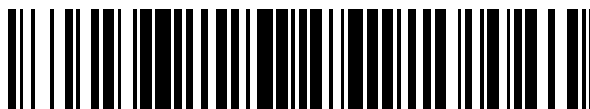


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 777 218**

51 Int. Cl.:

H04N 19/467 (2014.01)

H04N 19/593 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **25.03.2014** **PCT/US2014/031734**

87 Fecha y número de publicación internacional: **02.10.2014** **WO14160714**

96 Fecha de presentación y número de la solicitud europea: **25.03.2014** **E 14720424 (2)**

97 Fecha y número de publicación de la concesión europea: **11.12.2019** **EP 2979452**

54 Título: **Inhabilitación de ocultación de datos de signo en codificación de vídeo**

30 Prioridad:

25.03.2013 US 201361805094 P
05.04.2013 US 201361809199 P
05.04.2013 US 201361809203 P
08.04.2013 US 201361809811 P
08.04.2013 US 201361809870 P
09.04.2013 US 201361810179 P
09.04.2013 US 201361810218 P
05.07.2013 US 201361843144 P
24.03.2014 US 201414223955

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
04.08.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

JOSHI, RAJAN LAXMAN;
SOLE ROJALS, JOEL y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 777 218 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

Inhabilitación de ocultación de datos de signo en codificación de vídeo

[0001] Esta solicitud reivindica el beneficio de la solicitud de patente provisional de EE. UU. n.º 61/805,094, presentada el 25 de marzo de 2013, la solicitud de patente provisional de EE. UU. n.º 61/809,199, presentada el 5 de abril de 2013, la solicitud de patente provisional de EE. UU. n.º 61/809,203, presentada el 5 de abril de 2013, la solicitud de patente provisional de EE. UU. n.º 61/809,811, presentada el 8 de abril de 2013, la solicitud de patente provisional de EE. UU. n.º 61/809,870, presentada el 8 de abril de 2013, la solicitud de patente provisional de EE. UU. n.º 61/810,179, presentada el 9 de abril de 2013, la solicitud de patente provisional de EE. UU. n.º 61/810,218, presentada el 9 de abril de 2013 y la solicitud de patente provisional de EE. UU. n.º 61/843,144, presentada el 5 de julio de 2013.

CAMPO TÉCNICO

[0002] Esta divulgación se refiere a la codificación y la compresión de vídeo.

ANTECEDENTES

[0003] Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos de radio celulares o por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) y ampliaciones de dichas normas, para transmitir, recibir y almacenar información de vídeo digital más eficazmente.

[0004] Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca de las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un sector de vídeo se puede dividir en bloques de vídeo, que también se pueden denominar bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo de un sector intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen. Los bloques de vídeo de un sector intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen, o predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes se pueden denominar tramas, y las imágenes de referencia se pueden denominar tramas de referencia.

[0005] JOEL SOLE *ET AL*: "Transform Coefficient Coding in HEVC", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE SERVICE CENTER, PISCATAWAY, NJ, EE. UU., vol. 22, n.º 12, 1 de diciembre de 2012 (1-12-2012), páginas 1765-1777, XP011487805, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2012.2223055 divulga la aplicación de la ocultación de datos de signo (SDH) a la HEVC.

[0006] GARY J SULLIVAN *ET AL*: "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE SERVICE CENTER, PISCATAWAY, NJ, EE. UU., vol. 22, n.º 12, 1 de diciembre de 2012 (1-12-2012), páginas 1649-1668, XP011487803, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2012.2221191 describe los modos de codificación especiales de HEVC, que incluyen el modo sin pérdida y el modo de omisión de transformada.

BREVE DESCRIPCIÓN

[0007] En general, un codificador de vídeo 20 puede generar un flujo de bits que incluye una secuencia de bits que forman una representación codificada de los datos de vídeo. Como parte de la generación del flujo de bits, el codificador de vídeo puede determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando codificación con pérdida sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación por codificación de impulsos diferencial (DPCM) residual. De forma similar, un codificador de vídeo 20 puede obtener elementos de sintaxis ("syntax elements") de un flujo de bits que incluye una secuencia de bits que forman una representación codificada de los datos de vídeo. Como parte de la obtención de los elementos de sintaxis del flujo de bits, el decodificador de vídeo puede determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando codificación con pérdida sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de DPCM residual. El decodificador de vídeo puede reconstruir una imagen de los datos de vídeo en base a, al menos en parte, los elementos de sintaxis obtenidos del flujo de bits.

[0008] En un ejemplo, esta divulgación describe un procedimiento de decodificación de datos de vídeo, comprendiendo el procedimiento: obtener elementos de sintaxis de un flujo de bits que incluye una secuencia de bits que forman una representación codificada de los datos de vídeo, en el que obtener los elementos de sintaxis del flujo de bits comprende determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando codificación con pérdida sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación por codificación de impulsos diferencial (DPCM) residual; y reconstruir una imagen de los datos de vídeo en base al menos en parte a los elementos de sintaxis obtenidos del flujo de bits.

[0009] En otro ejemplo, esta divulgación describe un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento: generar un flujo de bits que incluye una secuencia de bits que forma una representación codificada de los datos de vídeo, en el que generar el flujo de bits comprende determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación por codificación de impulsos diferencial (DPCM) residual; y facilitar el flujo de bits.

[0010] En otro ejemplo, esta divulgación describe un aparato de codificación de vídeo que comprende: una memoria que almacena datos; y uno o más procesadores configurados para determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando codificación con pérdida sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación por codificación de impulsos diferencial (DPCM) residual.

[0011] En otro ejemplo, esta divulgación describe un aparato de decodificación de vídeo que comprende: medios para obtener elementos de sintaxis de un flujo de bits que incluye una secuencia de bits que forma una representación codificada de los datos de vídeo, en el que obtener los elementos de sintaxis del flujo de bits comprende determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando una codificación con pérdida sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación por codificación de impulsos diferencial (DPCM) residual; y medios para reconstruir una imagen de los datos de vídeo en base al menos en parte a los elementos de sintaxis obtenidos del flujo de bits.

[0012] En otro ejemplo, esta divulgación describe un aparato de codificación de vídeo que comprende: medios para generar un flujo de bits que incluye una secuencia de bits que forma una representación codificada de los datos de vídeo, en el que generar el flujo de bits comprende determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando una codificación con pérdida sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación por codificación de impulsos diferencial (DPCM) residual; y medios para facilitar el flujo de bits.

[0013] En otro ejemplo, esta divulgación describe un medio de almacenamiento de datos legible por ordenador no transitorio que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que uno o más procesadores: determinen que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando codificación con pérdida sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación por codificación de impulsos diferencial (DPCM) residual.

[0014] Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetivos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones. La presente invención se define en las reivindicaciones independientes.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0015]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo de ejemplo que puede utilizar las técnicas de esta divulgación.

La FIG. 2 es un diagrama conceptual que ilustra un bloque de tamaño M (altura) × N (anchura).

La FIG. 3 es un diagrama conceptual que ilustra unas direcciones de modo de intrapredicción de ejemplo.

La FIG. 4 es un diagrama conceptual que ilustra unas muestras de ejemplo que se pueden usar para la predicción en la codificación de vídeo.

La FIG. 5A muestra una dirección de modulación por codificación de impulsos diferencial (DPCM) residual para modos casi verticales.

La FIG. 5B muestra una dirección de DPCM residual para modos casi horizontales.

La FIG. 6 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas de esta divulgación.

La FIG. 7 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar las técnicas de esta divulgación.

La FIG. 8A es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 8B es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 9A es un diagrama de flujo que ilustra una operación de ejemplo de un decodificador de vídeo, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 9B es un diagrama de flujo que ilustra una operación de ejemplo de un decodificador de vídeo, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 10A es un diagrama de flujo que ilustra una operación de codificador de vídeo de ejemplo, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 10B es un diagrama de flujo que ilustra una operación de decodificador de vídeo de ejemplo para ocultación de datos de signo, de acuerdo con una o más técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0016] En general, esta divulgación describe técnicas para mejorar la intrapredicción en la codificación de vídeo de alta eficacia (HEVC) y otras normas de codificación de vídeo. La intrapredicción es un proceso de generación, en base a unos valores de muestra de una imagen actual, de un bloque predictivo para un bloque de vídeo de la imagen actual. Por tanto, cuando un bloque de vídeo de una imagen actual se codifica usando intrapredicción, el codificador de vídeo no usa valores de muestra de otras imágenes para generar o identificar de otro modo el bloque predictivo para el bloque de vídeo.

[0017] Después de generar un bloque predictivo, un codificador de vídeo puede usar el bloque predictivo para determinar un bloque de muestras residuales (es decir, un bloque residual). Las muestras residuales del bloque residual pueden indicar la diferencia entre unas muestras del bloque predictivo y unas muestras originales correspondientes del bloque de vídeo. El codificador de vídeo puede generar un bloque de coeficientes de transformada aplicando una transformada al bloque residual. La transformada puede convertir las muestras residuales de un dominio de píxel en un dominio de frecuencia. El codificador de vídeo puede cuantificar a continuación los coeficientes de transformada del bloque de coeficientes de transformada para reducir las profundidades de bits de los coeficientes de transformada. El codificador de vídeo puede codificar entrópicamente elementos de sintaxis que representan los coeficientes de transformada cuantificados e incluir los elementos de sintaxis codificados entrópicamente resultantes en un flujo de bits.

[0018] Un decodificador de vídeo puede realizar este proceso a la inversa. Es decir, el decodificador de vídeo puede decodificar entrópicamente elementos de sintaxis del flujo de bits para determinar coeficientes de transformada cuantificados. A continuación, el decodificador de vídeo puede cuantificar inversamente los coeficientes de transformada cuantificados para determinar los coeficientes de transformada. Además, el decodificador de vídeo puede aplicar una transformada inversa a los coeficientes de transformada para determinar el bloque residual. Además, el decodificador de vídeo puede determinar un bloque predictivo (por ejemplo, usando intrapredicción). El decodificador de vídeo puede usar muestras en el bloque predictivo y muestras residuales correspondientes en el bloque residual para reconstruir muestras del bloque de vídeo.

