir de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales
50 identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación de entropía 56 y a la unidad de compensación de movimiento 44.

La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar obtener o generar el bloque predictivo basándose en el vector de movimiento determinado mediante estimación de
55 movimiento, realizando posiblemente interpolaciones hasta la precisión de subpíxel. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo a los valores de píxeles del bloque de vídeo actual que se está codificando, y generando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque y pueden incluir componentes de
60 diferencia de luma y croma. El sumador 50 representa el componente o los componentes que realizan esta operación de sustracción. La unidad de compensación de movimiento 44 también puede generar elementos sintácticos asociados a los bloques de vídeo y al segmento de vídeo para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del segmento de vídeo.

65 La unidad de predicción intra 46 puede realizar la predicción intra de un bloque actual, como alternativa a la

predicción inter llevada a cabo por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de predicción intra 46 puede determinar un modo de predicción intra que se va a utilizar para codificar un bloque actual. En algunos ejemplos, la unidad de predicción intra 46 puede codificar un bloque actual mediante varios modos de predicción intra, por ejemplo, durante diferentes pases de codificación, y la unidad de predicción intra 46 (o la unidad de selección de modo 35, en algunos ejemplos) puede seleccionar un modo adecuado de predicción intra que se va a utilizar a partir de los modos probados. Por ejemplo, la unidad de predicción intra 46 puede calcular valores de velocidad-distorsión mediante un análisis de velocidad-distorsión para los diversos modos de predicción intra probados, y seleccionar el modo de predicción intra que tenga las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina en general una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para generar el bloque codificado, así como una velocidad de bits (es decir, un número de bits) utilizada para generar el bloque codificado. La unidad de predicción intra 46 puede calcular proporciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de predicción intra presenta el mejor valor de velocidad-distorsión para el bloque.

En cualquier caso, tras seleccionar un modo de predicción intra para un bloque, la unidad de predicción intra 46 puede proporcionar información que indica el modo de predicción intra seleccionado para el bloque a la unidad de codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica el modo de predicción intra seleccionado de acuerdo con las técnicas de entropía descritas en el presente documento. El codificador de vídeo 20 puede incluir datos de configuración en el flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modos de predicción intra y una pluralidad de tablas de índices de modos de predicción intra modificados (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones del modo de predicción intra más probable, una tabla de índices de modos de predicción intra y una tabla de índices de modos de predicción intra modificados que se van a utilizar para cada uno de los contextos.

Después de que la unidad de procesamiento de predicción 41 genere el bloque predictivo para el bloque de vídeo actual, ya sea mediante la predicción inter o la predicción intra, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo del bloque de vídeo actual. Los datos de vídeo residuales del bloque residual pueden incluirse en una o más TU y aplicarse a la unidad de procesamiento de transformada 52. La unidad de procesamiento de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales mediante una transformada, tal como una transformada de coseno discreta (DCT) o una transformada similar desde un punto de vista conceptual. La unidad de procesamiento de transformada 52 puede convertir los datos de vídeo residuales de un dominio de píxel a un dominio de transformada, tal como un dominio de frecuencia. En algunos casos, la unidad de procesamiento de transformada 52 puede aplicar una transformada bidimensional (2D, en la dirección horizontal y vertical) a los datos residuales de las TU. En algunos ejemplos, la unidad de procesamiento de transformada 52 puede aplicar en cambio una transformada 1D horizontal, una transformada 1D vertical o ninguna transformada a los datos residuales de cada una de las TU.

La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de procesamiento de cuantificación 54. La unidad de procesamiento de cuantificación 54 cuantifica los coeficientes de transformada para reducir más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos de los coeficientes o a todos. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de procesamiento de cuantificación 54 puede llevar a cabo entonces un barrido de la matriz, incluidos los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación de entropía 56 puede llevar a cabo el barrido.

Como se ha descrito anteriormente, el barrido realizado en un bloque de transformada puede basarse en el tamaño del bloque de transformada. La unidad de proceso de cuantificación 54 y/o la unidad de codificación de entropía 56 pueden leer mediante barrido bloques de transformada 8x8, 16x16 y 32x32 mediante cualquier combinación de los barridos de subbloque descritos anteriormente con respecto a la FIG. 1, las FIGS. 2A-2B y las FIGS. 3A-3B. En un ejemplo, unos bloques de transformada 32x32 y unos bloques de transformada 16x16 pueden leerse mediante el barrido diagonal de subbloque 4x4 descrito anteriormente con respecto a la FIG. 1, y unos bloques de transformada 8x8 pueden leerse mediante los barridos de subbloque 4x4 descritos anteriormente con respecto a la FIG. 1 y las FIGS. 3A-3B. Cuando está disponible más de un barrido para un bloque de transformada, la unidad de codificación de entropía 56 puede seleccionar un barrido basándose en un parámetro de codificación asociado con el bloque de transformada, tal como un modo de predicción asociado con una unidad de predicción correspondiente al bloque de transformada. A continuación se proporcionan más detalles con respecto a la unidad de codificación de entropía 56, en relación con la FIG. 8.

La unidad de procesamiento de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican una cuantificación inversa y una transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, para su posterior uso como bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual

reconstruido para calcular valores de píxel subentero y utilizarlos en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque predictivo con compensación de movimiento generado por la unidad de compensación de movimiento 44 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden utilizar el bloque de referencia como bloque de referencia para realizar la predicción inter de un bloque en una trama o imagen de vídeo subsiguiente.

Tras la cuantificación, la unidad de codificación de entropía 56 realiza la codificación de entropía de los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación de entropía 56 puede realizar una codificación de longitud variable adaptativa según el contexto (CAVLC), una codificación aritmética binaria adaptativa según el contexto (CABAC), una codificación aritmética binaria adaptativa según el contexto basada en la sintaxis (SBAC), una codificación de entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos o técnicas de codificación de entropía. Tras la codificación de entropía realizada por la unidad de codificación de entropía 56, el flujo de bits codificado puede transmitirse al decodificador de vídeo 30, o archivar para su posterior transmisión o recuperación por el decodificador de vídeo 30. La unidad de codificación de entropía 56 también puede realizar la codificación de entropía de los vectores de movimiento y los otros elementos sintácticos para el segmento de vídeo actual que se está codificando. La unidad de codificación de entropía 56 puede realizar la codificación de entropía de elementos sintácticos tales como los elementos sintácticos **significant_coeff_group_flag**, **significant_coefficient_flag**, **coeff_abs_level_remain**, **coeff_abs_level_greater1_flag**, **coeff_abs_level_greater2_flag** y **coeff_sign_flag**, descritos anteriormente mediante CABAC.

La FIG. 8 es un diagrama de bloques que ilustra un ejemplo de unidad de codificación de entropía 56 que puede implementar las técnicas descritas en esta divulgación. La unidad de codificación de entropía 56 ilustrada en la FIG. 8 puede ser un codificador CABAC. El ejemplo de unidad de codificación de entropía 56 puede incluir una unidad de binarización 502, una unidad de codificación aritmética 510, que incluye un motor de codificación de derivación 504, un motor de codificación convencional 508 y una unidad de modelización de contextos 506.

La unidad de codificación de entropía 56 puede recibir uno o más elementos sintácticos, tales como cualquiera de los elementos sintácticos **significant_coeff_group_flag** o **coded_sub_block_flag** descritos anteriormente y los elementos sintácticos **significant_coefficient_flag**, **coeff_abs_level_greater1_flag**, **coeff_abs_level_greater2_flag**, **coeff_sign_flag** y **coeff_abs_level_remain**. La unidad de binarización 502 recibe un elemento sintáctico y genera una cadena de *bins* (es decir, una cadena binaria). La unidad de binarización 502 puede utilizar, por ejemplo, una cualquiera o una combinación de las siguientes técnicas para generar una cadena de *bins*: codificación de longitud fija, codificación unaria, codificación unaria truncada, codificación de Rice truncada, codificación de Golomb, codificación de Golomb exponencial y codificación de Golomb-Rice. Además, en algunos casos, la unidad de binarización 502 puede recibir un elemento sintáctico, tal como una cadena binaria, y simplemente transferir los valores de *bin*. En un ejemplo, la unidad de binarización 502 recibe el elemento sintáctico **significant_coeff_group_flag** y genera una cadena de *bins*.

La unidad de codificación aritmética 510 está configurada para recibir una cadena de *bins* desde la unidad de binarización 502 y aplicar una codificación aritmética a la cadena de *bins*. Como se muestra en la FIG. 8, la unidad de codificación aritmética 510 puede recibir valores de *bin* desde una ruta de derivación o la ruta de codificación convencional. Los valores de *bin* que siguen la ruta de derivación pueden identificarse como valores de *bin* codificados por derivación, y los valores de *bin* que siguen la ruta de codificación convencional pueden identificarse como valores de *bin* codificados mediante CABAC. En consonancia con el proceso CABAC descrito anteriormente, en caso de que la unidad de codificación aritmética 510 reciba valores de *bin* desde una ruta de derivación, el motor de codificación de derivación 504 podrá aplicar una codificación aritmética a los valores de *bin* sin utilizar un contexto adaptativo asignado a un valor de *bin*. En un ejemplo, el motor de codificación de derivación 504 puede suponer las mismas probabilidades para los posibles valores de un *bin*.

En caso de que la unidad de codificación aritmética 510 reciba valores de *bin* a través de la ruta convencional, la unidad de modelización de contexto 506 podrá proporcionar una variable de contexto (por ejemplo, un estado de contexto), de manera que el motor de codificación convencional 508 podrá llevar a cabo una codificación aritmética basándose en las asignaciones de contexto proporcionadas por la unidad de modelización de contextos 506. Las asignaciones de contexto pueden definirse de acuerdo con una norma de codificación de vídeo, tal como la futura norma HEVC. Además, en un ejemplo, la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden configurarse para asignar contextos a *bins* de los elementos sintácticos **significant_coeff_group_flag** y **significant_coefficient_flag** basándose en las técnicas descritas en el presente documento. Las técnicas pueden incorporarse a la norma HEVC u otra norma de codificación de vídeo. Los modelos de contexto pueden almacenarse en memoria. La unidad de modelización de contextos 506 puede incluir una serie de tablas indexadas y/o utilizar funciones de correlación para determinar un contexto y una variable de contexto para un *bin* particular. Tras codificar un valor de *bin*, el motor de codificación convencional 508 puede actualizar un contexto basándose en los valores de *bin* reales.

Como se ha descrito anteriormente, las asignaciones de contexto ilustradas en las FIGS. 5A-5D no son óptimas para los barridos horizontales o verticales de subbloque 4x4 ilustrados en las FIGS. 3A-3B. Por ejemplo, los patrones

ilustrados en la FIG. 5A y la FIG. 5D tienen zonas de asignación de contexto divididas a lo largo de una línea diagonal. Estas zonas no se corresponden con la ubicación esperada de los coeficientes significativos cuando se aplica un barrido horizontal o vertical de subbloque 4x4. Además, la primera fila de un barrido horizontal de subbloque 4x4 tiene una probabilidad de ser significativa mucho mayor que la segunda fila. De manera similar, la primera columna de un barrido vertical de subbloque 4x4 tiene una probabilidad de ser significativa mucho mayor que una segunda columna. Por lo tanto, los patrones de contextos ilustrados en las FIGS. 5A a 5D pueden modificarse para proporcionar asignaciones de contexto más óptimas para indicadores de coeficientes significativos que se han leído mediante un barrido horizontal de subbloque 4x4 o un barrido vertical de subbloque 4x4. Por lo tanto, además de realizar la codificación aritmética basada en las asignaciones de contexto definidas de acuerdo con la norma HEVC WD7, el motor de codificación convencional 508 puede configurarse para realizar la codificación aritmética basada en contextos obtenidos de acuerdo con las técnicas divulgadas en el presente documento.

Las FIGS. 9A-9D ilustran patrones de contextos que se basan en posiciones esperadas de coeficientes significativos con respecto a un barrido horizontal de subbloque 4x4 o un barrido vertical de subbloque 4x4. Cabe señalar que aunque en las FIGS. 9A-9D, así como con los patrones de contextos ilustrados en las FIGS. 5A-5D, la numeración del contexto comienza en 0, esto tiene una finalidad ilustrativa y no refleja los números de contexto reales, sino solo la numeración de contexto relativa. Las FIGS. 9A a 9D ilustran cuatro patrones diferentes que pueden utilizarse para asignaciones de contexto de indicadores de coeficientes significativos de un subbloque 4x4 dependiendo del CGF_R y el CGF_B , donde CGF_R se refiere al indicador de grupo de contexto para el grupo de codificación derecho (es decir, subbloque derecho), y CGF_B se refiere al indicador de grupo de contexto para el grupo de codificación inferior (es decir, el subbloque inferior). De nuevo, el elemento sintáctico de indicador de grupo de contexto indica si cualquiera de los coeficientes de transformada de un grupo de codificación es distinto de cero.

De acuerdo con las técnicas descritas en esta divulgación, el codificador de vídeo 20 selecciona un patrón de contextos para codificar elementos sintácticos de significancia, y el decodificador de vídeo 30 selecciona un patrón de contextos para decodificar elementos sintácticos de significancia. Las FIGS. 9A a 9D ilustran ejemplos de una pluralidad de patrones de contextos bidimensionales a partir de los cuales el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar un patrón de contextos para codificar y decodificar. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar un patrón de contextos a partir de los mismos patrones de contextos (por ejemplo, los ilustrados en las FIGS. 9A-9D) para una pluralidad de tipos de barrido. Por ejemplo, para un barrido horizontal, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar entre los mismos patrones de contextos. Para un barrido vertical, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar entre los mismos patrones de contextos que los del barrido horizontal. Para un barrido diagonal, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden seleccionar entre los mismos patrones de contextos que los del barrido horizontal y el barrido vertical.

En general, se puede decir que los patrones de contextos ilustrados en las Figuras 9A-9D tienen un comportamiento de asignación diagonal menor y un comportamiento de asignación de fila en fila/columna en columna mayor en comparación con los patrones respectivos ilustrados en las FIGS. 5A-5D. La unidad de codificación de entropía 56 puede estar configurada para asignar contextos a los elementos sintácticos **significant_coefficient_flag** basándose en los patrones de contextos ilustrados en las FIGS. 9A-9D.