[0019] La aplicación de la transformada y el uso de la cuantificación causan una pérdida de información. Por tanto, las muestras de un bloque de vídeo reconstruidas por el decodificador de vídeo pueden no tener el mismo nivel de precisión que las muestras originales del bloque de vídeo. En consecuencia, la aplicación de la transformada y el uso de la cuantificación puede ser una forma de codificación "con pérdida". En algunos casos, el codificador de vídeo puede codificar un bloque de vídeo usando codificación sin pérdida. Cuando el codificador de vídeo codifica un bloque de vídeo usando codificación sin pérdida, el codificador de vídeo no aplica la transformada a las muestras residuales y no cuantifica las muestras residuales. Del mismo modo, el decodificador de vídeo no aplica la cuantificación inversa

o la transformada inversa. Como resultado, las muestras del bloque de vídeo reconstruidas por el descodificador de vídeo pueden tener el mismo nivel de precisión que las muestras originales del bloque de vídeo.

[0020] En otros casos, el codificador de vídeo puede realizar un tipo de codificación con pérdida en la cual el codificador de vídeo no aplica una transformada a las muestras residuales, pero sí cuantifica las muestras residuales. Del mismo modo, el descodificador de vídeo puede aplicar una cuantificación inversa a las muestras residuales, pero no aplicar una transformada inversa a las muestras residuales. Debido a que el codificador de vídeo sigue aplicando cuantificación a las muestras residuales, las muestras reconstruidas por el descodificador de vídeo pueden tener menos precisión que las muestras originales, pero la pérdida de precisión puede ser potencialmente menor que si se hubiera aplicado la transformada.

[0021] Como se indica anteriormente, un codificador de vídeo (por ejemplo, un codificador de vídeo o un descodificador de vídeo) puede usar intrapredicción para generar un bloque predictivo. Más específicamente, el codificador de vídeo usa un modo de intrapredicción en particular de entre una pluralidad de modos de intrapredicción disponibles para generar el bloque predictivo. En HEVC y otras normas de codificación de vídeo, los modos de intrapredicción incluyen una pluralidad de modos de intrapredicción direccionales, un modo de intrapredicción plano y un modo de intrapredicción DC. En general, cuando el codificador de vídeo genera un bloque predictivo usando el modo de intrapredicción plano, las muestras del bloque predictivo se pueden determinar en base a una combinación de proyecciones lineales. Cuando el codificador de vídeo genera un bloque predictivo usando el modo de intrapredicción DC, el codificador de vídeo puede determinar un valor de intrapredicción DC. El valor de intrapredicción DC puede ser un valor promedio de muestras adyacentes a un borde izquierdo y un borde superior del bloque predictivo. El codificador de vídeo puede hacer que cada valor de muestra del bloque predictivo sea igual al valor de intrapredicción DC.

[0022] Algunas técnicas de esta divulgación proporcionan mejoras del modo de intrapredicción DC cuando un codificador de vídeo usa codificación sin pérdida. En la codificación sin pérdida, un codificador de vídeo puede usar valores originales de muestras cuando se usa un modo de intrapredicción DC para determinar valores de muestras en un bloque predictivo. En la codificación con pérdida, un descodificador de vídeo no tiene acceso a los valores originales de las muestras cuando se usa la intrapredicción DC para determinar valores de muestras en un bloque predictivo. Sin embargo, en la codificación sin pérdida, el descodificador de vídeo sí tiene acceso a los valores reconstruidos de las muestras cuando se usa intrapredicción DC para determinar valores en el bloque predictivo. En la codificación sin pérdida, los valores reconstruidos de las muestras son los mismos que los valores originales de las muestras.

[0023] Como se describe en el presente documento, el codificador de vídeo puede generar un bloque predictivo. Como parte de la generación del bloque predictivo, el codificador de vídeo puede usar al menos una de una muestra reconstruida sin pérdida a la izquierda de una muestra actual en una fila actual de un bloque predictivo y una muestra reconstruida sin pérdida para una fila del bloque predictivo encima de la fila actual para la predicción DC de la muestra actual. Además, en algunos casos, esto puede permitir que el descodificador de vídeo canalice la determinación de valores de muestra en el bloque predictivo.

[0024] Además, como se indica anteriormente, un codificador de vídeo puede realizar una forma de codificación con pérdida en la que se usa la cuantificación pero se omite la transformada, lo que se puede denominar codificación con omisión de transformada. De acuerdo con una o más técnicas adicionales de esta divulgación, el codificador de vídeo puede aplicar una forma de modulación de codificación de impulsos diferencial (DPCM) residual para preparar las muestras residuales no transformadas, pero cuantificadas, para la codificación. Esta forma de DPCM residual se describe en detalle en otra parte de esta divulgación. A diferencia de otras propuestas para usar DPCM en la intracodificación con pérdida, esta forma de DPCM residual descrita en esta divulgación puede incrementar el rendimiento del codificador y/o descodificador de vídeo.

[0025] Como se indica anteriormente, un codificador de vídeo puede codificar entrópicamente elementos de sintaxis que representan coeficientes de transformada cuantificados. En la codificación sin pérdida, o la codificación con pérdida cuando se omite la transformada, se pueden usar los mismos elementos de sintaxis para representar muestras residuales. En HEVC y otras normas de codificación de vídeo, los elementos de sintaxis que representan un coeficiente de transformada o una muestra residual pueden incluir un elemento de sintaxis de signo que indica si el coeficiente de transformada o la muestra residual son positivos o negativos. En algunos casos, puede ser innecesario incluir elementos de sintaxis de signo para indicar si un coeficiente de transformada o una muestra residual son positivos o negativos. En su lugar, la información que indica si un coeficiente de transformada o una muestra residual son positivos o negativos puede estar incluida en unos valores de otros elementos de sintaxis para el coeficiente de transformada o la muestra residual. Incluir dicha información en valores de otros elementos de sintaxis, en lugar de señalar elementos de sintaxis de signo, se puede denominar ocultación de datos de signo.

[0026] Sin embargo, la ocultación de datos de signo puede ser difícil de implementar para bloques que se codifican usando codificación con pérdida para los cuales se omite la transformada y un modo de intrapredicción plano, un modo de intrapredicción DC (por ejemplo, un modo de intrapredicción DC en el que se usan muestras reconstruidas correspondientes a unas muestras del bloque predictivo para determinar el valor de las muestras predictivas en el

bloque predictivo), o se usa DPCM residual. Además, en la codificación con omisión de transformada, la ocultación de datos de signo puede introducir, en los valores residuales, errores que se agravan cuando se aplica DPCM residual. Dichos errores se pueden propagar a muestras residuales posteriores, dando como resultado una degradación del rendimiento. Por tanto, de acuerdo con una o más técnicas de esta divulgación, la ocultación de datos de signo se puede inhabilitar normativamente para dichos bloques, incluso si uno o más elementos de sintaxis indican que la ocultación de datos de signo está habilitada para dichos bloques.

[0027] Por ejemplo, en algunos ejemplos, el descodificador de vídeo determina que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando codificación con pérdida sin la aplicación de una transformada a los datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa DPCM residual. En dichos ejemplos, cuando la ocultación de datos de signo está inhabilitada para el bloque actual, el descodificador de vídeo puede obtener, del flujo de bits, para cada valor significativo respectivo del bloque, un elemento de sintaxis respectivo que indica si el valor significativo respectivo es positivo o negativo.

[0028] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo 10 de ejemplo que puede utilizar las técnicas de esta divulgación. Como se describe en el presente documento, el término "codificador de vídeo" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" se pueden referir genéricamente a la codificación de vídeo o la descodificación de vídeo.

[0029] Como se muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. En consecuencia, el dispositivo de origen 12 se puede denominar dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 puede descodificar los datos de vídeo codificados generados por el dispositivo de origen 12. En consecuencia, el dispositivo de destino 14 se puede denominar dispositivo de descodificación de vídeo o aparato de descodificación de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo.

[0030] El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, incluyendo ordenadores de escritorio, dispositivos informáticos móviles, ordenadores tipo *notebook* (por ejemplo, portátiles), tabletas electrónicas, descodificadores multimedia, aparatos telefónicos tales como los denominados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores para vehículos o similares.

[0031] El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 por medio de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de transferir los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permiten al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El uno o más medios de comunicación pueden incluir medios de comunicación inalámbrica y/o alámbrica, tales como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global (por ejemplo, Internet). El canal 16 puede incluir diversos tipos de dispositivos, tales como encaminadores, conmutadores, estaciones base u otros equipos que facilitan la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0032] En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento, por ejemplo, por medio de acceso a disco o acceso a tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria *flash* u otros medios de almacenamiento digital adecuados para almacenar datos de vídeo codificados.

[0033] En otro ejemplo, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacena datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de archivos o en otro dispositivo de almacenamiento intermedio mediante transmisión continua o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo de destino 14. Entre los ejemplos de servidores de archivos se incluyen servidores web (por ejemplo, para una página web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a la red (NAS), unidades de disco local y similares.

[0034] El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, tal como una conexión a Internet. Los ejemplos de tipos de conexiones de datos pueden incluir canales

inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones alámbricas (por ejemplo, DSL, módem por cable, etc.) o combinaciones de ambos que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0035] Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, por medio de Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, decodificación de datos de vídeo almacenados en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, un sistema de codificación de vídeo puede estar configurado para admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0036] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos ejemplos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. La fuente de vídeo 18 puede incluir un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene datos de vídeo previamente capturados, una interfaz de imágenes de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de vídeo, o una combinación de dichas fuentes de datos de vídeo.

[0037] El codificador de vídeo 20 puede codificar datos de vídeo de la fuente de vídeo 18. En algunos ejemplos, el dispositivo de origen 12 transmite directamente los datos de vídeo codificados al dispositivo de destino 14 por medio de la interfaz de salida 22. En otros ejemplos, los datos de vídeo codificados también se pueden almacenar en un medio de almacenamiento o en un servidor de archivos para un acceso posterior por el dispositivo de destino 14 para su decodificación y/o su reproducción.

[0038] En el ejemplo de la FIG. 1, el dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. En algunos ejemplos, la interfaz de entrada 28 incluye un receptor y/o un módem. La interfaz de entrada 28 puede recibir datos de vídeo codificados a través del canal 16. El dispositivo de visualización 32 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En general, el dispositivo de visualización 32 visualiza datos de vídeo decodificados. El dispositivo de visualización 32 puede comprender 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.

[0039] La FIG. 1 es meramente un ejemplo y las técnicas de esta divulgación se pueden aplicar a configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o decodificación de vídeo) que no incluyen necesariamente ninguna comunicación de datos entre el dispositivo de codificación de vídeo y el dispositivo de decodificación de vídeo. En otros ejemplos, los datos se recuperan de una memoria local, se transmiten en continuo por una red, o similares. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en una memoria, y/o un dispositivo de decodificación de vídeo puede recuperar y decodificar datos de una memoria. En muchos ejemplos, la codificación y decodificación de vídeo se realiza mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y decodifican datos de la memoria.

[0040] Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos 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, hardware o cualquier combinación de los mismos. Si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado de almacenamiento no transitorio legible por ordenador, y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cualquiera de los anteriores (incluyendo hardware, software, una combinación de hardware y software, etc.) se puede considerar como uno o más procesadores. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

[0041] Esta divulgación en general puede indicar, con referencia al codificador de vídeo 20, que "señaliza" determinada información. El término "señalizar" se puede referir, en general, a la comunicación de elementos de sintaxis y/u otros datos usados para decodificar los datos de vídeo comprimidos. Dicha comunicación se puede producir en tiempo real o casi real. De forma alternativa, dicha comunicación se puede producir durante un lapso de tiempo, tal como se podría producir cuando se almacenan elementos de sintaxis en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que a continuación un dispositivo de decodificación de vídeo puede recuperar en cualquier momento tras haber sido almacenado en este medio.

[0042] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 funcionan de acuerdo con una norma de compresión de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC). Un borrador de la norma HEVC, denominado "HEVC Working Draft 6" se describe en el documento de Bross *et al.*, "High Efficiency Video Coding (HEVC) text specification draft 6," Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 7.^a reunión: Ginebra, Suiza, noviembre de 2011. Desde el 5 de abril de 2013, HEVC Working Draft 6 está disponible en http://phenix.intevry.fr/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-H1003-v22.zip. Otro borrador de la norma HEVC, denominado "HEVC Working Draft 9" se describe en el documento de Bross *et al.*, "High Efficiency Video Coding (HEVC) text specification draft 9," Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 11.^a reunión: Shanghai, China, 2012. Desde el 24 de marzo de 2014, HEVC Working Draft 9 está disponible en http://phenix.intevry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zip. Otro borrador de la norma HEVC, denominado "HEVC Working Draft 10", se describe en el documento de Bross *et al.*, "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent)," Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12.^a reunión: Ginebra, CH, 14-23 de enero de 2013. Desde el 24 de marzo de 2014, HEVC Working Draft 10 está disponible en http://phenix.intevry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v20.zip. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma o técnica de codificación en particular.

[0043] Además, se está elaborando una especificación de ampliación de alcance para HEVC. La especificación de ampliación de alcance contempla modos de muestreo alternativos, tales como muestreo de croma 4:0:0, 4:2:0, 4:2:2 y 4:4:4. El documento de Flynn *et al.*, "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 3," Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 13.^a reunión: Incheon, KR, 18-26 de abril de 2013, (en adelante, "JCTVC-M1005_v2") es un borrador de la especificación de ampliación de alcance para HEVC. Desde el 24 de marzo de 2014, JCTVC-M1005_v2 está disponible en http://phenix.intevry.fr/jct/doc_end_user/documents/13_Incheon/pending/JCTVC-M1005-v2.zip.

[0044] Como se ha mencionado anteriormente de forma resumida, el codificador de vídeo 20 codifica datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes es una imagen fija que forma parte de un vídeo. Cuando el codificador de vídeo 20 codifica los datos de vídeo, el codificador de vídeo 20 puede generar un flujo de bits. El flujo de bits puede incluir una secuencia de bits que forman una representación codificada de los datos de vídeo. El flujo de bits puede incluir imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen. Los datos asociados pueden incluir conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS) y otras estructuras sintácticas. Un SPS puede contener parámetros aplicables a cero o más secuencias de imágenes. Un PPS puede contener parámetros aplicables a cero o más imágenes.

[0045] Una imagen puede incluir tres matrices de muestras, indicadas por S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luma. Las muestras de luma también se pueden denominar en el presente documento muestras "Y". S_{Cb} es una matriz bidimensional de muestras de crominancia Cb. S_{Cr} es una matriz bidimensional de muestras de crominancia Cr. Las muestras de crominancia también se pueden denominar en el presente documento muestras de "croma". Las muestras de crominancia Cb se pueden denominar en el presente documento "muestras U". Las muestras de crominancia Cr se pueden denominar en el presente documento "muestras V".

[0046] En algunos ejemplos, el codificador de vídeo 20 puede submuestrear las matrices de croma de una imagen (es decir, S_{Cb} y S_{Cr}). Por ejemplo, el codificador de vídeo 20 puede usar un formato de vídeo YUV 4:2:0, un formato de vídeo YUV 4:2:2 o un formato de vídeo 4:4:4. En el formato de vídeo YUV 4:2:0, el codificador de vídeo 20 puede submuestrear las matrices de croma, de modo que las matrices de croma tengan la $\frac{1}{2}$ de la altura y la $\frac{1}{2}$ de la anchura de la matriz de luma. En el formato de vídeo YUV 4:2:2, el codificador de vídeo 20 puede submuestrear las matrices de croma, de modo que las matrices de croma tengan la $\frac{1}{2}$ de la anchura y la misma altura que la matriz de luma. En el formato de vídeo YUV 4:4:4, el codificador de vídeo 20 no submuestrea las matrices de croma.

[0047] Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede ser un bloque de árbol de codificación de muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma y unas estructuras sintácticas usadas para codificar las muestras de los bloques de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque NxN de muestras. Una CTU también se puede denominar "bloque de árbol" o "unidad de codificación de mayor tamaño" (LCU). Las CTU de HEVC pueden ser análogas en términos generales a los macrobloques de otras normas, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño en particular y puede incluir una o más unidades de codificación (CU).

[0048] Como parte de la codificación de una imagen, el codificador de vídeo 20 puede generar representaciones codificadas de cada sector de la imagen (es decir, sectores codificados). Para generar un sector codificado, el codificador de vídeo 20 puede codificar una serie de CTU. Esta divulgación se puede referir a una representación

codificada de una CTU como una CTU codificada. En algunos ejemplos, cada uno de los sectores incluye un número entero de CTU codificadas.

[0049] Para generar una CTU codificada, el codificador de vídeo 20 puede realizar de forma recursiva una división de árbol cuaternario en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación; de ahí el nombre de "unidades de árbol de codificación". Un bloque de codificación es un bloque NxN de muestras. Una CU puede ser un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras Cb y una matriz de muestras Cr, y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas, o imágenes codificadas usando planos de color separados, una CU puede comprender un único bloque de codificación de muestras y estructuras sintácticas usadas para codificar el bloque de codificación. El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción puede ser un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras a las que se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede ser un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma de una imagen y estructuras sintácticas usadas para predecir las muestras de bloque de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr, para bloques de predicción de luma, Cb y C, de cada PU de la CU. En imágenes monocromáticas, o imágenes codificadas usando planos de color separados, una CU puede comprender un único bloque de codificación de muestras y estructuras sintácticas usadas para codificar el bloque de codificación.

[0050] El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa intrapredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a unas muestras descodificadas de la imagen asociada a la PU.

[0051] Si el codificador de vídeo 20 usa interpredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a unas muestras descodificadas de una o más imágenes distintas a la imagen asociada a la PU. El codificador de vídeo 20 puede usar unipredicción o bipredicción para generar los bloques predictivos de una PU. Cuando el codificador de vídeo 20 usa unipredicción para generar los bloques predictivos para una PU, la PU puede tener un único vector de movimiento. Cuando el codificador de vídeo 20 usa unipredicción para generar los bloques predictivos para una PU, la PU puede tener dos vectores de movimiento.

[0052] Después de que el codificador de vídeo 20 genere bloques predictivos (por ejemplo, bloques de luma, Cb y Cr) para una o más PU de una CU, el codificador de vídeo 20 puede generar un bloque residual para la CU. Por ejemplo, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra del bloque residual de luma de la CU indica una diferencia entre una muestra de luma de uno de los bloques predictivos de luma de la CU y una muestra correspondiente del bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual Cb para la CU. Cada muestra del bloque residual Cb de la CU puede indicar una diferencia entre una muestra Cb en uno de los bloques predictivos Cb de la CU y una muestra correspondiente en el bloque de codificación Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual Cr para la CU. Cada muestra del bloque residual Cr de la CU puede indicar una diferencia entre una muestra Cr en uno de los bloques predictivos Cr de la CU y una muestra correspondiente del bloque de codificación Cr original de la CU.

[0053] Además, el codificador de vídeo 20 puede usar una división de árbol cuaternario para descomponer los bloques residuales de una CU en bloques de transformada. Por ejemplo, el codificador de vídeo 20 puede usar una división de árbol cuaternario para descomponer bloques residuales de luma, Cb y Cr de una CU en bloques de transformada de luma, Cb y Cr. Un bloque de transformada puede ser un bloque rectangular de muestras al que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede ser un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformada. Por tanto, cada TU de una CU puede estar asociada a un bloque de transformada de luma, un bloque de transformada Cb y un bloque de transformada Cr. El bloque de transformada de luma asociado a la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada Cr puede ser un subbloque del bloque residual Cr de la CU. En imágenes monocromáticas o imágenes codificadas usando tres planos de color separados, una TU puede comprender un único bloque de transformada y estructuras sintácticas usadas para transformar las muestras del bloque de transformada. Un tamaño de TU puede ser el tamaño de un bloque de transformada de una TU.

[0054] El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de una TU para generar un bloque de coeficientes para la TU. Por ejemplo, el codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficientes de luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. Además, el codificador de vídeo 20 puede aplicar una o más

transformadas a un bloque de transformada Cb de una TU para generar un bloque de coeficientes Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada Cr de una TU para generar un bloque de coeficientes Cr para la TU.

[0055] Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes Cb o un bloque de coeficientes Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. 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 usados para representar los coeficientes de transformada, proporcionando más compresión. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede codificar entrópicamente unos elementos de sintaxis que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos de sintaxis que indican los coeficientes de transformada cuantificados. El codificador de vídeo 20 puede facilitar los elementos de sintaxis codificados entrópicamente en un flujo de bits.