Como se ilustra en las FIGS. 9A-9D, cada patrón de contextos incluye 16 valores de contexto, donde cada valor de contexto corresponde al coeficiente situado en la posición respectiva. Cabe señalar que el patrón de contextos de la FIG. 9D (es decir, para $CGF_B = 1$, $CGF_R = 1$) es uniforme para todas las posiciones (es decir, contexto = 2). La FIG. 9D se puede contrastar con la FIG. 5D, donde el patrón de contextos no es uniforme. Además, los patrones de contextos para $CGF_B = 0$, $CGF_R = 1$, ilustrados en la FIG. 9B, y $CGF_B = 1$, $CGF_R = 0$, ilustrados en la FIG. 9C, dan más importancia a la primera fila y columna, respectivamente, que los patrones de contextos ilustrados en la FIG. 5B y la FIG. 5C. Además, el patrón de contextos para $CGF_B = 0$, $CGF_R = 0$, ilustrado en la FIG. 9A, tiene una forma más cuadrada y da más importancia al coeficiente de la parte superior izquierda en comparación con el patrón de contextos ilustrado en la FIG. 5A.

En un ejemplo, la unidad de codificación de entropía 56 puede estar configurada para asignar contextos a los elementos sintácticos **significant_coefficient_flag** basándose en los patrones de contextos ilustrados en las FIGS. 9A-9D para los barridos horizontal y vertical de subbloque 4x4, y asignar contextos a los elementos sintácticos **significant_coefficient_flag** basándose en los patrones de contextos ilustrados en las FIGS. 5A-5D para el barrido diagonal de subbloque 4x4. En otro ejemplo, con el fin de limitar el número total de patrones de contextos, la unidad de codificación de entropía 56 puede estar configurada para asignar contextos a los elementos sintácticos **significant_coefficient_flag** basándose en los patrones de contextos ilustrados en las FIGS. 9A-9D para todos los barridos diagonales, horizontales y verticales de subbloque 4x4.

Además, pueden utilizarse combinaciones de los patrones de contextos ilustrados en las FIGS. 5A-5D y 9A-9D para asignar contextos para los elementos sintácticos **significant_coefficient_flag**. Por ejemplo, los barridos horizontales de subbloque 4x4 pueden utilizar los patrones de contextos ilustrados en la FIG. 5A, la FIG. 9B, la FIG. 5C y la FIG. 9D para valores respectivos de CGF_B y CGF_R . En este ejemplo, el barrido horizontal de subbloque 4x4 no utiliza ningún patrón con características verticales (es decir, el patrón ilustrado en la FIG. 9C). Esto puede mejorar la

codificación, porque las estadísticas de un barrido horizontal no suelen coincidir con la distribución ilustrada en la FIG. 9C. En otro ejemplo, para el barrido horizontal de subbloque 4x4, en lugar de utilizar el patrón ilustrado en la FIG. 9C para el caso donde ($CGF_B = 1, CBF_R = 0$), se puede utilizar el patrón ilustrado en la FIG. 9D tanto para el caso ($CGF_B = 1, CBF_R = 0$) como el caso ($CGF_B = 1, CBF_R = 1$). En este ejemplo, se comparte un patrón para diferentes configuraciones de CGF para un barrido dado. Dicho uso compartido de patrón también se puede aplicar a los otros tipos de barrido.

Como se ha descrito anteriormente, en un ejemplo, la unidad de procesamiento de cuantificación 54 y/o la unidad de codificación de entropía 56 pueden leer bloques de transformada 32x32 y bloques de transformada 16x16 mediante el barrido diagonal de subbloque 4x4 descrito anteriormente con respecto a la FIG. 1, y los bloques de transformada 8x8 pueden leerse mediante los barridos de subbloque 4x4 descritos anteriormente con respecto a la FIG. 1 y las FIGS. 3A y 3B. En un ejemplo, la unidad de codificación de entropía 56 puede configurarse para asignar contextos a los elementos sintácticos **significant_coefficient_flag** para bloques de transformada 32x32 y 16x16 basándose en los patrones de contextos utilizados para asignar contexto a los elementos sintácticos **significant_coefficient_flag** basándose en los patrones de contextos utilizados para los bloques de transformada 8x8.

En un ejemplo, la unidad de codificación de entropía 56 puede utilizar los patrones de contextos ilustrados en las FIGS. 9A-9D para obtener contextos para cada uno de los bloques de transformada 32x32, 16x16 y 8x8. En otro ejemplo, la unidad de codificación de entropía 56 puede utilizar el conjunto de patrones de contextos, tales como los ilustrados en las FIGS. 5A-5B, para obtener contextos para bloques de transformada 32x32, 16x16 y 8x8 cuando se aplica el barrido diagonal de subbloque 4x4, y utilizar un conjunto diferente de patrones de contextos, tales como los ilustrados en las FIGS. 9A-9D, para obtener los contextos para un bloque de transformada 8x8 cuando se aplica el barrido horizontal o el barrido vertical de subbloque 4x4. En este ejemplo, la obtención de contextos puede compartirse para TU de tamaños variables y puede depender del tipo de barrido.

Además, en relación con el caso de obtención y asignación de contextos a subbloques DC y no DC, aunque la obtención de contextos puede compartirse para cada uno de los bloques de transformada 32x32, 16x16 y 8x8, los contextos reales pueden diferir para cada tamaño de bloque de transformada. Por ejemplo, cada uno de los contextos reales utilizados para bloques de transformada 32x32, 16x16 y 8x8 puede basarse en los patrones de contextos ilustrados en las FIGS. 5A-5B, pero puede aplicarse un desplazamiento a cada uno de los patrones de contextos basándose en el tamaño de la TU. En este ejemplo, cada uno de los bloques de transformada 32x32, 16x16 y 8x8 compartiría una obtención de contexto, pero no contextos reales. En otro ejemplo, la obtención de contextos puede ser igual para todos los subbloques independientemente del tamaño de las TU o el tipo de barrido (por ejemplo, los patrones ilustrados en las FIGS. 9A-9D puede utilizarse para todos los casos), pero puede haber tres conjuntos de contextos reales, un conjunto de contextos para las TU grandes (16x16 y 32x32), un conjunto de contextos para la TU 8x8 con barrido diagonal y un conjunto de contextos para la TU 8x8 cuando se utiliza el barrido horizontal o vertical. Los conjuntos pueden definirse aplicando diferentes desplazamientos a un conjunto de patrones de contextos. De este modo, la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden estar configuradas para asignar contextos a **significant_coefficient_flag** mediante una obtención de contextos unificada para todos los órdenes de barrido.

Como se describió anteriormente, se puede asignar un conjunto de contextos a **significant_coefficient_flag** para el subbloque de DC diferente al de los subbloques no DC. Es decir, pueden aplicarse desplazamientos a los patrones de contextos al determinar los contextos reales. La razón de esto es que las estadísticas para el subbloque DC son típicamente muy diferentes a las estadísticas para los subbloques no DC cuando se utiliza un barrido diagonal de subbloque 4x4. Sin embargo, cuando los subbloques se leen mediante un barrido horizontal o vertical de subbloque 4x4, las estadísticas para el subbloque DC y un subbloque no DC pueden ser similares. Por ejemplo, para una TU 8x8 que utiliza un barrido horizontal de subbloque, el subbloque situado a la derecha del subbloque DC puede tener estadísticas que son más similares al subbloque DC que a los otros subbloques no-DC. De manera similar, para el barrido vertical, el subbloque situado debajo del subbloque DC puede tener estadísticas que son más similares al subbloque DC que a los otros subbloques no DC.

Con el fin de compensar el hecho de que uno de los subbloques no DC pueda tener estadísticas que sean similares al subbloque DC, la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden estar configuradas para utilizar un primer conjunto de contextos para el subbloque DC y un subbloque no-DC adyacente, y pueden utilizar un segundo conjunto de contextos para asignar contextos para los otros subbloques no DC. Por ejemplo, cuando se utiliza un barrido horizontal de subbloque 4x4 para una TU 8x8, la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden estar configuradas para utilizar un primer conjunto de contextos a fin de asignar contextos a la primera fila de subbloques, y un segundo conjunto de contextos a fin de asignar contextos a la segunda fila de los subbloques. Por ejemplo, la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden estar configuradas para utilizar los patrones de contextos ilustrados en las FIGS. 5A-5D y los patrones de contextos ilustrados en las FIGS. 9A-9D para la segunda fila. Además, pueden seguir aplicándose desplazamientos para los subbloques no DC, de manera que el subbloque DC pueda tener todavía un conjunto de contextos único. De manera similar, para un tipo de barrido vertical de subbloque, los patrones de contextos pueden asignarse por columnas. Además, este concepto se puede ampliar a TU más grandes con más de dos columnas o filas. La técnica de obtener y asignar un contexto basándose en la fila

o columna del subbloque puede aplicarse a las TU de todos los tamaños. De este modo, la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden configurarse para asignar contextos a **significant_coefficient_flag** basándose en un tipo de barrido de subbloque y la ubicación de un subbloque dentro de un bloque de transformada.

5 Como se ha descrito anteriormente, las FIGS. 9A-9D ilustran ejemplos de patrones de contextos que el codificador de vídeo 20 selecciona para determinar los contextos para codificar elementos sintácticos de significancia de coeficientes de transformada de un subbloque de un bloque de transformada. En algunos ejemplos, para una pluralidad de tipos de barrido de un subbloque (es decir, dependiendo de si el subbloque va a leerse mediante
10 barrido horizontal, vertical o diagonal), el codificador de vídeo 20 puede seleccionar el patrón de contextos entre los patrones de contextos ilustrados en las FIGS. 9A-9D. En otras palabras, los patrones de contextos a partir de los cuales el codificador de vídeo 20 puede seleccionar un patrón de contextos puede ser los mismos para una pluralidad de tipos de barrido (por ejemplo, un barrido horizontal, un barrido vertical y un barrido diagonal).

15 Además, como se ilustra en las FIGS. 9A-9D, cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más subbloques vecinos. Por ejemplo, el uno o más subbloques vecinos incluyen un primer subbloque vecino y un segundo subbloque vecino. En algunos ejemplos, cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en el primer subbloque vecino (por ejemplo, un subbloque inferior
20 que se halla debajo del subbloque actual), y la inclusión o no de algún coeficiente de transformada distinto de cero en el segundo subbloque vecino (por ejemplo, un subbloque derecho que se halla a la derecha del subbloque actual). Como se ha descrito anteriormente, el CGF_B indica si un subbloque inferior incluye algún coeficiente de transformada distinto de cero y el CGF_R indica si un subbloque derecho incluye algún coeficiente de transformada distinto de cero.

25 El codificador de vídeo 20 puede seleccionar uno de los patrones de contextos ilustrados en las FIGS. 9A-9D basándose en diversos factores, como se describe a continuación. En cualquier caso, el codificador de vídeo 20 puede asignar contextos a cada uno de los elementos sintácticos de significancia de los coeficientes de transformada basándose en el patrón de contextos seleccionado.

30 Por ejemplo, si el codificador de vídeo 20 selecciona el patrón de contextos asociado a la condición sobre la no inclusión de ningún coeficiente de transformada distinto de cero en el primer subbloque vecino y la inclusión de al menos un coeficiente de transformada distinto de cero en el segundo subbloque vecino (es decir, CGF_B es igual a 0 y CGF_R es igual a 1), entonces el codificador de vídeo 20 puede asignar un contexto a una primera fila de los
35 elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual que es diferente de los contextos para otras filas de los elementos de significancia de los coeficientes de transformada para el subbloque actual. Por ejemplo, si CGF_B es igual a 0 y CGF_R es igual a 1, la FIG 9B ilustra que a la primera fila del bloque de transformada se le asigna el contexto de 2 (o 5 suponiendo un desplazamiento de 3) para codificar los elementos sintácticos de significancia de la primera fila del subbloque, que es diferente al contexto para cualquier
40 otra fila del subbloque.

El decodificador de vídeo 30 puede funcionar de una manera sustancialmente similar. Por ejemplo, si el decodificador de vídeo 30 selecciona el patrón de contextos ilustrado en la FIG. 9B (por ejemplo, la condición de que CGF_B es igual a 0 y CGF_R es igual a 1), entonces el decodificador de vídeo 30 asigna contextos a los elementos
45 sintácticos de significancia de un subbloque de un bloque de transformada, tal como corresponda. Por ejemplo, de forma similar al codificador de vídeo 20, el decodificador de vídeo 30 asigna un contexto a una primera fila de los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual, que es diferente de los contextos para otras filas de los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual.

50 En otro ejemplo, si el codificador de vídeo 20 selecciona el patrón de contextos asociado a la condición sobre la inclusión de al menos un coeficiente de transformada distinto de cero en el primer subbloque y la no inclusión de ningún coeficiente de transformada distinto de cero en el segundo subbloque vecino (es decir, CGF_B es igual a 1 y CGF_R es igual a 0), entonces el codificador de vídeo 20 puede asignar un contexto a una primera columna de los
55 elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual, que es diferente de los contextos para otras columnas de los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual. Por ejemplo, si CGF_B es igual a 1 y CGF_R es igual a 0, la FIG 9C ilustra que a la primera columna del bloque de transformada se le asigna el contexto de 2 (o 5 suponiendo un desplazamiento de 3) para codificar los elementos sintácticos de significancia de la primera columna del subbloque, que es diferente
60 del contexto para cualquier otra columna del subbloque.

El decodificador de vídeo 30 puede funcionar de una manera sustancialmente similar. Por ejemplo, si el decodificador de vídeo 30 selecciona el patrón de contextos ilustrado en la FIG. 9C (por ejemplo, la condición de que CGF_B es igual a 1 y CGF_R es igual a 0), entonces el decodificador de vídeo 30 asigna contextos a los elementos
65 sintácticos de significancia de un subbloque de un bloque de transformada, tal como corresponda. Por ejemplo, de forma similar al codificador de vídeo 20, el decodificador de vídeo 30 asigna un contexto a una primera columna de

los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual, que es diferente de los contextos para otras columnas de los elementos de significancia de los coeficientes de transformada para el subbloque actual.

5 En otro ejemplo, si el codificador de vídeo 20 selecciona el patrón de contextos asociado a la condición sobre la inclusión de al menos un coeficiente de transformada distinto de cero en el primer subbloque vecino y la inclusión de al menos un coeficiente de transformada distinto de cero en el segundo subbloque vecino (es decir, CGF_B es igual a 1 y CGF_R es igual a 1), entonces el codificador de vídeo 20 puede asignar un mismo contexto a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual. Por ejemplo, si CGF_B es igual a 1 y CGF_R es igual a 1, la FIG. 9D ilustra que todos los contextos son iguales para el elemento sintáctico de significancia (es decir, 2).

15 El decodificador de vídeo 30 puede funcionar de una manera sustancialmente similar. Por ejemplo, si el decodificador de vídeo 30 selecciona el patrón de contextos ilustrado en la FIG. 9D (por ejemplo, la condición de que CGF_B es igual a 1 y CGF_R es igual a 1), entonces el decodificador de vídeo 30 asigna contextos a los elementos sintácticos de significancia de un subbloque de un bloque de transformada, tal como corresponda. Por ejemplo, de forma similar al codificador de vídeo 20, el decodificador de vídeo 30 asigna un mismo contexto a los elementos de significancia de los coeficientes de transformada para el subbloque actual.

20 Si CGF_B es igual a 0 y CGF_R es igual a 0, el codificador de vídeo 20 puede seleccionar el patrón de contextos ilustrado en la FIG. 9A, y asignar contextos a los elementos sintácticos de significancia de un subbloque de un bloque de transformada, tal como corresponda. El decodificador de vídeo 30 puede funcionar de una manera sustancialmente similar si CGF_B es igual a 0 y CGF_R es igual a 0.

25 Además, los patrones de contextos ilustrados en las FIGS. 9A-9D pueden incluir características adicionales a las descritas anteriormente. Por ejemplo, una de las características del patrón de contextos (por ejemplo, cuando CGF_B es igual a 0 y CGF_R es igual a 0) es que el patrón de contextos incluye contextos que, si se leen mediante barrido horizontal o vertical, dan como resultado un mismo vector unidimensional.

30 Por ejemplo, si el patrón de contextos ilustrado en la FIG. 9A se lee mediante barrido horizontal desde la parte inferior derecha hasta la parte superior izquierda, el vector unidimensional resultante es: [0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 2]. Si el patrón de contextos ilustrado en la FIG. 9A se lee mediante barrido vertical desde la parte inferior derecha hasta la parte superior izquierda, el vector unidimensional resultante es: [0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 2]. Como puede observarse, estos dos vectores unidimensionales son iguales.

35 En otro ejemplo de las características de los patrones de contextos, dos de los patrones de contextos son transposiciones entre sí de tal manera que un barrido horizontal de uno de los patrones de contextos y un barrido vertical de otro de los patrones de contextos dan como resultado el mismo vector unidimensional. Por ejemplo, un barrido horizontal del patrón de contextos ilustrado en la FIG. 9B desde la parte inferior derecha hasta la parte superior izquierda da como resultado el vector unidimensional: [0 0 0 0 0 0 0 1 1 1 1 2 2 2 2]. Una barrido vertical del patrón de contextos ilustrado en la FIG. 9C desde la parte inferior derecha hasta la parte superior izquierda da como resultado el vector unidimensional: [0 0 0 0 0 0 0 1 1 1 2 2 2 2]. Como puede observarse, estos dos vectores unidimensionales son iguales.

45 En otro ejemplo de las características de los patrones de contextos, uno de los patrones de contextos incluye contextos que si se leen mediante barrido horizontal, vertical y diagonal dan como resultado el mismo vector unidimensional. Por ejemplo, un barrido horizontal, un barrido vertical o un barrido diagonal de los contextos del patrón de contextos ilustrado en la FIG. 9D dan como resultado el mismo vector unidimensional: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2].

50 Como se ha descrito anteriormente, el codificador de vídeo 20 selecciona uno de los patrones de contextos ilustrados en las FIGS. 9A-9D basándose en diversos factores. Por ejemplo, el codificador de vídeo 20 selecciona el patrón de contextos basándose en el CGF de los subbloques inferior y derecho. Sin embargo, también puede haber factores adicionales. Por ejemplo, el codificador de vídeo 20 puede seleccionar el patrón de contextos basándose en un tipo de barrido. Si el tipo de barrido es un barrido horizontal del subbloque, entonces el codificador de vídeo 20 puede seleccionar el patrón de contextos ilustrado en la FIG. 9B. Si el tipo de barrido es un barrido vertical del subbloque, entonces el codificador de vídeo 20 puede seleccionar el patrón de contextos ilustrado en la FIG. 9C. El decodificador de vídeo 30 puede funcionar de manera similar.

60 En algunos ejemplos, como los descritos anteriormente, el codificador de vídeo 20 puede excluir patrones de contextos de los patrones de contextos que el codificador de vídeo 20 evalúa para determinar qué patrón de contextos va a seleccionar. Por ejemplo, si el tipo de barrido del subbloque es horizontal, entonces el codificador de vídeo 20 puede determinar que el patrón de contextos ilustrado en la FIG. 9C no puede seleccionarse como el patrón de contextos incluso si el CGF_B es igual a 1 y el CGF_R es igual a 0. En este caso, el codificador de vídeo 20 puede seleccionar el patrón de contextos ilustrado en la FIG. 9D. Por ejemplo, si el tipo de barrido del subbloque es horizontal, entonces el codificador de vídeo 20 puede excluir el patrón de contextos ilustrado en la FIG. 9C. A partir

de los patrones de contextos restantes, el codificador de vídeo 20 puede seleccionar el patrón de contextos ilustrado en la FIG. 9D. En este caso, puede compartirse un patrón de contextos para diferentes valores de CGF_B y CGF_R . Por ejemplo, para el barrido horizontal, el codificador de vídeo 20 puede seleccionar el patrón de contextos ilustrado en la FIG. 9D si CGF_B es igual a 0 y CGF_R es igual a 1 o si CGF_B es igual a 1 y CGF_R es igual a 1.

5 Para un tipo de barrido vertical, el codificador de vídeo 20 puede funcionar de manera similar, excepto porque el codificador de vídeo 20 puede excluir el patrón de contextos ilustrado en la FIG. 9B incluso si CGF_B es igual a 0 y CGF_R es igual a 1. En este ejemplo, el codificador de vídeo 20 selecciona el patrón de contextos de la pluralidad restante de patrones de contextos. El decodificador de vídeo 30 funciona de una manera similar.

10 Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 determinan el tipo de barrido del subbloque actual y determinan al menos un patrón de contextos de la pluralidad de patrones de contextos que no pueden seleccionarse como patrón de contextos basándose en el tipo de barrido determinado del subbloque actual. El codificador de vídeo 20 y el decodificador de vídeo 30 seleccionan el patrón de contextos basándose en la pluralidad de patrones de contextos y excluyendo el por lo menos un patrón de contextos determinado. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 seleccionan el patrón de contextos basándose en la pluralidad de patrones de contextos y excluyendo el por lo menos un patrón de contextos determinado, independientemente de si el subbloque vecino inferior incluye algún coeficiente de transformada distinto de cero y si el subbloque vecino derecho incluye algún coeficiente de transformada distinto de cero.

20 En las FIGS. 5A-5D y las FIGS. 9A-9D, los patrones de contextos se ilustran y definen como bloques 2D. Sin embargo, en algunas implementaciones prácticas, un codificador de vídeo, tal como el codificador de vídeo 20, podría representar un bloque 2D como un vector 1D de acuerdo con el tipo de barrido de subbloque seleccionado, y almacenar el vector 1D para agilizar el proceso de asignación de contexto. En esta situación, incluso si se utiliza el mismo patrón de contextos 2D para asignar contextos para subbloques que utilizaron diferentes tipos de barrido de subbloque, se pueden obtener diferentes vectores 1D basándose en el tipo de barrido de subbloque seleccionado. Por ejemplo, el vector 1D del patrón de contextos ilustrado en la FIG. 9C, leído mediante barrido horizontal, tendría la siguiente representación de vector 1D:

$$30 \quad \textit{Scan_Pattern} = [2 \ 1 \ 0 \ 0 \ 2 \ 1 \ 0 \ 0 \ 2 \ 1 \ 0 \ 0 \ 2 \ 1 \ 0 \ 0]$$

mientras que el patrón de contextos ilustrado en la FIG. 9C, leído mediante barrido vertical, tendría la siguiente representación de vector 1D:

$$35 \quad \textit{Scan_Pattern} = [2 \ 2 \ 2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

40 En este caso, si un codificador de vídeo, tal como el codificador de vídeo 20, almacena patrones de contextos como vectores 1D (es decir, patrones de contextos unidimensionales), puede haber varios vectores para cada patrón de contextos. Una manera de superar el almacenamiento de varios vectores 1D diferentes para cada patrón de contextos es definiendo los patrones de contextos directamente como vectores 1D (es decir, patrones de contextos unidimensionales) y utilizando el mismo vector para dos o más tipos de barrido de subbloque. Por ejemplo, los patrones de contextos con un valor constante (es decir, todo 2) proporcionan el mismo barrido 10D independientemente del tipo de barrido. En este ejemplo, un vector 1D puede especificar un mismo contexto (por ejemplo, 2) o todos los elementos sintácticos de significancia. El vector 1D puede representarse como sigue:

$$45 \quad \textit{Scan_Pattern} = [2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2]$$

50 En otro ejemplo, el patrón de contextos unidimensional define un primer contexto para un primer elemento sintáctico de significancia en un orden de barrido, define un segundo contexto para un segundo y un tercer elementos sintácticos de significancia en el orden de barrido y define un tercer contexto para el resto elementos sintácticos de significancia en el orden de barrido. Por ejemplo, un vector 1D puede especificar un contexto de 2 para el primer indicador de coeficiente significativo, especificar un contexto de 1 para la segunda y tercera asignaciones y especificar un contexto de 0 para las asignaciones restantes, y puede representarse como sigue:

$$55 \quad \textit{Scan_Pattern} = [2 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

60 Otro patrón de contextos posible es $\textit{Scan_Pattern} = [10000000000000 0]$. En este ejemplo, el patrón de contextos define un primer contexto (por ejemplo, 1) para un primer elemento sintáctico de significancia en un orden de barrido y define un segundo contexto (por ejemplo, 0) para los elementos sintácticos de significancia restantes en el orden de barrido. La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de patrón para la asignación de contexto para los coeficientes de un subbloque. El patrón de barrido resultante para el patrón de contextos ilustrado en la FIG. 10 es el mismo para los barridos diagonal, horizontal y vertical de subbloque 4x4 como se ha definido anteriormente.

La unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden configurarse para almacenar patrones de contextos como vectores 1D (es decir, patrones de contextos unidimensionales). En un ejemplo, puede utilizarse el mismo patrón de barrido almacenado para asignar contextos a **significant_coefficient_flag** para una pluralidad de tipos de barrido de subbloque, tales como los barridos diagonal, horizontal y vertical de subbloque 4x4.

En algunos ejemplos, los patrones de contextos unidimensionales pueden precalcularse a partir de patrones de contextos bidimensionales tales como los ilustrados en las FIGS. 9A-9D. Por ejemplo, los patrones de contextos ilustrados en las figuras 9A-9D pueden leerse mediante barrido horizontal, vertical y diagonal para generar los patrones de contextos unidimensionales. Por consiguiente, en este ejemplo puede haber hasta 12 patrones de contextos unidimensionales. Sin embargo, las características de los patrones de contextos ilustrados en las FIGS. 9A-9D pueden ser tales que se precalculan y almacenan menos de 12 patrones de contextos unidimensionales.

Por ejemplo, como se ha descrito anteriormente, el barrido horizontal y el barrido vertical del patrón de contextos ilustrado en la FIG. 9A da como resultado el mismo vector unidimensional. Por consiguiente, el barrido horizontal, vertical y diagonal del patrón de contextos ilustrado en la FIG. 9A da como resultado dos patrones de contextos unidimensionales únicos, en lugar de tres.

También, para los patrones de contextos ilustrados en las FIGS. 9B y 9C, puede haber para uno de ellos tres patrones de contextos unidimensionales únicos (es decir, uno para cada tipo de barrido). Sin embargo, para el otro, puede haber solo dos patrones de contextos unidimensionales únicos. Esto se debe a que el barrido horizontal del patrón de contextos ilustrado en la FIG. 9B y el barrido vertical del patrón de contextos ilustrado en la FIG. 9C da como resultado el mismo patrón de contextos unidimensional. Por consiguiente, hay un total de cinco patrones de contextos unidimensionales únicos entre los patrones de contextos ilustrados en las FIGS. 9B y 9C. En otras palabras, el uno de la pluralidad de patrones de contextos unidimensionales se calcula previamente a partir del patrón de contextos ilustrado en la FIG. 9B (por ejemplo, un primer patrón de contextos bidimensional). El primer patrón de contextos bidimensional incluye contextos que, si se leen mediante barrido horizontal, dan como resultado un mismo patrón de contextos unidimensional que cuando se lee mediante barrido vertical un segundo patrón de contextos bidimensional. Un ejemplo del segundo patrón de contextos bidimensional, cuando el primer patrón bidimensional se ilustra en la FIG. 9B, es el patrón de contextos bidimensional ilustrado en la FIG. 9C.

Para el patrón de contextos ilustrado en la FIG. 9D, solo hay un único patrón de contextos unidimensional (es decir, los barridos diagonales, horizontales y verticales dan como resultado el mismo patrón de contextos unidimensional). Por ejemplo, cuando se utiliza el patrón de contextos ilustrado en la FIG. 9D para precalcular un patrón unidimensional, el patrón unidimensional resultante (tanto si se ha leído mediante barrido vertical, horizontal o diagonal) da como resultado un patrón de contextos que define un mismo contexto (por ejemplo, 2) para todos los elementos sintácticos de los coeficientes de transformada de un subbloque. Por lo tanto, las características de los patrones de contextos ilustrados en las FIGS. 9A-9D dan como resultado un total de ocho patrones de contextos unidimensionales (es decir, dos de la FIG. 9A, cinco de las FIGS. 9B y 9C y uno de la FIG. 9D), que es una cantidad menor que los 12 patrones de contextos unidimensionales que se hubieran necesitado almacenar si los patrones de contextos no comprendieran las características de los patrones de contextos ilustrados en las FIGS. 9A-9D.

Como se ha descrito anteriormente, además de asignar contextos a los elementos sintácticos **significant_coefficient_flag**, en un ejemplo la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden estar configuradas para asignar contextos a **significant_coeff_group_flag**. Como se describe anteriormente, en HEVC WD7 la obtención del contexto de **significant_coeff_group_flag** depende del orden de barrido (es decir, si se aplica un barrido diagonal 4x4, horizontal no cuadrado o vertical). En el caso en el que los barridos no cuadrados se sustituyen por los barridos ilustrados en las FIGS. 3A-3B, la obtención del contexto de **significant_coeff_group_flag** puede modificarse a partir de la obtención de contexto descrita en HEVC WD7. En un ejemplo, la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden estar configuradas para asignar contextos a **significant_coeff_group_flag** mediante la misma obtención de contexto para todos los subbloques, independientemente del tipo de barrido y el tamaño de las TU asociadas con el subbloque. Sin embargo, en un ejemplo, el contexto real asignado a **significant_coeff_group_flag** puede diferir en función de si un subbloque se lee mediante barrido diagonal, horizontal y vertical de subbloque 4x4. En un ejemplo, se puede utilizar un primer conjunto de contextos para asignar el contexto a **significant_coeff_group_flag** cuando se aplica el barrido diagonal 4x4 y se puede utilizar un segundo conjunto de contextos para asignar contextos a **significant_coeff_group_flag** cuando se aplican los barridos horizontal o vertical de subbloque 4x4. En un ejemplo, el segundo conjunto de contextos puede obtenerse añadiendo un desplazamiento al primer conjunto de contextos.