[0056] El descodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar el flujo de bits para descodificar elementos de sintaxis del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base, al menos en parte, a los elementos de sintaxis descodificados del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20. Por ejemplo, el descodificador de vídeo 30 puede usar vectores de movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantificar inversamente bloques de coeficientes de transformada asociados a unas TU de la CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir bloques de transformada asociados a las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a unas muestras correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

[0057] Como se indica anteriormente, un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede usar intrapredicción para generar un bloque predictivo para una PU actual. Cuando un codificador de vídeo usa intrapredicción para generar un bloque predictivo para una PU actual, el codificador de vídeo puede determinar unos valores de muestras en el bloque predictivo usando un conjunto de muestras de referencia. Por ejemplo, en la intrapredicción HEVC, las muestras ya reconstruidas de los bloques vecinos superior e izquierdo se pueden usar para una predicción. Estas muestras reconstruidas se pueden denominar muestras de referencia.

[0058] La FIG. 2 ilustra muestras de referencia de un bloque para intrapredicción HEVC. En otras palabras, la FIG. 2 es un diagrama conceptual que ilustra un bloque de tamaño M (altura) $\times$ N (anchura). En la FIG. 2, M indica filas y N indica columnas. Además, en la FIG. 2, las muestras de un bloque se denotan por $P_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$. En esta divulgación, el término "muestras" se puede referir a los valores de píxel originales de un componente de entrada (por ejemplo, R, G o B en una imagen RGB, Y, Cb o Cr, en imágenes YCbCr, etc.) o valores de muestra de un componente después de aplicar una transformada de color a los componentes de entrada. En el ejemplo de la FIG. 2, los píxeles de referencia se denotan por $P_{-1,j}$, donde $-1 \leq j < 2N$ y $P_{i,-1}$, donde $-1 \leq i < 2M$.

[0059] Como se muestra en el ejemplo de la FIG. 2, las muestras de referencia pueden incluir un conjunto de muestras de referencia a la izquierda de la PU actual y un conjunto de muestras de referencia encima de la PU actual. En esta divulgación, se puede denominar predictor superior al conjunto de muestras de referencia de encima de la PU actual. En esta divulgación, se puede denominar predictor izquierdo al conjunto de muestras de referencia a la izquierda de la PU actual. En otras palabras, en la intrapredicción HEVC, las muestras ya reconstruidas de los bloques vecinos superior e izquierdo se usan para la predicción (el bloque vecino "superior" también se puede denominar bloque vecino "de encima"). Estas muestras se denominan muestras de referencia. En algunos ejemplos, si no hay píxeles de referencia disponibles, un codificador de vídeo que usa HEVC puede usar un proceso de relleno específico para generar muestras de referencia ausentes.

[0060] Cuando un codificador de vídeo usa intrapredicción para generar un bloque predictivo, el codificador de vídeo puede generar el bloque predictivo de acuerdo con un modo de intrapredicción a partir de una pluralidad de modos de intrapredicción disponibles. Los modos de intrapredicción pueden incluir una pluralidad de modos de intrapredicción direccional (es decir, angular). Por ejemplo, en algunas versiones de HEVC, hay 33 modos de intrapredicción direccional. Cada uno de los modos de intrapredicción direccional corresponde a una dirección diferente. La FIG. 3 es un diagrama conceptual que ilustra unas direcciones de modo de intrapredicción de ejemplo. Cuando el codificador de vídeo genera un bloque predictivo de acuerdo con un modo de intrapredicción direccional, el codificador de vídeo puede, para cada muestra respectiva del bloque predictivo, asignar a la muestra respectiva un valor de una muestra de referencia (o una combinación ponderada de muestras de referencia) que está alineada con la muestra respectiva en una dirección correspondiente al modo de intrapredicción direccional. Cuando un codificador de vídeo usa un modo de intrapredicción direccional (es decir, angular) para generar un bloque predictivo para un bloque actual, se puede decir que el codificador de vídeo realiza una intrapredicción angular.

[0061] Además, en algunas versiones de HEVC, los modos de intrapredicción incluyen un modo de intrapredicción DC. En dichas versiones de HEVC, cuando el codificador de vídeo usa intrapredicción DC para generar un bloque predictivo, el codificador de vídeo puede determinar un valor medio de las muestras de referencia. El codificador de vídeo puede determinar a continuación que cada muestra del bloque predictivo tiene el valor medio determinado. Por tanto, cuando el codificador de vídeo usa intrapredicción DC para generar un bloque predictivo, todas las muestras del bloque predictivo tienen el mismo valor. Por ejemplo, se va a suponer que se ha llevado a cabo un proceso de relleno para que todas las muestras de referencia estén disponibles. En este ejemplo, con respecto al bloque 4x4 mostrado en el ejemplo de la FIG. 2, la predicción DC se puede constituir como:

$$\left(\sum_{i=0}^3 P_{i,-1} + \sum_{j=0}^3 P_{-1,j} + 4 \right) \gg 3, \quad (1)$$

donde >> denota una operación a nivel de bit de desplazamiento a la derecha.

[0062] En algunas versiones de HEVC, los modos de intrapredicción incluyen un modo de intrapredicción plano. Cuando un codificador de vídeo genera un bloque predictivo para una PU usando el modo de intrapredicción plano, el codificador de vídeo puede determinar un conjunto de muestras vecinas, $p[x][y]$, con $x = -1$, $y = -1..nTbS*2-1$ y $x = 0..nTbS*2-1$, $y = -1$, donde $nTbS$ es el tamaño de la PU actual. Además, $predSamples[x][y]$ puede denotar el valor de una muestra en la posición x , y del bloque de predicción. El codificador de vídeo puede determinar las muestras del bloque predictivo como sigue:

$$\begin{aligned} predSamples[x][y] = & \left((nTbS - 1 - x) * p[-1][y] + \right. \\ & (x + 1) * p[nTbS][-1] + \\ & (nTbS - 1 - y) * p[x][-1] + \\ & \left. (y + 1) * p[-1][nTbS] + nTbS \right) \gg (\log_2(nTbS) + 1) \end{aligned} \quad (2)$$

En términos generales, cuando se usa el modo de intrapredicción plano, el valor de una muestra del bloque predictivo es un promedio de dos interpolaciones lineales del valor. En la primera interpolación lineal, a medida que los valores de x se incrementan de izquierda a derecha a lo través de una fila del bloque predictivo, un peso otorgado a una muestra de referencia a la izquierda de la fila disminuye, mientras que un peso otorgado a una muestra de referencia encima y a la derecha de una esquina superior derecha del bloque predictivo se incrementa. En la segunda interpolación lineal, a medida que los valores de y se incrementan hacia abajo por una columna del bloque predictivo, un peso otorgado a una muestra de referencia encima de la columna disminuye, mientras que un peso otorgado a una muestra debajo y a la izquierda de una esquina inferior izquierda del bloque predictivo se incrementa.

[0063] En el ejemplo de la FIG. 2, la intrapredicción plana puede usar muestras $P_{-1,j}$, donde $0 \leq j \leq (N - 1)$ y $P_{M,-1}$ para generar una predicción bilineal en la dirección vertical. De forma similar, las muestras $P_{i,-1}$, donde $0 \leq i \leq (M - 1)$ y $P_{-1,N}$ se pueden usar para generar una predicción bilineal en la dirección horizontal. Por último, en este ejemplo, las predicciones horizontales y verticales se pueden promediar (o posiblemente combinar con otra operación matemática). Por ejemplo, si los valores resultantes de una predicción plana se denotan por T_{ij} y se supone que el bloque es cuadrado, es decir, $M = N$. En este ejemplo, T_{ij} se puede determinar como:

$$\begin{aligned} T_{i,j}^V &= (M - i) * P_{-1,j} + i * P_{M,-1}, \\ T_{i,j}^H &= (N - j) * P_{i,-1} + j * P_{-1,N} \text{ y} \\ T_{i,j} &= (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1). \end{aligned} \quad (3)$$

En este ejemplo, $*$ denota producto, $\gg$ denota una operación a nivel de bit de desplazamiento a la derecha y los superíndices H y V denotan predicciones en las direcciones horizontal y vertical, respectivamente.

[0064] En algunos casos, el codificador de vídeo 20 y el decodificador de vídeo 30 implementan un modo de codificación sin pérdida como se describe en el presente documento. Típicamente, cuando el codificador de vídeo 20 codifica un bloque, el codificador de vídeo 20 transforma (por ejemplo, usando una transformada de coseno discreta) y cuantifica datos residuales (es decir, el error de predicción) para el bloque. En otras palabras, el error de predicción se transforma y cuantifica. Sin embargo, cuando el codificador de vídeo 20 codifica un bloque (por ejemplo, una CU) usando codificación sin pérdida, el codificador de vídeo 20 puede no aplicar una transformada o cuantificación a los datos residuales para el bloque. En otras palabras, en el modo de codificación sin pérdida (por ejemplo, para una CU o una imagen completa), se pueden omitir las etapas de transformada y cuantificación. En cambio, el codificador de vídeo 20 puede tratar los valores de muestra de los datos residuales de la misma manera que los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede codificar entrópicamente elementos de

sintaxis que representan valores de muestra de los datos residuales e incluir los datos resultantes en un flujo de bits. Por tanto, los datos residuales no sufren ninguna pérdida de información debido a la transformación o la cuantificación.

[0065] De manera similar, en algunos casos donde el descodificador de vídeo 30 descodifica un bloque (por ejemplo, una CU) que se ha codificado usando codificación sin pérdida, el descodificador de vídeo 30 puede no aplicar una cuantificación inversa o unas transformadas inversas a los datos residuales para el bloque. En su lugar, el descodificador de vídeo 30 puede descodificar entrópicamente elementos de sintaxis que representan valores de muestra de los datos residuales y, a continuación, reconstruir valores de muestra del bloque en base al menos en parte a unos valores de muestra de los datos residuales.

[0066] Varias técnicas de ejemplo de esta divulgación se refieren a la codificación sin pérdida. Como se describe en el presente documento, en lugar de usar muestras de referencia de bloques vecinos para una predicción, se pueden usar muestras de un bloque actual para mejorar la predicción. Por ejemplo, varias técnicas de ejemplo de esta divulgación describen modificaciones que pueden ser aplicables a un modo de intrapredicción DC para codificación sin pérdida en la norma HEVC. Además, varias técnicas de ejemplo de esta divulgación describen modificaciones que pueden ser aplicables a un modo de intrapredicción plana para codificación sin pérdida en la norma HEVC. Las técnicas de esta divulgación también pueden ser aplicables a otros tipos de predicción, y también pueden ser aplicables a otras normas de codificación.

[0067] En la codificación sin pérdida, en lugar de usar muestras de referencia de bloques vecinos para predicción, las muestras del bloque actual se pueden usar para mejorar la predicción. Por ejemplo, se describen unas técnicas para intrapredicción angular para modos de codificación con pérdida, así como unos modos de codificación sin pérdida cuando se omite la transformada, en el documento de Lan *et al.*, "Intra and inter coding tools for screen contents", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT -VC) de ITU-T SG 16 WP3 e ISO/IEC JTC1/SC29/WG11, 5.^a reunión, Ginebra, CH, 16-23 de marzo de 2011, documento JCTVC-E145 (a continuación en el presente documento, "JCTVC-E145"). Además, para la codificación sin pérdida, se omiten las etapas de transformada y cuantificación, por lo que puede ser posible mejorar el proceso para determinar muestras de un bloque predictivo cuando se usa el modo intra DC.

[0068] De acuerdo con una o más técnicas de esta divulgación, para el cálculo del bloque predictivo, se puede suponer que las muestras se están procesando en un barrido a lo largo de las filas. Sin embargo, en otros ejemplos, las mismas técnicas se pueden ampliar al barrido a lo largo de las columnas (o incluso a exploraciones en diagonal o en zig-zag, pero perdiendo cierto potencial de paralelización de las técnicas). Cuando no se aplica ninguna transformada o cuantificación al error de predicción en modo sin pérdida, se puede suponer que después de la descodificación entrópica del error de predicción (es decir, el residuo), la muestra original se puede reconstruir sin pérdida. Por consiguiente, en el contexto de codificación sin pérdida, el término "muestra original" o "valor de muestra original" se puede referir a valores de muestra originales reales o valores de muestra reconstruidos (es decir, muestras no residuales). Debido al barrido a lo largo de las filas, todas las muestras de las filas previas, así como todas las muestras a la izquierda de la muestra actual de la fila actual, están disponibles para la predicción DC. Una o más técnicas de esta divulgación aprovechan esto para mejorar la intrapredicción usando el modo de intrapredicción DC.

[0069] Esta divulgación describe varios modos de intrapredicción de ejemplo. Los modos de intrapredicción de esta divulgación pueden reemplazar el modo de intrapredicción plano o se pueden entender como modos de intrapredicción planos. Por tanto, los modos de intrapredicción de esta divulgación pueden reemplazar el modo actual plano en HEVC para una codificación sin pérdida. Los detalles descritos en los ejemplos de esta divulgación se pueden combinar con uno o más detalles de otros ejemplos de esta divulgación. Es decir, los detalles se pueden combinar en cualquiera de una amplia variedad de formas diferentes para obtener aún más ejemplos.

[0070] De acuerdo con algunos ejemplos de esta divulgación en los que un codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el descodificador de vídeo 30) genera un bloque predictivo usando un modo de intrapredicción DC, el codificador de vídeo puede procesar muestras de un bloque predictivo en un orden de barrido. Cuando el codificador de vídeo procesa una muestra del bloque predictivo, el codificador de vídeo puede usar un vecindario causal de la muestra del bloque predictivo para formar un valor de predicción DC para la muestra. En general, un vecindario causal de una muestra de un bloque predictivo es un conjunto de muestras reconstruidas (por ejemplo, muestras no predictivas y no residuales) que corresponden a muestras del bloque predictivo que ya se han determinado. Por ejemplo, cuando se usa un orden de barrido a lo largo de las filas empezando desde una muestra superior izquierda del bloque predictivo, el vecindario causal de una muestra del bloque predictivo puede incluir muestras reconstruidas que corresponden a las ubicaciones de encima y de la izquierda de la muestra. En uno de dichos ejemplos, cuando un codificador de vídeo genera un bloque predictivo usando un modo de intrapredicción DC, el codificador de vídeo calcula el valor de predicción DC, $DC_{i,j}$, para una muestra actual $P_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, como:

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + 1) \gg 1. \quad (4)$$

o de forma similar,

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j}) \gg 1. \quad (5)$$

Por tanto, en las ecuaciones (4) y (5), para cada muestra respectiva de un bloque predictivo, el codificador de vídeo 20 puede determinar un valor de intrapredicción DC (es decir, $DC_{i,k}$) para la muestra respectiva como un promedio de un valor reconstruido de una muestra de encima de la muestra respectiva (es decir, $P_{i,j-1}$) y un valor reconstruido de una muestra de la izquierda de la muestra respectiva (es decir, $P_{i-1,j}$). Debido a que el codificador de vídeo 20 está usando una codificación sin pérdida, el valor reconstruido de la muestra de la izquierda de la muestra respectiva y el valor reconstruido de la muestra de encima de la muestra respectiva son los mismos que los valores originales de la muestra de la izquierda de la muestra respectiva y el valor original de la muestra de encima de la muestra respectiva. De forma similar, en las ecuaciones (4) y (5), para cada muestra respectiva de un bloque predictivo, el descodificador de vídeo 30 puede determinar un valor de intrapredicción DC (es decir, $DC_{i,k}$) para la muestra respectiva como un promedio de un valor reconstruido de una muestra de encima de la muestra respectiva (es decir, $P_{i,j-1}$) y un valor reconstruido de una muestra de la izquierda de la muestra respectiva (es decir, $P_{i-1,j}$). Debido a que el descodificador de vídeo 30 está usando codificación sin pérdida, el valor reconstruido de la muestra de encima de la muestra respectiva y el valor reconstruido de la muestra de la izquierda de la muestra respectiva son los mismos que el valor original de la muestra de encima de la muestra respectiva y el valor original de la muestra de la izquierda de la muestra respectiva.

[0071] Debido al barrido y al proceso de relleno para muestras de referencia, las muestras reconstruidas $P_{i,j-1}$ y $P_{i-1,j}$ pueden estar siempre disponibles. En otras palabras, una muestra de encima de la muestra actual y una muestra de referencia de la izquierda de la muestra actual pueden estar siempre disponibles. Además, debido a la naturaleza no lineal de la operación de desplazamiento a la derecha usada al calcular el valor de predicción DC $DC_{i,j}$, puede ser difícil para el descodificador de vídeo 30 procesar múltiples muestras en paralelo. Por ejemplo, $DC_{i,j+1}$ se puede representar como

$$DC_{i,j+1} = (P_{i,j} + P_{i-1,j+1} + 1) \gg 1,$$

$$DC_{i,j+1} = \left(R_{i,j} + \left((P_{i,j-1} + P_{i-1,j} + 1) \gg 1 \right) + P_{i-1,j+1} + 1 \right) \gg 1,$$

En las ecuaciones anteriores, $R_{i,j}$ es la predicción residual para la muestra en la ubicación (i,j). Debido a que el desplazamiento de bits a la derecha es un proceso no lineal, puede que no haya forma de calcular $DC_{i,j+1}$ antes de terminar el cálculo de $DC_{i,j}$. Por ejemplo, puede ser difícil para el descodificador de vídeo 30 procesar múltiples muestras del bloque predictivo en paralelo.

[0072] Cuando el descodificador de vídeo 30 descodifica una representación codificada de un bloque actual que se codifica usando codificación sin pérdida, el descodificador de vídeo 30 puede obtener, de un flujo de bits, elementos de sintaxis que indican valores de muestra residual del bloque actual. Por lo tanto, cuando el bloque actual se codifica usando codificación sin pérdida, el descodificador de vídeo 30 no necesita aplicar ni cuantificación inversa ni transformada inversa para determinar los valores de muestra residual del bloque actual. Obtener los elementos de sintaxis del flujo de bits puede conllevar descodificar entrópicamente los elementos de sintaxis. En consecuencia, cuando el bloque actual se codifica usando codificación sin pérdida o con pérdida, se puede suponer que ya se han descodificado entrópicamente los residuos de predicción (es decir, el error de predicción) del bloque actual.

[0073] Por tanto, cuando un bloque actual se codifica usando codificación sin pérdida o con pérdida, los residuos de predicción están disponibles para su uso en la determinación de los valores reconstruidos de unas muestras que el descodificador de vídeo 30 usa al determinar los valores de predicción DC para muestras de un bloque predictivo para el bloque actual. Suponiendo que los residuos de predicción ya se han descodificado entrópicamente, puede ser posible canalizar el procesamiento de muestras en diferentes filas con un retardo de una muestra. Por tanto, de acuerdo con una o más técnicas de esta divulgación, el descodificador de vídeo 30 puede comenzar a procesar una segunda fila de muestras después de que se haya reconstruido una muestra de la primera fila. De esta manera, el descodificador de vídeo 30 puede procesar múltiples filas de muestras en paralelo. En consecuencia, como parte de la generación de un bloque predictivo, un codificador de vídeo puede canalizar el procesamiento de muestras en diferentes filas del bloque predictivo, en el que existe un retardo de un ciclo para la predicción DC entre las filas del bloque predictivo.

[0074] En otro ejemplo, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el descodificador de vídeo 30) puede calcular un valor de predicción DC $DC_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$, como:

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1}). \quad (6)$$

Por tanto, en este ejemplo, para cada muestra respectiva de un bloque predictivo, el codificador de vídeo 20 puede determinar un valor de predicción DC (es decir, $DC_{i,j}$) para la muestra respectiva como una suma de un valor reconstruido de una muestra de encima de la muestra respectiva (es decir, $P_{i,j-1}$) y un valor reconstruido de una muestra de la izquierda de la muestra respectiva (es decir, $P_{i-1,j}$), menos un valor original de una muestra justo encima y a la izquierda de la muestra respectiva (es decir, $P_{i-1,j-1}$). De forma similar, para cada muestra respectiva de un bloque predictivo, el descodificador de vídeo 30 puede determinar un valor de predicción DC (es decir, $DC_{i,j}$) para la muestra respectiva como una suma de un valor reconstruido de una muestra de encima de la muestra respectiva (es decir, $P_{i,j-1}$) y un valor reconstruido de una muestra de la izquierda de la muestra respectiva (es decir, $P_{i-1,j}$), menos un valor reconstruido de una muestra justo encima y a la izquierda de la muestra respectiva (es decir, $P_{i-1,j-1}$). Debido a que se está usando una codificación sin pérdida, el valor reconstruido de las muestras es el mismo que el valor original de las muestras.