Por lo tanto, hay varias técnicas que la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 utilizan para asignar contextos a los elementos sintácticos **significant_coeff_group_flag** y **significant_coefficient_flag**. La unidad de codificación de entropía 56 puede estar configurada para asignar contextos a los elementos sintácticos **significant_coeff_group_flag** y **significant_coefficient_flag** mediante cualquier combinación de las técnicas descritas anteriormente.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de codificación de datos de vídeo de acuerdo con las

técnicas de esta divulgación. Aunque el proceso de la FIG. 11 se describe a continuación como si se llevara a cabo en general mediante el codificador de vídeo 20, el proceso puede llevarse a cabo mediante cualquier combinación formada por el codificador de vídeo 20, la unidad codificación de entropía 56 y/o la unidad de modelización de contextos 506.

5 Como se ilustra en la FIG. 11, el codificador de vídeo 20 genera elementos sintácticos de significancia para los coeficientes de transformada de un subbloque actual de un bloque (1102). El elemento sintáctico de significancia (por ejemplo, indicador de coeficiente de significancia) de un coeficiente de transformada indica si el valor del coeficiente de transformada es cero (es decir, un coeficiente de transformada cero) o distinto de cero (es decir, un coeficiente de transformada distinto de cero). En algunos ejemplos, el subbloque es un subbloque 4x4, y el bloque es un bloque de transformada 8x8.

15 El codificador de vídeo 20 selecciona un patrón de contextos de una misma pluralidad de patrones de contextos bidimensionales para una pluralidad de tipos de barrido (por ejemplo, un barrido horizontal, un barrido vertical y un barrido diagonal) para los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual (1104). Los ejemplos de patrones de contextos incluyen los patrones de contextos ilustrados en las FIGS. 9A-9D. En las técnicas descritas en esta divulgación, el codificador de vídeo 20 puede seleccionar a partir de la misma pluralidad de patrones de contextos bidimensionales si el subbloque se lee mediante barrido horizontal, vertical o diagonal. En otras palabras, cuando el tipo de barrido es horizontal o vertical, el codificador de vídeo 20 selecciona entre la misma pluralidad de patrones de contextos bidimensionales a partir de la cual selecciona el codificador de vídeo 20 si el tipo de barrido es diagonal.

25 Además, como se ha descrito anteriormente, cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más subbloques vecinos. Por ejemplo, el uno o más subbloques vecinos incluyen un primer subbloque vecino y un segundo subbloque vecino, y cada uno de los patrones de contextos puede estar asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en el primer subbloque vecino y la inclusión o no de algún coeficiente de transformada distinto de cero en el segundo subbloque vecino (es decir, cada contexto está asociado a una condición de si el valor de CGF_B y CGF_R es 1 o 0).

30 El codificador de vídeo 20 asigna contextos a cada uno de los elementos sintácticos de significancia del coeficiente de transformada basándose en el patrón de contextos seleccionado (1106). Por ejemplo, como se ha descrito anteriormente, si el codificador de vídeo 20 selecciona el patrón de contextos asociado a la condición de que CGF_B es igual a 0 y CGF_R es igual a 1 (es decir, el subbloque inferior no incluye ningún coeficiente de transformada distinto de cero), entonces el codificador de vídeo 20 asigna un contexto (por ejemplo, contexto 2 o 5 con desplazamiento de 3) a una primera fila de los elementos sintácticos de significancia del subbloque, que es diferente del contexto para las otras filas.

40 Si el codificador de vídeo 20 selecciona el patrón de contextos asociado a la condición de que CGF_B es igual a 1 y CGF_R es igual a 0 (es decir, el subbloque inferior incluye al menos un coeficiente de transformada distinto de cero y el bloque derecho no incluye ningún coeficiente de transformada distinto de cero), entonces el codificador de vídeo 20 asigna un contexto (por ejemplo, el contexto 2 o 5 con desplazamiento de 3) a una columna de los elementos sintácticos de significancia del subbloque, que es diferente del contexto para las otras columnas. Si el codificador de vídeo 20 selecciona el patrón de contextos asociado a la condición de que CGF_B es igual a 1 y CGF_R es igual a 1 (es decir, el subbloque inferior incluye al menos un coeficiente de transformada distinto de cero y el bloque derecho incluye al menos un coeficiente de transformada distinto de cero), entonces el codificador de vídeo 20 asigna un mismo contexto (por ejemplo, el contexto 2 o 5 con el desplazamiento de 3) a los elementos sintácticos de significancia del subbloque actual.

50 El codificador de vídeo 20 codifica mediante CABAC los elementos sintácticos de significancia basándose en los contextos asignados (1108). El codificador de vídeo 20 facilita los elementos sintácticos de significancia codificados como parte del flujo de bits codificado (1110).

55 La FIG. 12 es un diagrama de flujo que ilustra un ejemplo de codificación de datos de vídeo de acuerdo con las técnicas de esta divulgación. Aunque el proceso de la FIG. 12 se describe a continuación como si se llevara a cabo en general mediante el codificador de vídeo 20, el proceso puede llevarse a cabo mediante cualquier combinación formada por el codificador de vídeo 20, la unidad codificación de entropía 56 y/o la unidad de modelización de contextos 506.

60 Como se ilustra en la FIG. 12, el codificador de vídeo 20 genera elementos sintácticos de significancia para los coeficientes de transformada de un subbloque actual de un bloque (1202). El elemento sintáctico de significancia (por ejemplo, indicador de coeficiente de significancia) de un coeficiente de transformada indica si el valor del coeficiente de transformada es cero (es decir, un coeficiente de transformada cero) o distinto de cero (es decir, un coeficiente de transformada distinto de cero). En algunos ejemplos, el subbloque es un subbloque 4x4, y el bloque es un bloque de transformada 8x8.

65

El codificador de vídeo 20 selecciona un patrón de contextos (por ejemplo, uno que se almacena como patrón de contextos unidimensional, 1204). En algunos ejemplos, el patrón de contextos identifica contextos para dos o más tipos de barrido del subbloque actual. Por ejemplo, el patrón de contextos seleccionado es para un tipo de barrido entre un barrido horizontal, un barrido vertical y un barrido diagonal.

5 En un ejemplo, el patrón de contextos seleccionado define un primer contexto para un primer elemento sintáctico de significancia en un orden de barrido, define un segundo contexto para un segundo y un tercer elementos sintácticos de significancia en el orden de barrido y define un tercer contexto para los elementos sintácticos de significancia restantes en el orden de barrido. En otro ejemplo, el patrón de contextos seleccionado define un primer contexto para un primer elemento sintáctico de significancia en un orden de barrido y define un segundo contexto para los elementos sintácticos de significancia restantes en el orden de barrido. En otro ejemplo, el patrón de contextos seleccionado define un mismo contexto para todos los elementos sintácticos de significancia.

15 En algunos ejemplos, el patrón de contextos seleccionado se selecciona a partir de una pluralidad de patrones de contextos que se almacenan como patrones de contextos unidimensionales. Por ejemplo, la pluralidad de patrones de contextos se precaculan y almacenan a partir de los patrones de contextos bidimensionales ilustrados en las FIGS. 9A-9D. En un ejemplo, uno de la pluralidad de patrones de contextos se precacula a partir de un patrón de contextos bidimensional que incluye contextos que, si se leen mediante barrido horizontal o vertical, dan como resultado un mismo patrón de contextos unidimensional. Un ejemplo de dicho patrón de contextos bidimensional es el patrón de contextos ilustrado en la Fig. 9A. En otro ejemplo, uno de la pluralidad de patrones de contextos se precacula a partir de un patrón de contextos bidimensional que incluye contextos que, si se leen mediante barrido horizontal, vertical o diagonal, dan todos como resultado el mismo patrón de contextos unidimensional. Un ejemplo de dicho patrón de contextos bidimensional es el patrón de contextos ilustrado en la FIG. 9D.

25 En otro ejemplo, uno de la pluralidad de patrones de contextos se precacula a partir de un primer patrón de contextos bidimensional que incluye contextos que, si se leen mediante barrido horizontal, dan como resultado un mismo patrón de contextos unidimensional que, cuando se lee mediante barrido vertical, un segundo patrón de contextos bidimensional. Un ejemplo del primer patrón de contextos bidimensional es el patrón de contextos ilustrado en la FIG. 9B. Un ejemplo del segundo patrón de contextos bidimensional es el patrón de contextos ilustrado en la FIG. 9C.

El codificador de vídeo 20 asigna contextos a elementos sintácticos de significancia basándose en el contexto seleccionado (1206). El codificador de vídeo 20 codifica mediante CABAC los elementos sintácticos de significancia basándose en los contextos asignados (1208). El codificador de vídeo 20 facilita los elementos sintácticos de significancia codificados como parte del flujo de bits codificado (1210).

La FIG. 13 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede implementar las técnicas descritas en esta divulgación. En el ejemplo de la FIG. 13, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 80, una unidad de procesamiento de predicción 81, una unidad de procesamiento de cuantificación inversa 86, una unidad de procesamiento de transformada inversa 88, un sumador 90 y una memoria de imágenes de referencia 92. La unidad de procesamiento de predicción 81 incluye la unidad de compensación de movimiento 82 y el módulo de predicción intra 84. En algunos ejemplos, el decodificador de vídeo 30 puede realizar un pase de decodificación que en general es recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 de la FIG. 7.

45 Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un segmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación de entropía 80 del decodificador de vídeo 30 realiza la decodificación de entropía del flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de decodificación de entropía 80 envía los vectores de movimiento y otros elementos sintácticos al módulo de predicción 81. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel del segmento de vídeo y/o el nivel del bloque de vídeo.

La FIG. 14 es un diagrama de bloques que ilustra un ejemplo de unidad de decodificación de entropía 70 que puede implementar las técnicas descritas en esta divulgación. La unidad de decodificación de entropía 70 recibe un flujo de bits sometido a codificación de entropía y decodifica los elementos sintácticos del flujo de bits. Los elementos sintácticos pueden incluir los elementos sintácticos **significant_coeff_group_flag**, **significant_coefficient_flag**, **coeff_abs_level_remain**, **coeff_abs_level_greater1_flag**, **coeff_abs_level_greater2_flag** y **coeff_sign_flag**, descritos anteriormente. El ejemplo de unidad de decodificación de entropía 70 de la FIG. 14 incluye una unidad de decodificación aritmética 702, que puede incluir un motor de decodificación de derivación 704 y un motor de decodificación convencional 706. El ejemplo de unidad de decodificación de entropía 70 incluye además una unidad de modelización de contextos 708 y una unidad de binarización inversa 710. El ejemplo de unidad de decodificación de entropía 70 puede llevar a cabo las funciones recíprocas a las del ejemplo de unidad de codificación de entropía 56 descritas con respecto a la FIG. 8. De esta manera, la unidad de decodificación de entropía 70 puede llevar a cabo la decodificación de entropía basándose en las técnicas descritas en el presente documento.

La unidad de decodificación aritmética 702 recibe un flujo de bits codificado. Como se muestra en la FIG. 14, la unidad de decodificación aritmética 702 puede procesar valores de *bin* codificados de acuerdo con una ruta de derivación o la ruta de codificación convencional. Una indicación sobre si un valor de *bin* codificado debería procesarse de acuerdo con una ruta de derivación o una ruta convencional puede señalizarse en el flujo de bits con

5 sintaxis de nivel superior. En consonancia con el proceso CABAC descrito anteriormente, en caso de que la unidad de decodificación aritmética 702 reciba valores de *bin* desde una ruta de derivación, el motor de decodificación de derivación 704 puede llevar a cabo una codificación aritmética de los valores de *bin*, sin utilizar un contexto asignado a un valor de *bin*. En un ejemplo, el motor de decodificación de derivación 704 puede suponer las mismas probabilidades para los posibles valores de un *bin*.

10 En caso de que la unidad de decodificación aritmética 702 reciba valores de *bin* a través de la ruta convencional, la unidad de modelización de contextos 708 puede proporcionar una variable de contexto, de manera que el motor de decodificación convencional 706 pueda llevar a cabo una codificación aritmética basándose en las asignaciones de contexto proporcionadas por la unidad de modelización de contextos 708. Las asignaciones de contexto pueden definirse según una norma de codificación de vídeo, tal como la HEVC. Los modelos de contexto pueden almacenarse en memoria. La unidad de modelización de contextos 708 puede incluir una serie de tablas indexadas y/o utilizar funciones de correlación para determinar un contexto y una parte de variable de contexto de un flujo de bits codificado. Además, en un ejemplo, la unidad de modelización de contextos 506 y/o la unidad de codificación de entropía 56 pueden configurarse para asignar contextos a *bins* de los elementos sintácticos

15 **significant_coefr_group_flag** y **significant_coefficient_flag** basándose en las técnicas descritas en el presente documento. Tras decodificar un valor de *bin*, el motor de codificación convencional 706 puede actualizar un contexto basándose en los valores de *bin* decodificados. Además, la unidad de binarización inversa 710 puede llevar a cabo una binarización inversa en un valor de *bin* y utilizar una función de correspondencia de *bins* para determinar si un valor de *bin* es válido. La unidad de binarización inversa 710 también puede actualizar la unidad de modelización de contextos basándose en la determinación de correspondencia. Por tanto, la unidad de binarización inversa 710

20 **facilita** elementos sintácticos de acuerdo con una técnica de decodificación adaptativa según el contexto.

30 Cuando el segmento de vídeo se codifica como un segmento intracodificado (I), el módulo de predicción intra 84 del módulo de predicción 81 puede generar datos de predicción para un bloque de vídeo del segmento de vídeo actual, basándose en un modo de predicción intra señalizado y datos de bloques decodificados previamente de la trama o imagen actual. Cuando la trama de vídeo se codifica como un segmento intercodificado (es decir, B o P), la unidad de compensación de movimiento 82 del módulo de predicción 81 genera bloques predictivos para un bloque de vídeo del segmento de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de decodificación de entropía 80. Los bloques predictivos pueden generarse a partir de una de las imágenes de referencia de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede crear las listas de tramas de referencia, la Lista 0 y la Lista 1, mediante técnicas de construcción predeterminadas, basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

40 La unidad de compensación de movimiento 82 determina información de predicción para un bloque de vídeo del segmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y utiliza la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 82 utiliza algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, predicción intra o predicción inter)

45 utilizado para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento de predicción inter (por ejemplo, segmento B o segmento P), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de vídeo intercodificado del segmento, el estado de predicción inter para cada bloque de vídeo intercodificado del segmento y otra información para decodificar los bloques de vídeo del segmento de vídeo actual.