[0075] En este ejemplo, puede ser más sencillo procesar múltiples valores de muestra de una fila usando una lógica adicional. Por ejemplo, un codificador de vídeo puede calcular $DC_{i,j+1}$ para una muestra actual $P_{i,j}$ de un bloque predictivo sin esperar a $P_{i,j}$, como sigue. $DC_{i,j+1}$ se puede expresar como:

$$DC_{i,j+1} = ((P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1}) + r_{i,j} + P_{i-1,j+1} - P_{i-1,j}) \quad (7)$$

La ecuación (7) puede reescribirse como sigue:

$$DC_{i,j+1} = \sum_{n=0}^j r_{i,j-n} + P_{i-1,j+1} + P_{i-1,j} + P_{i-1,j-1}. \quad (8)$$

En las ecuaciones (7) y (8), $r_{i,j}$ es el residuo de error de predicción para la muestra $P_{i,j}$. Por tanto, en este ejemplo, el cálculo de un valor de intrapredicción DC para una muestra en particular en la ubicación $(i,j+1)$ (es decir, $DC_{i,j+1}$) no depende de los valores reconstruidos de las muestras de la izquierda de la muestra en particular. En su lugar, el cálculo del valor de intrapredicción DC $DC_{i,j+1}$ para una muestra en particular puede depender del valor reconstruido para la muestra directamente encima, así como de los valores residuales para todas las muestras de la izquierda de la muestra actual, y las muestras de referencia de la misma fila y la fila de encima. Esto puede permitir que el descodificador de vídeo 30 (así como el codificador de vídeo 20) calcule los valores de predicción DC para todas las muestras de una fila de un bloque en paralelo, suponiendo que los residuos para toda la fila ya se hayan descodificado.

[0076] Debido a que se está usando codificación sin pérdida, el valor reconstruido de las muestras (es decir, $P_{i,j-1}$, $P_{i-1,j}$, y $P_{i-1,j-1}$) es el mismo que el valor original de las muestras. Como se describe además en otra parte de esta divulgación, esta técnica se puede aplicar a la codificación con pérdida. En ese caso, para mantener la paralelización, es necesario usar un valor reconstruido sin recortar para la muestra de la izquierda para la predicción DC. Las muestras reconstruidas de la fila de encima pueden estar recortadas o no recortadas. Por ejemplo, para una secuencia de vídeo de 8 bits, las muestras reconstruidas están recortadas en el intervalo $[0,255]$.

[0077] En otro ejemplo, las muestras de la izquierda, de la esquina superior izquierda, de encima y de la esquina superior derecha se usan para determinar los valores de intrapredicción DC. En este ejemplo, un codificador de vídeo puede calcular un valor de predicción DC $DC_{i,j}$ para una muestra actual $P_{i,j}$ de un bloque predictivo, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$, como:

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1} + 2) \gg 2, \quad (9)$$

o de forma similar,

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1}) \gg 2. \quad (10)$$

Por tanto, en los ejemplos de las ecuaciones (9) y (10), para cada muestra respectiva de un bloque predictivo, un valor de intrapredicción DC para la muestra respectiva (es decir, $DC_{i,j}$) es un promedio de la muestra reconstruida de encima de la muestra respectiva (es decir, $P_{i,j-1}$), la muestra reconstruida de la izquierda de la muestra respectiva (es decir, $P_{i-1,j}$), la muestra reconstruida de encima y de la izquierda de la muestra respectiva (es decir, $P_{i-1,j-1}$), y la muestra reconstruida de encima y de la derecha de la muestra respectiva (es decir, $P_{i-1,j+1}$). Debido a que se está usando codificación sin pérdida, los valores reconstruidos de las muestras (es decir, $P_{i,j-1}$, $P_{i-1,j}$, $P_{i-1,j-1}$ y $P_{i-1,j+1}$) son los mismos que los valores originales de las muestras.

[0078] En los ejemplos que usan las ecuaciones (9) o (10) para determinar valores de intrapredicción DC, para las muestras de la última columna ($j = (N-1)$, $i > 0$), la muestra superior derecha no está disponible. Para superar esto, un codificador de vídeo puede suponer que las muestras de encima y superior derecha (es decir, $P_{i,j-1}$ y $P_{i-1,j+1}$, respectivamente) tienen el mismo valor. En otro ejemplo en el que un codificador de vídeo usa las ecuaciones (9) o (10) para determinar los valores de intrapredicción DC, el codificador de vídeo puede modificar la predicción DC para

usar solo muestras disponibles. En algunos ejemplos, una muestra puede no estar disponible si la muestra no está dentro de los límites de un sector o imagen actual, o si aún no se ha codificado.

[0079] Además, de acuerdo con otro ejemplo de esta divulgación, un codificador de vídeo puede realizar una intrapredicción DC en un tamaño de bloque más pequeño que el tamaño de TU. Por ejemplo, independientemente del tamaño de la TU, un codificador de vídeo puede realizar la intrapredicción DC en bloques de 2×2. El codificador de vídeo puede procesar los bloques 2×2 de un bloque predictivo en un orden de barrido. En este ejemplo, para las muestras $P_{2i,2j}$, $P_{2i,2j+1}$, $P_{2i+1,2j}$ y $P_{2i+1,2j+1}$, el codificador de vídeo calcula los valores de intrapredicción DC como:

$$(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1} + 2) \gg 2, \quad (11)$$

o de forma similar,

$$(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1}) \gg 2. \quad (12)$$

En este ejemplo, $0 \leq i \leq ((M/2) - 1)$ y $0 \leq j \leq ((N/2) - 1)$, donde M es la altura del bloque y N es la anchura del bloque. Además, en este ejemplo, se supone que M y N son pares. En este ejemplo, el codificador de vídeo puede procesar cuatro muestras en paralelo. En este ejemplo, el codificador de vídeo puede ser capaz de determinar los valores de intrapredicción DC de cada una de las cuatro muestras de un bloque 2×2 en paralelo. En ejemplos similares, el codificador de vídeo puede usar bloques 4×4 o bloques 8×8 en lugar de bloques 2×2.

[0080] De acuerdo con otro ejemplo de esta divulgación, se aprovecha la correlación entre los residuos después de realizar la predicción DC normal. Por ejemplo, se va a suponer que $r_{i,j}$, en el que $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, es el residuo de predicción después de realizar la intrapredicción DC como se especifica en HEVC (por ejemplo, en HEVC Working Draft 10). Por ejemplo, $r_{i,j}$ puede ser un valor residual de predicción después de realizar una intrapredicción DC como se describe en la ecuación (1) anterior, para un bloque 4×4. En este ejemplo, un codificador de vídeo puede generar a continuación valores intermedios $s_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$. El codificador de vídeo puede generar los valores intermedios $s_{i,j}$ como:

$$s_{i,j} = r_{i,2j}, \quad s_{i,(j+(N/2))} = r_{i,2j} - r_{i,2j+1} \quad (13)$$

En la ecuación (13) anterior, $0 \leq i \leq (M - 1)$, $0 \leq j \leq ((N/2) - 1)$.

[0081] El codificador de vídeo puede generar a continuación valores residuales modificados $t_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, como sigue:

$$t_{i,j} = s_{2i,j}, \quad t_{(i+(M/2)),j} = s_{2i,j} - s_{2i+1,j} \quad (14)$$

En la ecuación (14), $0 \leq i \leq ((M/2) - 1)$ y $0 \leq j \leq (N - 1)$.

[0082] El codificador de vídeo puede codificar entrópicamente los residuos modificados, $t_{i,j}$, como se describe en HEVC normal (por ejemplo, en el HEVC Working Draft 10). En el lado del decodificador (por ejemplo, en el decodificador de vídeo 30), este proceso se invierte. Por ejemplo, el decodificador de vídeo 30 puede determinar,

$$s_{2i,j} = t_{i,j}, \quad s_{2i+1,j} = t_{i,j} - t_{(i+(M/2)),j}, \quad (15)$$

donde $0 \leq i \leq ((M/2) - 1)$ y $0 \leq j \leq (N - 1)$. El decodificador de vídeo 30 también puede determinar,

$$r_{i,2j} = s_{i,j}, \quad r_{i,2j+1} = s_{i,j} - s_{i,(j+(N/2))}, \quad (16)$$

donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq ((N/2) - 1)$. En este ejemplo se supone que M y N son pares.

[0083] En otro ejemplo de esta divulgación, se puede usar un predictor potencialmente mejor en lugar de tomar una diferencia simple. Por ejemplo, un codificador de vídeo puede determinar $s_{i,j}$ como sigue:

$$s_{i,j} = r_{i,2j+1}, \quad s_{i,(j+(N/2))} = P_{i,2j} - ((P_{i,2j-1} + P_{i,2j+1} + 1) \gg 1) \quad (17)$$

$$t_{i,j} = s_{2i+1,j}, \quad t_{(i+(M/2)),j} = s_{2i,j} - ((s_{2i+1,j} + s_{2i+1,j} + 1) \gg 1) \quad (18)$$

En la ecuación (17) anterior, $0 \leq i < M$, $0 \leq j < (N/2)$. En la ecuación (18) anterior, $0 \leq i < M/2$, $0 \leq j < N$.

[0084] Las técnicas descritas en otros diversos ejemplos de esta divulgación se pueden aplicar para mejorar el modo de intrapredicción DC en la codificación con pérdida cuando la transformada se omite. En otras palabras, se pueden aplicar otros diversos ejemplos de esta divulgación para mejorar el modo de intrapredicción DC cuando un codificador de vídeo no aplica una transformada a muestras residuales de un bloque de transformada, pero sí cuantifica las muestras residuales del bloque de transformada. Por ejemplo, en el ejemplo descrito en el párrafo anterior, se usa un vecindario causal para calcular el valor de predicción DC para la muestra actual. En este ejemplo, en lugar de usar valores de muestra originales para calcular el valor de predicción DC como se hace en caso de codificación sin pérdida, se pueden usar valores de muestra reconstruidos (cuantificados) en el vecindario causal. Debido a que la aplicación de la transformada se omite, los valores reconstruidos en el vecindario causal están disponibles. Cabe destacar que para retener los beneficios de paralelización, la operación de recorte no se aplica a los valores reconstruidos de la fila actual hasta que no se termina el procesamiento para toda la fila. Para la fila de encima, se pueden usar valores reconstruidos recortados o no recortados.