50 La unidad de compensación de movimiento 82 también puede llevar a cabo una interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 82 puede utilizar filtros de interpolación como los utilizados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos, y utilizar los filtros de interpolación para generar bloques predictivos.

60 La unidad de procesamiento de cuantificación inversa 86 realiza la cuantificación inversa, es decir, descuantifica, los coeficientes de transformada cuantificados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación de entropía 80. El proceso de cuantificación inversa puede incluir la utilización de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del segmento de vídeo, a fin de determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse. La unidad de procesamiento de transformada inversa 88 aplica una transformada inversa, por ejemplo una DCT inversa, una transformada entera inversa o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada, con el fin de generar bloques residuales en el dominio del píxel.

65

En algunos casos, la unidad de procesamiento de transformada inversa 88 puede aplicar una transformada inversa bidimensional (2D, tanto en la dirección horizontal como la vertical) a los coeficientes. De acuerdo con las técnicas de esta divulgación, la unidad de procesamiento de transformada inversa 88 puede aplicar, en cambio, una transformada inversa 1D horizontal, una transformada inversa 1D vertical o ninguna transformada a los datos residuales de cada una de las TU. El tipo de transformada aplicada a los datos residuales en el codificador de vídeo 20 puede señalizarse al decodificador de vídeo 30 para aplicar un tipo apropiado de transformada inversa a los coeficientes de transformada.

Después de que la unidad de compensación de movimiento 82 genere el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales de la unidad de procesamiento de transformada inversa 88 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de eliminación de bloques para filtrar los bloques decodificados con el fin de eliminar distorsiones de efecto pixelado. También pueden utilizarse otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo decodificados de una trama o imagen dada se almacenan, a continuación, en la memoria de imágenes de referencia 92, que almacena imágenes de referencia utilizadas para una subsiguiente compensación de movimiento. La memoria de imágenes de referencia 92 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 34 de la FIG. 6.

La FIG. 15 es un diagrama de flujo que ilustra un ejemplo de decodificación de datos de vídeo de acuerdo con las técnicas de esta divulgación. Aunque el proceso de la FIG. 15 se describe a continuación como si se llevara a cabo en general mediante el decodificador de vídeo 30, el proceso puede llevarse a cabo mediante cualquier combinación del decodificador de vídeo 30, la unidad de decodificación de entropía 56 y/o la unidad de modelización de contextos 708.

Como se ilustra en la FIG. 15, el decodificador de vídeo 30 recibe, en un flujo de bits sometido a codificación de entropía, elementos sintácticos de significancia de coeficientes de transformada para un subbloque actual de un bloque (1502). El subbloque puede ser un subbloque 4x4, y el bloque puede ser un bloque de transformada 8x8. De forma similar al codificador de vídeo 20 (por ejemplo, el bloque 1104 de la FIG. 11), el decodificador de vídeo 30 selecciona un patrón de contextos a partir de una misma pluralidad de patrones de contextos de patrones de contextos bidimensionales para una pluralidad de tipos de barrido (por ejemplo, un barrido horizontal, un barrido vertical y un barrido diagonal) para los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual (1504). En este ejemplo, cada uno de los patrones de contextos está asociado a una condición sobre la inclusión o no de algún coeficiente de transformada distinto de cero en uno o más bloques vecinos (por ejemplo, un primer subbloque vecino y un segundo bloque vecino).

El decodificador de vídeo 30, de una manera similar a la descrita anteriormente con respecto al codificador de vídeo 20 (por ejemplo, el bloque 1106 de la FIG. 11), asigna contextos a cada uno de los elementos sintácticos de significancia de los coeficientes de transformada basándose en el patrón de contextos seleccionado (1506). Por ejemplo, si se selecciona el patrón de contextos asociado a la condición de que CGF_B es igual a 0 y CGF_R es igual a 1, entonces el decodificador de vídeo 30 asigna un contexto a una primera fila, que es diferente de los contextos para las otras filas. Si se selecciona el patrón de contextos asociado a la condición de que CGF_B es igual a 1 y CGF_R es igual a 0, entonces el decodificador de vídeo 30 asigna un contexto a una primera columna, que es diferente de los contextos para las otras columnas. Si se selecciona el patrón de contextos asociado a la condición de que CGF_B es igual a 1 y CGF_R es igual a 1, entonces el decodificador de vídeo 30 asigna un mismo contexto a los elementos sintácticos de significancia. El decodificador de vídeo 30 decodifica mediante CABAC los elementos sintácticos de significancia basándose en los contextos asignados (1508).

La FIG. 16 es un diagrama de flujo que ilustra un ejemplo de decodificación de datos de vídeo de acuerdo con las técnicas de esta divulgación. Aunque el proceso de la FIG. 16 se describe a continuación como si se llevara a cabo en general mediante el decodificador de vídeo 30, el proceso puede llevarse a cabo mediante cualquier combinación del decodificador de vídeo 30, la unidad de decodificación de entropía 70 y/o la unidad de modelización de contextos 708.

Como se ilustra en la FIG. 16, el decodificador de vídeo 30 recibe, en un flujo de bits sometido a codificación de entropía, elementos sintácticos de significancia de coeficientes de transformada para un subbloque actual de un bloque (1602). El subbloque puede ser un subbloque 4x4, y el bloque puede ser un bloque de transformada 8x8. De forma similar al codificador de vídeo 20 (por ejemplo, el bloque 1204 de la FIG. 12), el decodificador de vídeo 30 selecciona un patrón de contextos (por ejemplo, uno que se almacena como un patrón de contextos unidimensional, 1604). El patrón de contextos puede ser para dos o más tipos de barrido (por ejemplo, los tipos de barrido horizontal, diagonal y vertical).

En un ejemplo, tal como se ha descrito anteriormente, el patrón de contextos seleccionado define un primer contexto

para un primer elemento sintáctico de significancia en un orden de barrido, define un segundo contexto para un segundo y un tercer elementos sintácticos de significancia en el orden de barrido y define un tercer contexto para los elementos sintácticos de significancia restantes en el orden de barrido. En otro ejemplo, tal como se ha descrito anteriormente, el patrón de contextos seleccionado define un primer contexto para un primer elemento sintáctico de significancia en un orden de barrido y define un segundo contexto para los elementos sintácticos de significancia restantes en el orden de barrido. En otro ejemplo, el patrón de contextos seleccionado define un mismo contexto para todos los elementos sintácticos de significancia.

En algunos ejemplos, el patrón de contextos seleccionado se selecciona a partir de una pluralidad de patrones de contextos que se almacenan como patrones de contextos unidimensionales. Por ejemplo, la pluralidad de patrones de contextos se precaculan y almacenan a partir de los patrones de contextos bidimensionales ilustrados en las FIGS. 9A-9D. En un ejemplo, uno de la pluralidad de patrones de contextos se precacula a partir de un patrón de contextos bidimensional que incluye contextos que, si se leen mediante barrido horizontal o vertical, dan como resultado un mismo patrón de contextos unidimensional. Un ejemplo de dicho patrón de contextos bidimensional es el patrón de contextos ilustrado en la Fig. 9A. En otro ejemplo, uno de la pluralidad de patrones de contextos se precacula a partir de un patrón de contextos bidimensional que incluye contextos que, si se leen mediante barrido horizontal, vertical o diagonal, dan todos como resultado el mismo patrón de contextos unidimensional. Un ejemplo de dicho patrón de contextos bidimensional es el patrón de contextos ilustrado en la Fig. 9D.

En otro ejemplo, uno de la pluralidad de patrones de contextos se precacula a partir de un primer patrón de contextos bidimensional que incluye contextos que, si se leen mediante barrido horizontal, dan como resultado un mismo patrón de contextos unidimensional que, cuando se lee mediante barrido vertical, un segundo patrón de contextos bidimensional. Un ejemplo del primer patrón de contextos bidimensional es el patrón de contextos ilustrado en la FIG. 9B. Un ejemplo del segundo patrón de contextos bidimensional es el patrón de contextos ilustrado en la FIG. 9C.

El decodificador de vídeo 30 asigna contextos a elementos sintácticos de significancia basándose en el contexto seleccionado (1606). El decodificador de vídeo 30 decodifica mediante CABAC los elementos sintácticos de significancia basándose en los contextos asignados (1608).

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse como una o más instrucciones o código en un medio legible por ordenador, o transmitirse a través de este, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) unos medios de almacenamiento tangibles y legibles por ordenador que sean no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de manera limitativa, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda utilizarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión puede denominarse de forma correcta "medio legible por ordenador". Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota mediante un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, en el presente documento, incluyen un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, de los cuales los discos flexibles normalmente reproducen los datos magnéticamente, mientras que los otros discos reproducen ópticamente los datos mediante láser. Las combinaciones de los anteriores también deberían incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables *in situ* (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador" en el presente documento puede referirse a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento.

Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse por completo en uno o más circuitos o elementos lógicos.

- 5 Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). En esta divulgación se describen varios componentes, módulos o unidades para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero su realización mediante diferentes unidades de hardware no es necesaria. En su lugar, como se ha descrito anteriormente, diversas unidades pueden combinarse
- 10 en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuados.

- 15 Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

- 5 1. Un procedimiento para decodificar datos de vídeo, comprendiendo el procedimiento:

 recibir (1502, 1602), en un flujo de bits, elementos sintácticos de significancia que indican si cada coeficiente de transformada correspondiente de coeficientes de transformada de un subbloque actual de un bloque es un coeficiente de transformada distinto de cero;

10 seleccionar (1504, 1604) un patrón de contextos, que incluye valores de contextos que corresponden a los coeficientes de transformada situados en posiciones respectivas del subbloque, a partir de una pluralidad de patrones de contextos, en el que el patrón de contextos seleccionado identifica contextos para dos o más tipos de barrido del subbloque actual, en el que un tipo de barrido es un proceso de lectura de elementos de una matriz bidimensional para obtener una matriz unidimensional;

15 asignar (1506, 1606) valores de contextos del patrón de contextos seleccionado a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en posiciones de los valores de contextos en el patrón de contextos seleccionado y posiciones de los coeficientes de transformada correspondientes a elementos sintácticos significativos; y

 decodificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) (1508, 1608) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual mediante unos valores de contextos asignados respectivos.

20
2. Un procedimiento para codificar datos de vídeo, comprendiendo el procedimiento:

 generar (1102, 1202) elementos sintácticos de significancia que indican si cada coeficiente de transformada correspondiente de unos coeficientes de transformada de un subbloque actual de un bloque es un coeficiente de transformada distinto de cero;

25 seleccionar (1104, 1204) un patrón de contextos, que incluye valores de contextos que corresponden a los coeficientes de transformada situados en posiciones respectivas del subbloque, a partir de una pluralidad de patrones de contextos, en el que el patrón de contextos seleccionado identifica contextos para dos o más tipos de barrido del subbloque actual, en el que un tipo de barrido es un proceso de lectura de elementos de una matriz bidimensional para obtener una matriz unidimensional;

30 asignar (1106, 1206) valores de contextos del patrón de contextos seleccionado a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en unas posiciones de los valores de contextos en el patrón de contextos seleccionado y unas posiciones de los coeficientes de transformada correspondientes a unos elementos sintácticos significativos;

35 codificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) (1108, 1208) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual mediante valores de contextos asignados respectivos; y

 facilitar (1110, 1210) los elementos sintácticos de significancia codificados.

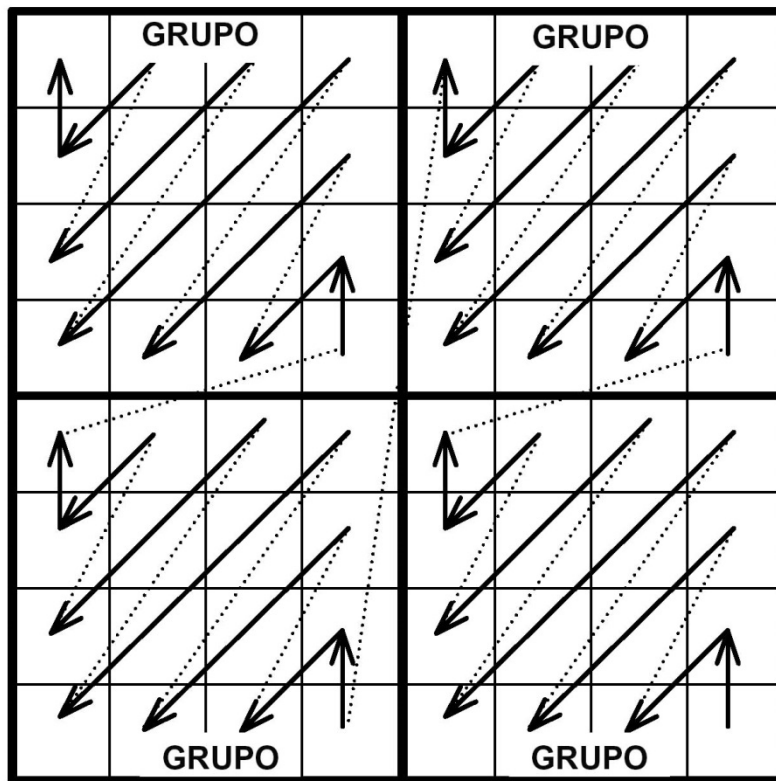
40
3. El procedimiento de la reivindicación 1, en el que seleccionar el patrón de contextos comprende seleccionar el patrón de contextos para un tipo de barrido entre un barrido horizontal, un barrido vertical y un barrido diagonal.
- 45 4. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que seleccionar el patrón de contextos comprende seleccionar el patrón de contextos que define un primer contexto para un primer elemento sintáctico de significancia en un orden de barrido, define un segundo contexto para un segundo y un tercer elemento sintácticos de significancia en el orden de barrido y define un tercer contexto para unos elementos sintácticos de significancia restantes en el orden de barrido.

50
5. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que seleccionar el patrón de contextos comprende seleccionar el patrón de contextos que define un primer contexto para un primer elemento sintáctico de significancia en un orden de barrido, y define un segundo contexto para unos elementos sintácticos de significancia restantes en el orden de barrido.

55
6. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que seleccionar el patrón de contextos comprende seleccionar el patrón de contextos que define un mismo contexto para todos los elementos sintácticos de significancia.
- 60 7. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que el subbloque actual comprende un subbloque 4x4, y el bloque comprende un bloque 8x8.
8. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que seleccionar el patrón de contextos comprende seleccionar (1204, 1604) el patrón de contextos a partir de una pluralidad de patrones de contextos que se almacenan cada uno como patrones de contextos unidimensionales.

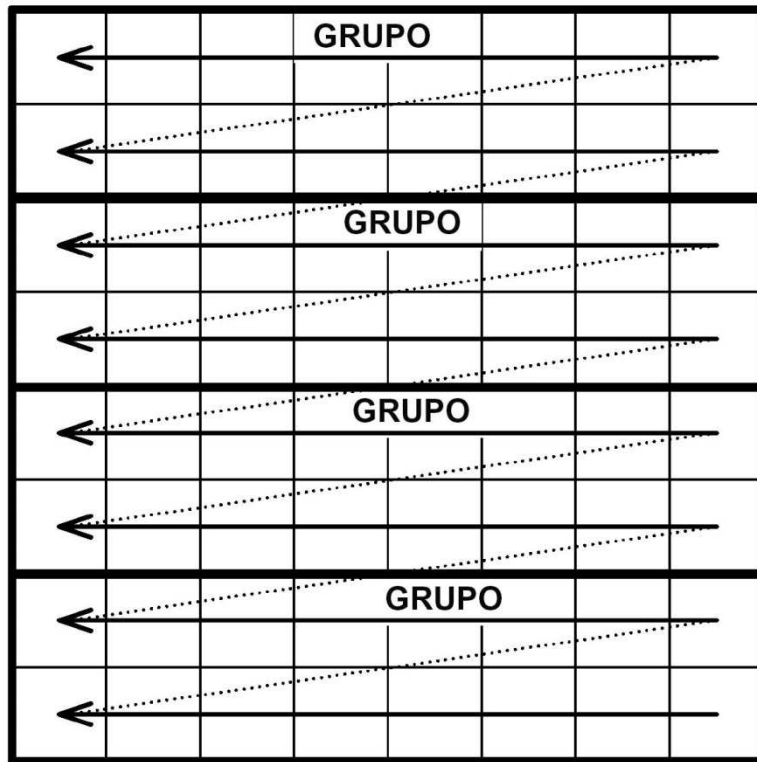
65

- 5
9. El procedimiento de la reivindicación 8, en el que uno de la pluralidad de patrones de contextos se precalcula a partir de un patrón de contextos bidimensional, y en el que el patrón de contextos bidimensional comprende contextos que si se leen mediante barrido horizontal o vertical dan como resultado un mismo patrón de contextos unidimensional.
- 10
10. El procedimiento de la reivindicación 8, en el que uno de la pluralidad de patrones de contextos se precalcula a partir de un patrón de contextos bidimensional, y en el que el patrón de contextos bidimensional comprende contextos que si se leen mediante barrido diagonal, horizontal y vertical dan como resultado un mismo patrón de contextos unidimensional.
- 15
11. El procedimiento de la reivindicación 8, en el que uno de la pluralidad de patrones de contextos se precalcula a partir de un primer patrón de contextos bidimensional, y en el que el primer patrón de contextos bidimensional comprende contextos que si se leen mediante barrido horizontal dan como resultado un mismo patrón de contextos unidimensional que cuando se lee mediante barrido vertical un segundo patrón de contextos bidimensional.
- 20
12. Un dispositivo para decodificar datos de vídeo, comprendiendo el dispositivo:
- medios para recibir, en un flujo de bits, elementos sintácticos de significancia que indican si cada coeficiente de transformada correspondiente de unos coeficientes de transformada de un subbloque actual de un bloque es un coeficiente de transformada distinto de cero;
- medios para seleccionar un patrón de contextos, que incluye valores de contextos que corresponden a los coeficientes de transformada situados en posiciones respectivas del subbloque, a partir de una pluralidad de patrones de contextos, en el que el patrón de contextos seleccionado identifica contextos para dos o
- 25
- más tipos de barrido del subbloque actual, en el que un tipo de barrido es un proceso de lectura de elementos de una matriz bidimensional para obtener una matriz unidimensional;
- medios para asignar valores de contextos del patrón de contextos seleccionado a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en unas posiciones de los valores de contextos en el patrón de contextos seleccionado y unas posiciones de
- 30
- los coeficientes de transformada correspondientes a unos elementos sintácticos significativos; y
- medios para decodificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual mediante unos valores de contextos asignados respectivos.
- 35
13. Un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo:
- medios para generar elementos sintácticos de significancia que indican si cada coeficiente de transformada correspondiente de unos coeficientes de transformada de un subbloque actual de un bloque es un coeficiente de transformada distinto de cero;
- 40
- medios para seleccionar un patrón de contextos, que incluye valores de contextos que corresponden a los coeficientes de transformada situados en posiciones respectivas del subbloque, a partir de una pluralidad de patrones de contextos, en el que el patrón de contextos seleccionado identifica contextos para dos o
- 45
- más tipos de barrido del subbloque actual, en el que un tipo de barrido es un proceso de lectura de elementos de una matriz bidimensional para obtener una matriz unidimensional;
- medios para asignar valores de contextos del patrón de contextos seleccionado a los elementos sintácticos de significancia de los coeficientes de transformada para el subbloque actual basándose en unas posiciones de los valores de contexto en el patrón de contextos seleccionado y unas posiciones de
- 50
- los coeficientes de transformada correspondientes a unos elementos sintácticos significativos;
- medios para codificar mediante codificación aritmética binaria adaptativa según el contexto (CABAC) los elementos sintácticos de significancia de los coeficientes de transformada del subbloque actual mediante unos valores de contexto asignados respectivos; y
- medios para facilitar los elementos sintácticos de significancia codificados.
- 55
14. El dispositivo de la reivindicación 12 o la reivindicación 13 que comprende medios para llevar a cabo un procedimiento de acuerdo con cualquiera de las reivindicaciones 3 a 11.
- 60
15. Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que cuando se ejecutan hacen que uno o más procesadores de un dispositivo lleven a cabo un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 11.

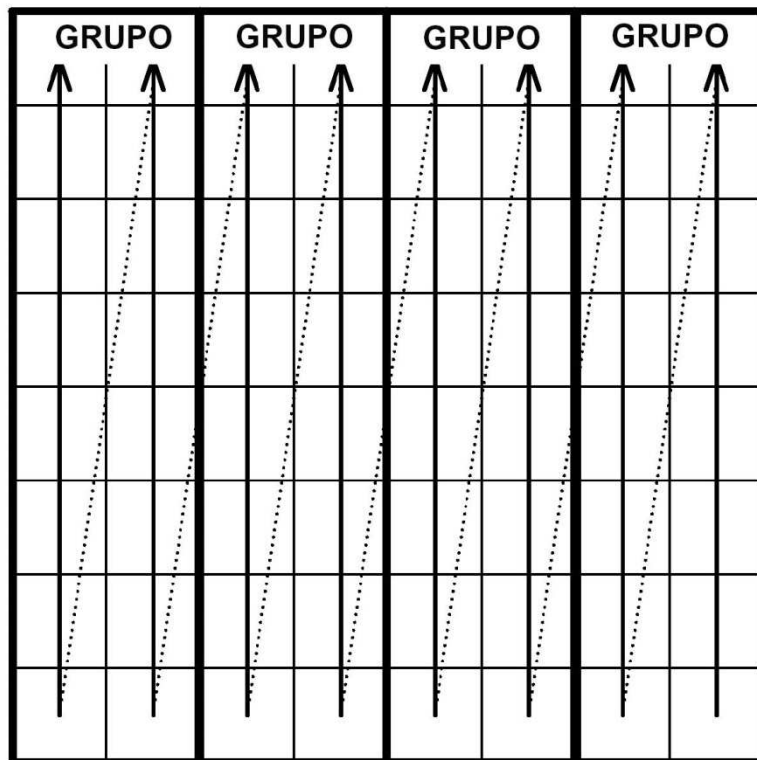


DIAGONAL DE SUBBLOQUE

FIG. 1



HORIZONTAL
FIG. 2A



VERTICAL
FIG. 2B

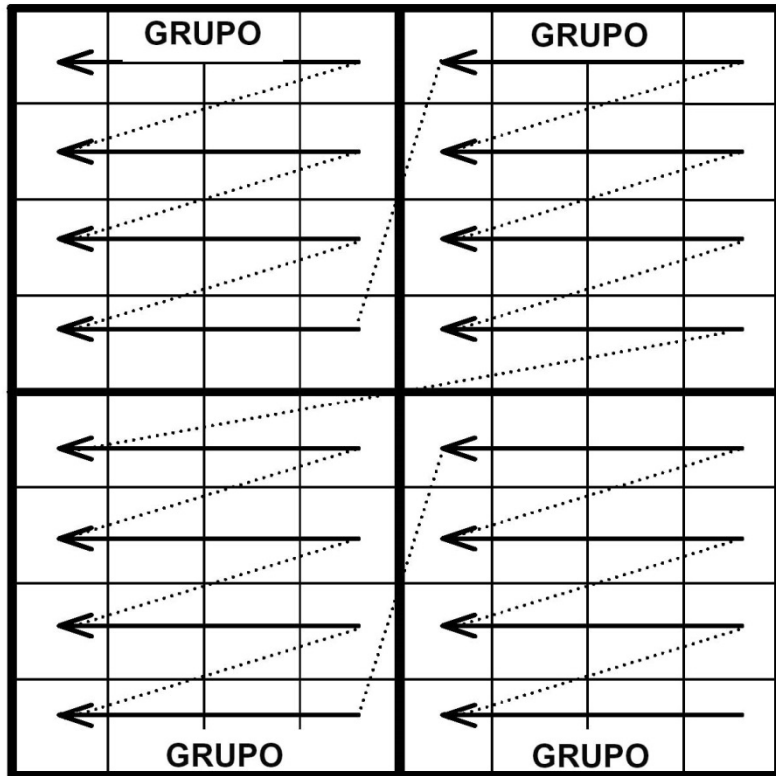


FIG. 3A

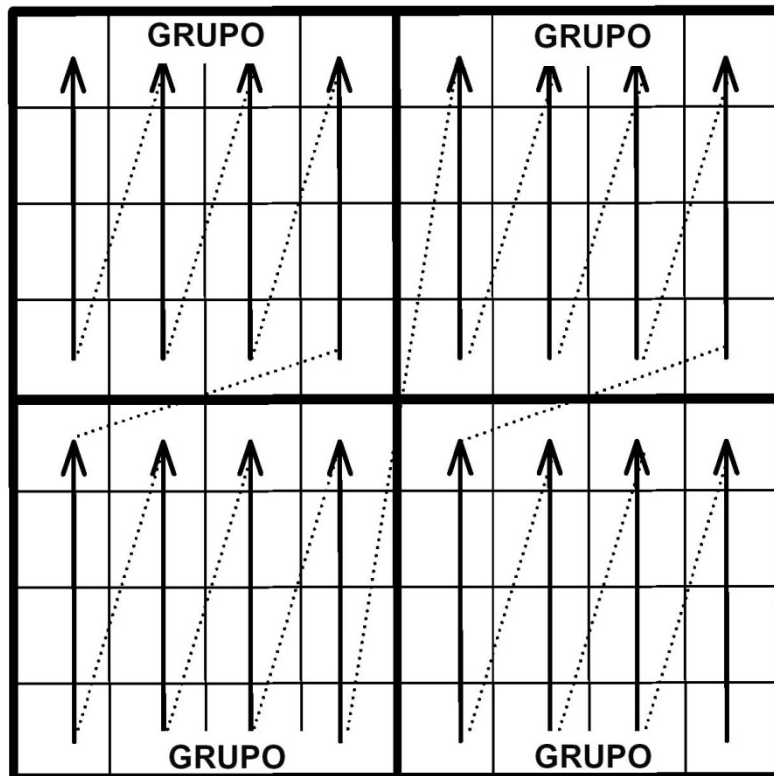


FIG. 3B

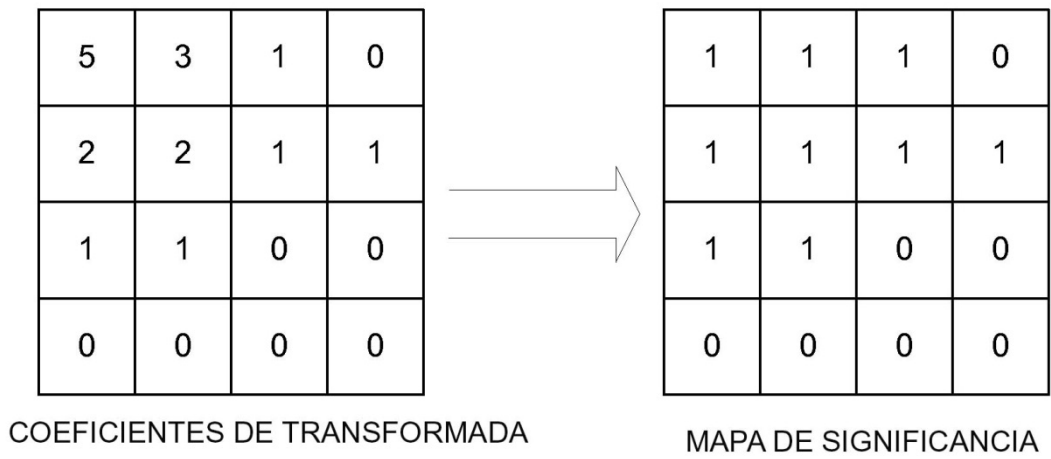


FIG. 4

CGF _B = 0, CBF _R = 0			
1	1	1	0
1	1	0	0
1	0	0	0
0	0	0	0

FIG. 5A

CGF _B = 0, CBF _R = 1			
1	1	1	1
1	1	1	1
0	0	0	0
0	0	0	0

FIG. 5B

CGF _B = 1, CBF _R = 0			
1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0