[0085] De forma similar, como se describe en el párrafo anterior, una TU se divide en bloques más pequeños (por ejemplo, bloques 2x2) y se calcula un valor de predicción DC para cada bloque más pequeño. En lugar de usar valores de muestra originales para calcular el valor de predicción DC como se hace en el caso de la codificación sin pérdida, se pueden usar valores de muestra reconstruidos (cuantificados) en el caso de codificación con pérdida donde la transformada se omite.

[0086] Anteriormente, en esta divulgación, se describen técnicas de codificación con pérdida. De acuerdo con algunas de dichas técnicas, se va a suponer que $P_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$, denotan valores de muestra originales. Además, se va a suponer que $Q(P_{i,j})$ denota una versión cuantificada de $P_{i,j}$. A continuación, de acuerdo con un ejemplo adicional de esta divulgación en el que se usa codificación con pérdida, un codificador de vídeo puede calcular el valor de predicción DC $DC_{i,j}$ como:

$$DC_{i,j} = (Q(P_{i,j-1}) + Q(P_{i-1,j}) - Q(P_{i-1,j-1})) \quad (\text{Ec. DC1})$$

Debe tenerse en cuenta que la ecuación DC1 es similar a la ecuación (6) anterior, excepto porque los valores de la muestra (es decir, $P_{i,j-1}$, $P_{i-1,j}$ y $P_{i-1,j-1}$) se cuantifican y, a continuación, descuantifican. Esta divulgación se puede referir a muestras tales como muestras cuantificadas o versiones cuantificadas de muestras originales. Por tanto, en la ecuación DC1, para cada muestra respectiva de un bloque predictivo, el codificador de vídeo puede calcular un valor de predicción DC para la muestra respectiva como una suma de una versión cuantificada de la muestra original de encima de la muestra respectiva (es decir, $Q(P_{i,j-1})$) y una versión cuantificada de la muestra original de la izquierda de la muestra respectiva (es decir, $Q(P_{i-1,j})$), menos una versión cuantificada de la muestra original de encima y de la izquierda de la muestra respectiva (es decir, $Q(P_{i-1,j-1})$). En la ecuación DC1, los términos de la forma $Q(P_{i,j})$ son muestras reconstruidas. El codificador de vídeo puede calcular a continuación la predicción residual como $r_{i,j} = P_{i,j} - DC_{i,j}$. En otras palabras, el valor residual $r_{i,j}$ es igual al valor de muestra $P_{i,j}$ menos el valor de intrapredicción DC correspondiente $DC_{i,j}$. El residuo reconstruido después de la cuantificación y la descuantificación se denota por $Q(r_{i,j})$.

[0087] El ejemplo descrito en el párrafo anterior puede tener algunas propiedades de rendimiento deseables en el lado del decodificador. Por ejemplo, puede ser posible que el decodificador de vídeo 30 calcule los valores de muestra reconstruidos para todas las muestras de una fila (o columna) de un bloque simultáneamente. Por ejemplo, el decodificador de vídeo 30 puede obtener los valores de muestra reconstruidos como:

$$Q(P_{i,j}) = Q(r_{i,j}) + DC_{i,j},$$

o

$$Q(P_{i,j}) = (\sum_{n=0}^j Q(r_{i,j-n})) + Q(P_{i-1,j}) + Q(P_{i,-1}) + Q(P_{i-1,-1}) \quad (\text{Ec. DC2})$$

En la ecuación DC2, $Q(P_{i-1,j})$ denota una muestra reconstruida que se puede recortar adecuadamente. Por ejemplo, para recortar adecuadamente un valor reconstruido con una profundidad de bits de entrada de 8, los valores de $Q(P_{i-1,j})$ se recortan entre 0 y 255.

[0088] Además, en algunos ejemplos, es posible usar versiones no recortadas de $Q(P_{i-j})$. Los otros valores, $Q(P_{i-1,-1})$ y $Q(P_{i-1,j})$ pertenecen a bloques reconstruidos previamente y ya están recortados. En dichos ejemplos, la muestra reconstruida $Q(P_{i-1,j})$ en la ecuación DC2 no está recortada pero se puede recortar adecuadamente sin afectar al rendimiento. La predicción especificada en la ecuación DC1 solo es aproximada si el decodificador de vídeo usa la ecuación DC2 para la reconstrucción. La predicción de la ecuación DC1 solo es aproximada, porque en lugar de $Q(P_{i,j-1})$, que es una versión recortada, se usa una versión no recortada. En dicho caso, la versión no recortada de $Q(P_{i,j-1})$ se puede usar en el lado del codificador también para generar la predicción DC para evitar una deriva entre el

codificador y el decodificador. Es posible usar la versión recortada, pero entonces puede que las muestras se tengan que reconstruir de una en una, viéndose afectado de este modo el rendimiento. Esto se debe a que, en ese caso, la ecuación DC 1 puede tener que usarse para la reconstrucción. Esto significa que la reconstrucción de una muestra puede depender de la realización de la reconstrucción de una muestra de la izquierda de la muestra. Se ha descrito en esta divulgación cómo se puede reconstruir en paralelo una fila de muestras. Se puede seguir un proceso similar para reconstruir todas las muestras de una columna en paralelo. Si se desea menos paralelismo, el término de suma se puede descomponer en trozos más pequeños, reduciendo potencialmente de este modo el rendimiento pero reduciendo el número promedio de operaciones adicionales necesarias para reconstruir una muestra.

[0089] Por tanto, en al menos algunos de los ejemplos de codificación con pérdida descritos anteriormente, un codificador de vídeo puede generar un bloque predictivo. Como parte de la generación del bloque predictivo, el codificador de vídeo puede usar al menos una de una primera muestra reconstruida (por ejemplo, $Q(P_{i-1,j})$) y una segunda muestra reconstruida (por ejemplo, $Q(P_{i,j-1})$) para la predicción DC de una muestra actual del bloque predictivo. La primera muestra reconstruida puede corresponder a una muestra de la izquierda de la muestra actual de la fila actual del bloque predictivo. La segunda muestra reconstruida puede corresponder a una muestra de una fila del bloque predictivo de encima de la fila actual. El codificador de vídeo puede reconstruir un bloque de codificación que se ha codificado usando codificación con pérdida añadiendo muestras del bloque predictivo a unas muestras residuales correspondientes.

[0090] Además, en al menos algunos de los ejemplos de codificación con pérdida descritos anteriormente, el codificador de vídeo puede determinar un valor reconstruido $Q(P_{i,j-1})$ correspondiente a una muestra de encima de la muestra actual. Además, el codificador de vídeo puede determinar un valor reconstruido $Q(P_{i-1,j})$ correspondiente a una muestra de la izquierda de la muestra actual. El codificador de vídeo también puede determinar un valor reconstruido $Q(P_{i-1,j-1})$ correspondiente a una muestra de la izquierda y de encima de la muestra actual. El codificador de vídeo puede calcular un valor de predicción DC $DC_{i,j}$ para la muestra actual $P_{i,j}$ como:

$$DC_{i,j} = (Q(P_{i,j-1}) + Q(P_{i-1,j}) - Q(P_{i-1,j-1})).$$

[0091] En al menos algunos de los ejemplos de codificación con pérdida descritos anteriormente, el codificador de vídeo puede, para cada valor reconstruido respectivo de entre $Q(P_{i,j-1})$, $Q(P_{i-1,j})$ y $Q(P_{i-1,j-1})$, determina el valor reconstruido respectivo de una de las siguientes maneras. En primer lugar, el codificador de vídeo puede determinar el valor reconstruido respectivo como un valor residual descuantificado para una muestra dada (por ejemplo, $Q(r_{i,j})$) más un valor de predicción DC para la muestra correspondiente (por ejemplo, $DC_{i,j}$). En segundo lugar, como se describe en la ecuación DC2, el codificador de vídeo puede determinar $\sum_{n=0}^j Q(r_{i,j-n})$ ruido respectivo como una suma de valores residuales descuantificados para muestras de encima de la muestra dada (por ejemplo, $(P_{i,j-1})$), más un valor reconstruido correspondiente a una muestra de la izquierda de la muestra dada (por ejemplo, $Q(P_{i-1,j})$), más un valor reconstruido correspondiente a una muestra de referencia de encima de una muestra superior de una columna del bloque predictivo que contiene la muestra dada (por ejemplo, $Q(P_{i-1,j-1})$), más un valor reconstruido correspondiente a una muestra de referencia de encima de una muestra superior de una columna del bloque predictivo de la izquierda de la muestra dada (por ejemplo, $Q(P_{i-1,j-1})$). En este ejemplo, la muestra dada corresponde al valor reconstruido respectivo. En algunos de dichos ejemplos, el codificador de vídeo puede recortar el valor reconstruido correspondiente a la muestra de la izquierda de la muestra dada.

[0092] Otro ejemplo de esta divulgación propone una modificación al proceso de predicción para el modo plano. Al observar las ecuaciones para el modo plano (por ejemplo, la ecuación (2)), la predicción vertical puede ser más exacta si se usan las muestras originales de la última fila para realizar una predicción vertical en lugar de usar P_{M-1} . De forma similar, la predicción horizontal puede ser más exacta si se usan las muestras originales de la última columna para realizar una predicción horizontal en lugar de usar $P_{-1,N}$. El uso de las muestras originales en modo intra plano es un motivo básico de uno o más ejemplos de esta divulgación.

[0093] Por ejemplo, un codificador de vídeo (por ejemplo, el codificador de vídeo 20) puede usar la predicción HEVC plana (por ejemplo, la predicción plana como se describe en el HEVC Working Draft 10) para la última fila y columna, es decir, $T_{i,j}$, donde $i = (M - 1)$ o $j = (N - 1)$. El codificador de vídeo puede restar los valores de predicción de los valores de muestra originales para generar residuos para la última fila y columna. Es decir, el codificador de vídeo puede determinar $r_{i,j}$, donde $i = (M - 1)$ o $j = (N - 1)$. El codificador de vídeo puede generar a continuación los valores de predicción $T_{i,j}$, donde $0 \leq i \leq (M - 2)$ y $0 \leq j \leq (N - 2)$, como sigue:

$$\begin{aligned} T_{i,j}^V &= (M - i) * P_{-1,j} + i * P_{M-1,j}, \\ T_{i,j}^H &= (N - j) * P_{i,-1} + j * P_{i,N-1} \text{ y} \\ T_{i,j} &= (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1). \end{aligned} \tag{19}$$

En este ejemplo se supone que $M = N$. Sin embargo, la ampliación de este concepto a bloques rectangulares es fácil. El codificador de vídeo puede generar los residuos restantes, $r_{i,j}$, donde $0 \leq i \leq (M - 2)$ y $0 \leq j \leq (N - 2)$, restando los valores de predicción de los valores de muestra originales. El codificador de vídeo puede codificar entrópicamente todo el bloque de residuos, $r_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, como en HEVC (por ejemplo, HEVC Working Draft 10).