FIG. 5C

CGF _B = 1, CBF _R = 1			
2	2	2	2
2	2	2	2
2	2	2	1
2	2	1	1

FIG. 5D

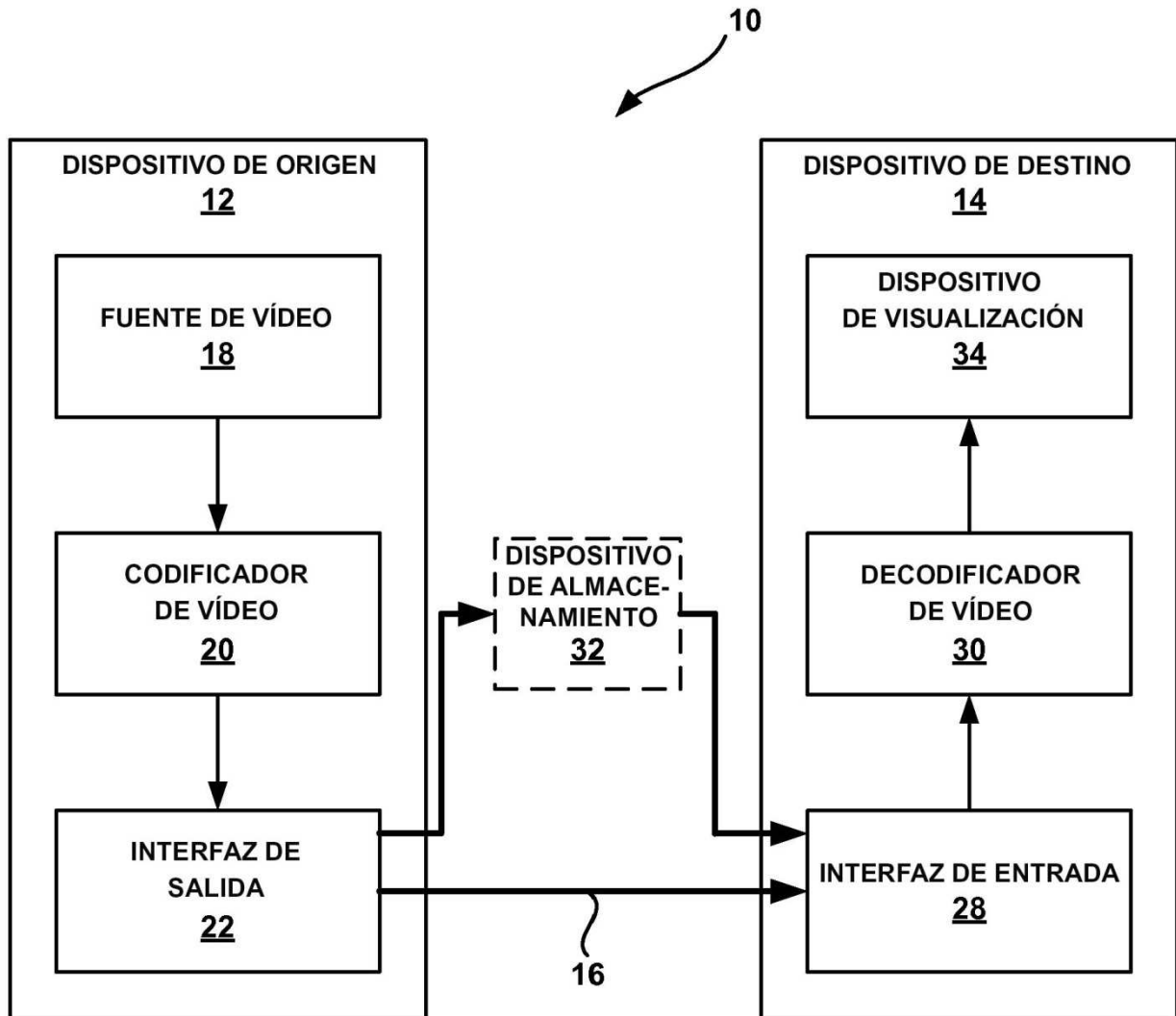


FIG. 6

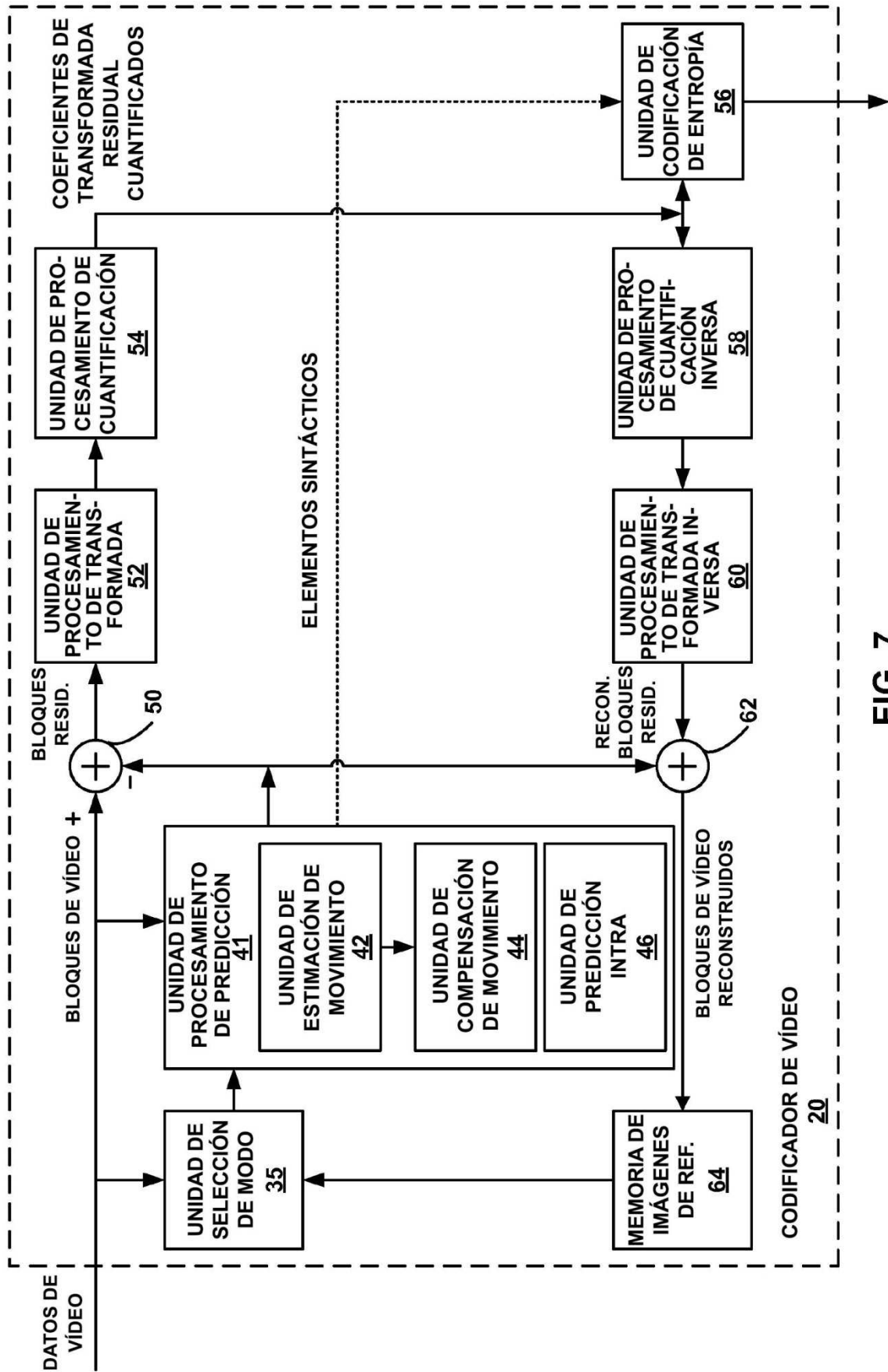


FIG. 7

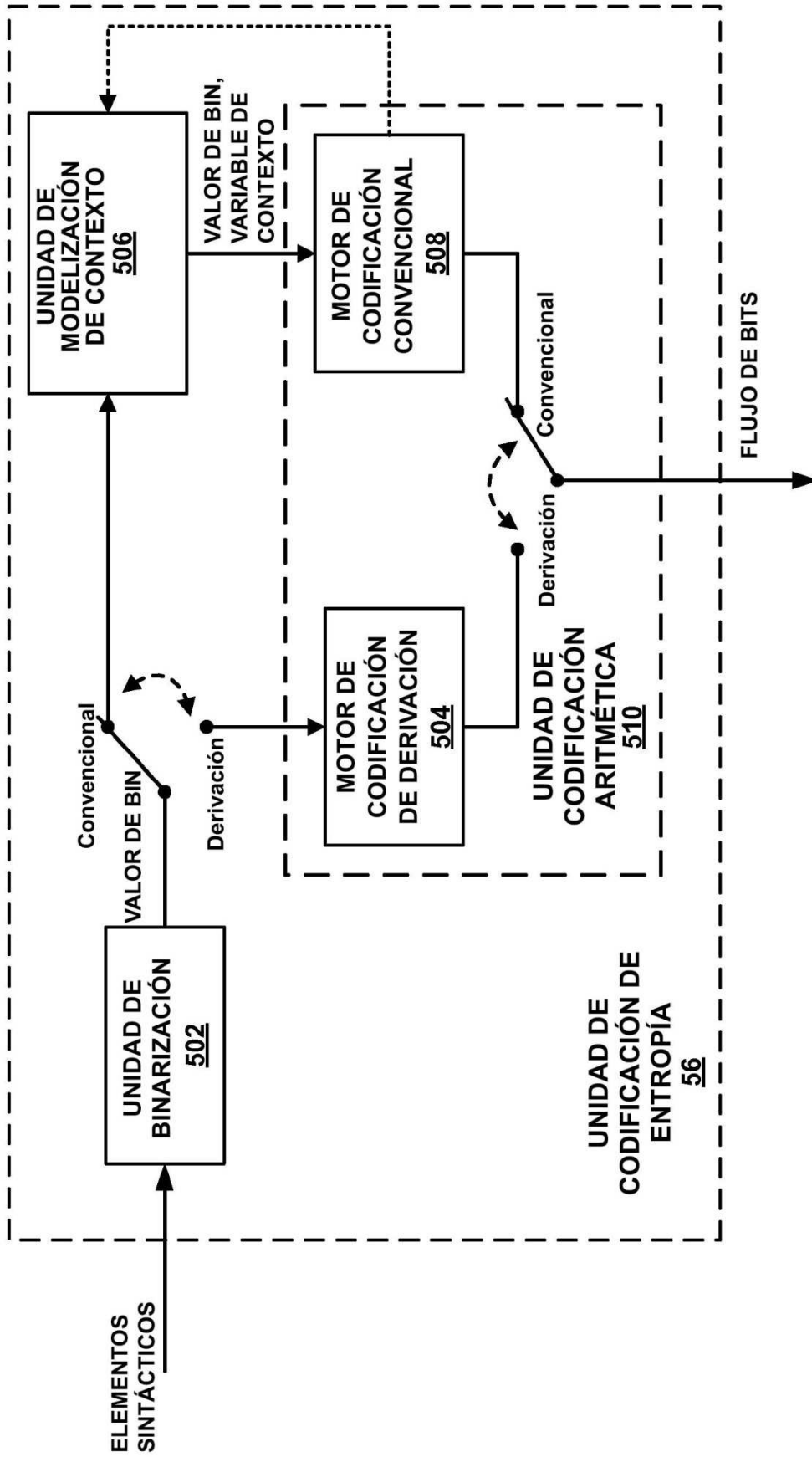


FIG. 8

CGF _B = 0, CBF _R = 0			
2	1	1	0
1	1	1	0
1	1	1	0
0	0	0	0

FIG. 9A

CGF _B = 0, CBF _R = 1			
2	2	2	2
1	1	1	1
0	0	0	0
0	0	0	0

FIG. 9B

CGF _B = 1, CBF _R = 0			
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0

FIG. 9C

CGF _B = 1, CBF _R = 1			
2	2	2	2
2	2	2	2
2	2	2	2
2	2	2	2

FIG. 9D

$CGF_B = X, CBF_R = X$			
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

FIG. 10

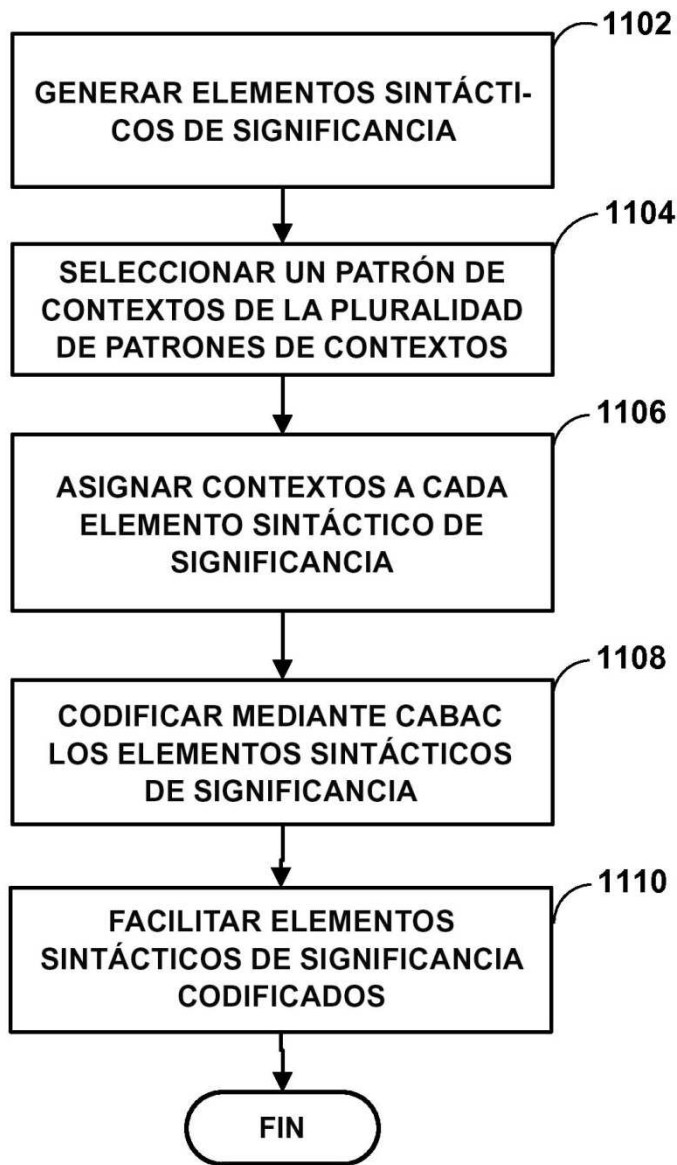


FIG. 11

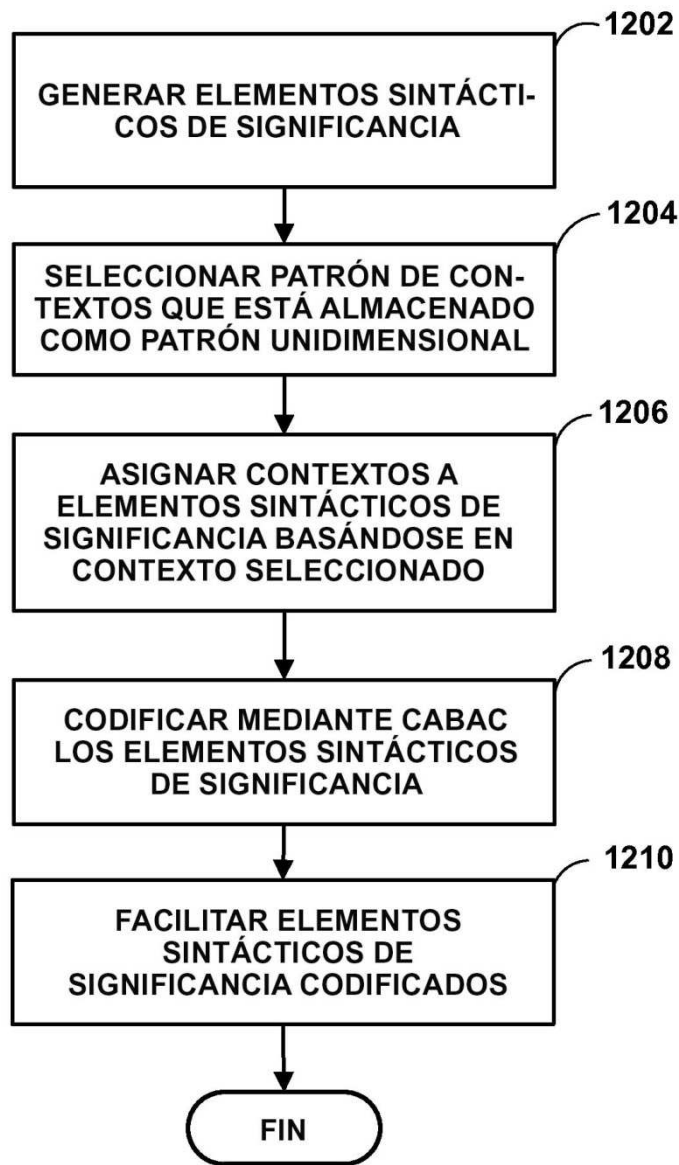


FIG. 12

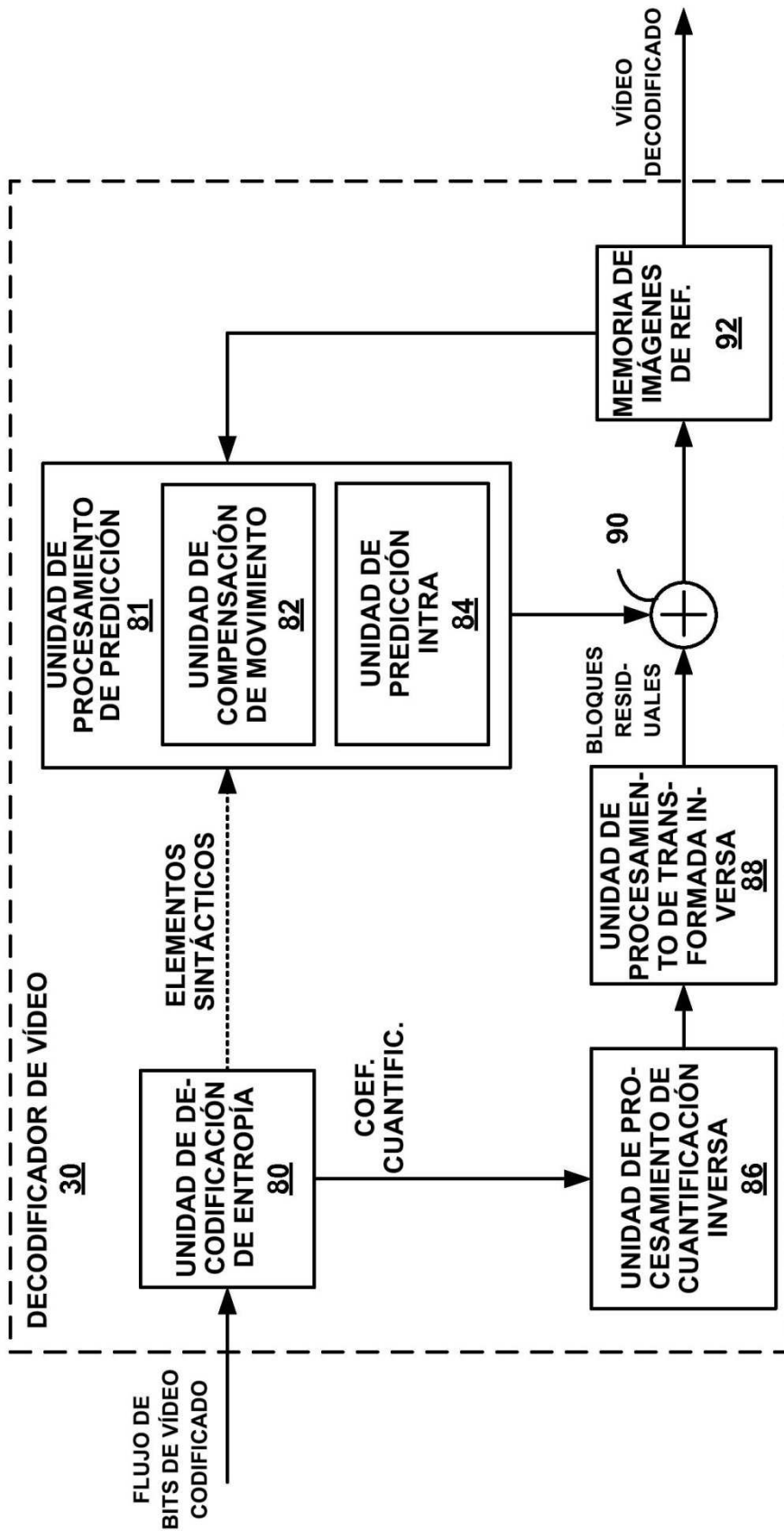


FIG. 13

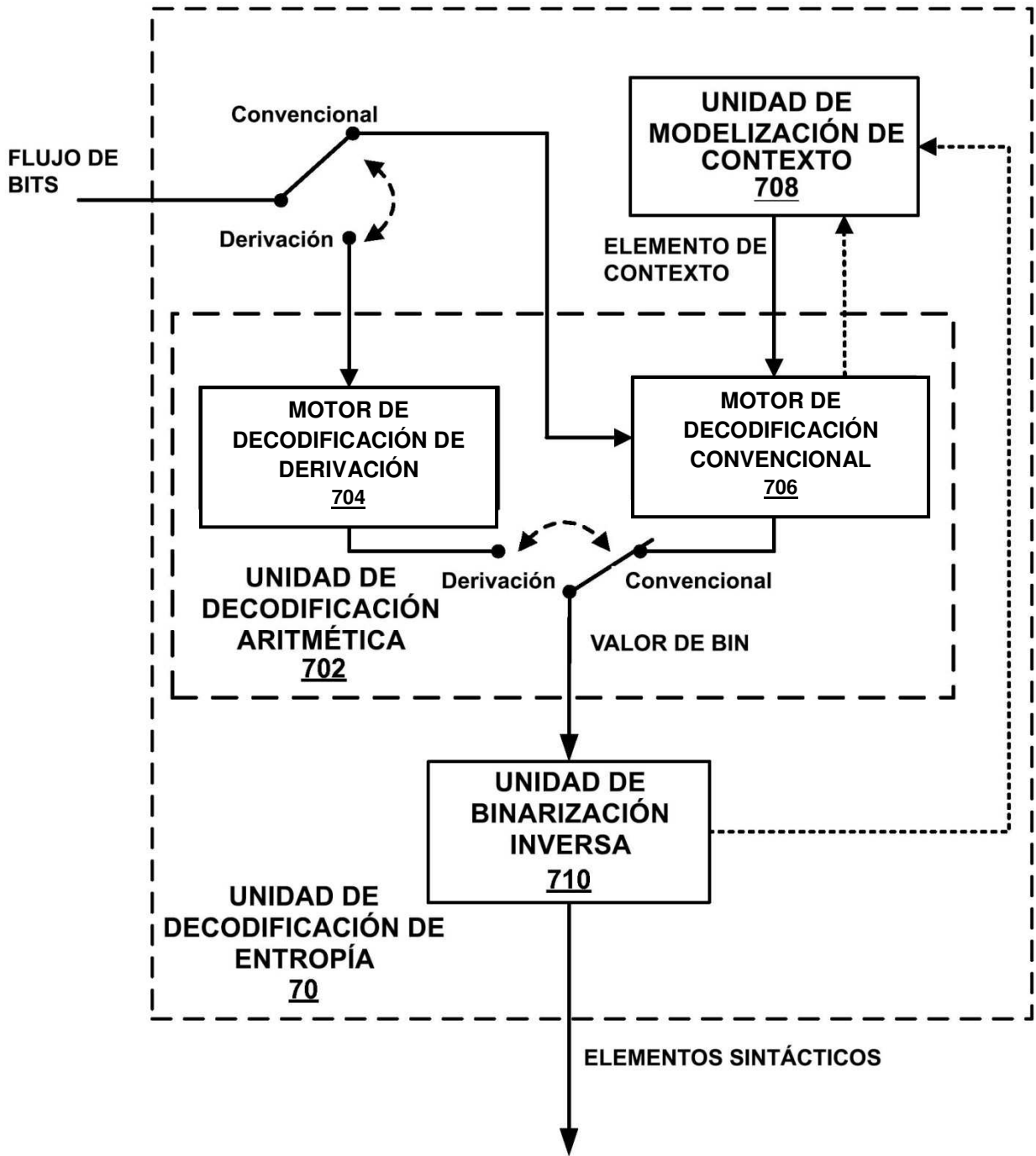


FIG. 14

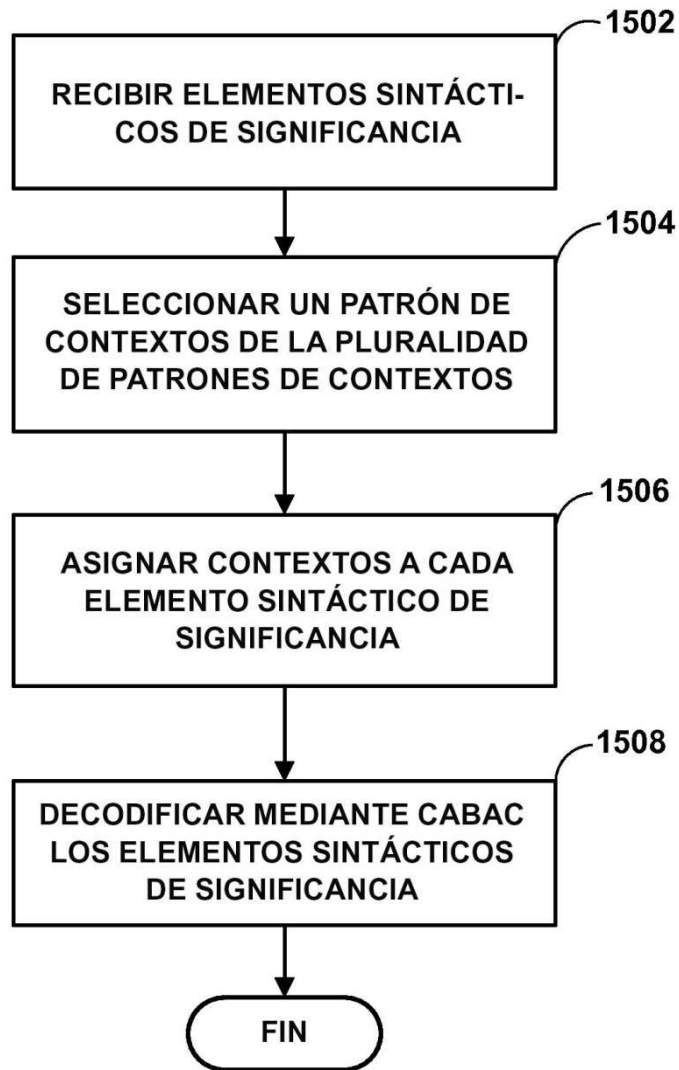


FIG. 15

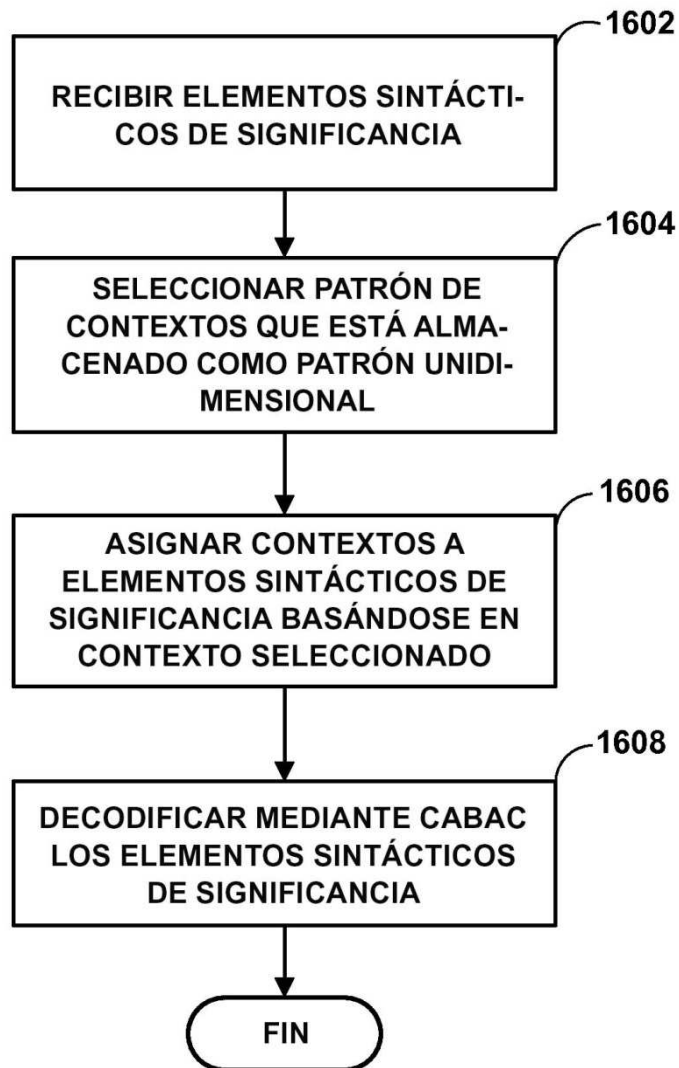


FIG. 16