[0094] En este ejemplo, en el lado del descodificador, un descodificador de vídeo (por ejemplo, el descodificador de vídeo 30) puede descodificar entrópicamente todo el bloque de residuos de predicción para generar valores residuales $r_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$. El descodificador de vídeo puede realizar a continuación predicciones HEVC planas para muestras de la última fila y columna del bloque. Es decir, el codificador de vídeo puede determinar $T_{i,j}$, donde $i = (M - 1)$ o $j = (N - 1)$, como se especifica en HEVC (por ejemplo, en HEVC Working Draft 10). Por ejemplo, el codificador de vídeo puede determinar $T_{i,j}$ usando la ecuación (3) anterior. Además, en este ejemplo, el descodificador de vídeo añade valores residuales para la última fila y columna a los valores de predicción anteriores para reconstruir los valores de muestra originales para la última fila y columna. Después de esto, el descodificador de vídeo genera los valores de predicción $T_{i,j}$, donde $0 \leq i \leq (M - 2)$ y $0 \leq j \leq (N - 2)$, exactamente igual que en el lado del codificador anterior. El descodificador de vídeo añade valores residuales $r_{i,j}$, donde $0 \leq i \leq (M - 2)$ y $0 \leq j \leq (N - 2)$, a los valores de predicción para reconstruir los valores de muestra para las posiciones de muestra restantes del bloque.

[0095] En otro ejemplo de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20) usa la predicción HEVC plana (por ejemplo, la predicción plana como se describe en HEVC Working Draft 10) para el elemento de la última fila y columna, es decir, $T_{i,j}$ donde $i = (M - 1)$ y $j = (N - 1)$. En este ejemplo, el codificador de vídeo resta un valor de predicción para una posición del valor de muestra original para la posición para generar un valor residual para esa posición. El codificador de vídeo puede predecir a continuación los elementos de la última fila y columna bilinealmente como:

$$\begin{aligned} T_{M-1,j} &= ((N - j) * P_{M-1,-1} + j * P_{M-1,N-1}) \gg (\log_2 N), \\ T_{i,N-1} &= ((M - i) * P_{-1,N-1} + i * P_{M-1,N-1}) \gg (\log_2 M). \end{aligned} \quad (20)$$

Además, en este ejemplo, el codificador de vídeo genera los valores de predicción $T_{i,j}$, donde $0 \leq i \leq (M - 2)$ y $0 \leq j \leq (N - 2)$, como sigue:

$$\begin{aligned} T_{i,j}^V &= (M - i) * P_{-1,j} + i * P_{M-1,j}, \\ T_{i,j}^H &= (N - j) * P_{i,-1} + j * P_{i,N-1} \text{ y} \\ T_{i,j} &= (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1). \end{aligned} \quad (21)$$

En este ejemplo, el codificador de vídeo genera valores residuales $r_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$ restando los valores de predicción de los valores de muestra originales. Además, en este ejemplo, el codificador de vídeo codifica entrópicamente todo el bloque de residuos, $r_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$ como en HEVC (por ejemplo, HEVC Working Draft 10).

[0096] Otro ejemplo de esta divulgación proporciona un perfeccionamiento de los ejemplos anteriores. En este ejemplo, la distribución de residuos debida a las técnicas de predicción de esta divulgación tiende a ser la inversa de la común en la compresión de vídeo cuando se emplea una transformada. Comúnmente, el residuo tiene valores más altos a frecuencias más bajas, y valores esperados más bajos a frecuencias más altas. Para el residuo proveniente de los ejemplos anteriores, la última fila y columna tienden a tener valores mayores. Un enfoque para mejorar el rendimiento mientras se aprovecha el procedimiento de codificación entrópica diseñado para el residuo de transformada es rotar el residuo proveniente de la predicción en los ejemplos proporcionados anteriormente. Es decir, el residuo se rota 180 grados, por lo que la parte superior izquierda se convierte en la parte inferior derecha, y viceversa. A continuación, este residuo rotado se codifica entrópicamente. En consecuencia, en el descodificador, el residuo se obtiene y a continuación se rota 180 grados.

[0097] En otros ejemplos de esta divulgación, el proceso de predicción plana se modifica como sigue. Para la primera fila y columna, la predicción se realiza como en el caso del modo de HEVC plano (por ejemplo, el modo plano como se describe en HEVC Working Draft 10) para generar $T_{i,j}$ y $r_{i,j}$, $i = 0$ o $j = 0$. En este ejemplo, en lugar de usar el procedimiento HEVC (por ejemplo, el procedimiento descrito en HEVC Working Draft 10, la ecuación (3), etc.) para generar la predicción para la primera fila y columna, se usan otros procedimientos. Por ejemplo, $P_{0,0}$ se puede predecir como $(P_{-1,0} + P_{0,-1} + 1) \gg 1$. Las muestras restantes de la primera columna se pueden predecir usando la muestra izquierda de la misma fila. De forma similar, las muestras restantes de la primera fila se pueden predecir usando la muestra de encima de la misma columna. Para las posiciones restantes, la predicción plana, $T_{i,j}$, donde $1 \leq i \leq (M - 1)$ y $1 \leq j \leq (N - 1)$, se genera como sigue:

$$\begin{aligned}
T_{i,j}^V &= P_{i-1,j} + w_v * (P_{i-1,j} - P_{i-2,j}), \\
T_{i,j}^H &= P_{i,j-1} + w_h * (P_{i,j-1} - P_{i,j-2}) \text{ y} \\
T_{i,j} &= (T_{i,j}^V + T_{i,j}^H + 1) \gg 1.
\end{aligned}
\tag{22}$$

En la ecuación (22), w_v y w_h son pesos. En algunos ejemplos, se usa un valor de 0,5 tanto para w_v como para w_h , ya que el valor de 0,5 se puede implementar como un desplazamiento de bits. Los residuos restantes (es decir, $r_{i,j}$, donde $1 \leq i \leq (M - 1)$ y $1 \leq j \leq (N - 1)$), se generan restando los valores de predicción de los valores de muestra originales. Todo el bloque de residuos (es decir, $r_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$), se puede codificar entrópicamente como en HEVC (por ejemplo, HEVC Working Draft 10).

[0098] En este ejemplo, en el lado del descodificador, un descodificador de vídeo (por ejemplo, el descodificador de vídeo 30) puede descodificar entrópicamente todo el bloque de residuos de predicción para generar valores residuales (es decir, $r_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$). A continuación, el descodificador de vídeo realiza una predicción plana para muestras de la primera fila y columna (es decir, $T_{i,j}$, donde $i = 0$ o $j = 0$) como se especifica en HEVC (por ejemplo, HEVC Working Draft 10) o como se describe anteriormente, por medio de cualquier otro procedimiento usado por el codificador de vídeo. Además, en este ejemplo, el descodificador de vídeo añade los valores residuales para la primera fila y columna a los valores de predicción anteriores para reconstruir los valores de muestra originales para la primera fila y columna. Posteriormente, el descodificador de vídeo genera los valores de predicción (es decir, $T_{i,j}$, donde $1 \leq i \leq (M - 1)$ y $1 \leq j \leq (N - 1)$) exactamente igual al lado del codificador anterior. En este ejemplo, el descodificador de vídeo añade valores residuales (es decir, $r_{i,j}$, donde $1 \leq i \leq (M - 1)$ y $1 \leq j \leq (N - 1)$) a los valores de predicción para reconstruir los valores de muestra para las posiciones restantes del bloque.

[0099] Los párrafos anteriores describen una forma diferente de realizar una predicción plana para una codificación sin pérdida. En un ejemplo adicional de esta divulgación, la columna más a la derecha y la fila inferior se predicen usando un procedimiento de predicción plana como se describe en HEVC (por ejemplo, HEVC Working Draft 10). En este ejemplo, los valores de muestra originales para la columna más a la derecha y la fila inferior se usan a continuación para realizar una predicción plana o angular para unas muestras restantes del bloque. En el caso de la codificación con pérdida cuando se omite la transformada, en lugar de usar valores de muestra originales para la columna más a la derecha y la fila inferior, se pueden usar valores de muestra reconstruidos (cuantificados) para realizar una predicción plana o angular en las muestras restantes.

[0100] En la HEVC (por ejemplo, HEVC Working Draft 10), los valores de predicción DC para la primera fila y columna se filtran (filtrado de predicción DC). De forma similar, para unos modos de intrapredicción horizontal y vertical, la primera fila y columna de los valores de predicción, respectivamente, se filtran (filtrado de gradiente). Cuando los procedimientos descritos anteriormente se aplican a la codificación con pérdida para los modos de intrapredicción DC horizontal o vertical y cuando se omite la transformada, se puede omitir el filtrado de predicción DC o el filtrado de gradiente.

[0101] En otro ejemplo de esta divulgación, en lugar de cambiar el modo de predicción plano, se aplica el mismo concepto a los modos de intrapredicción angular. Para cada modo angular, la última fila y columna se predicen como se especifica en HEVC (por ejemplo, HEVC Working Draft 10, ecuación (3), etc.). A continuación, los valores de muestra originales para la última fila y columna se usan para realizar una intrapredicción además de las muestras de referencia. La FIG. 4 muestra las muestras que se usan para la predicción. Las posiciones sombreadas en la FIG. 4 son las posiciones usadas como muestras de referencia para realizar la predicción. Si, para una dirección de predicción específica y una posición específica, se hubieran usado como valor de predicción muestras de referencia $P_{-1,j}$, donde $N \leq j \leq 2N$ (o su interpolación bilineal), entonces se usan los valores de muestra originales de la columna derecha (es decir, $P_{i,N-1}$, $0 \leq i \leq M - 1$) en su lugar. Si la posición donde el ángulo de predicción intercepta la columna más a la derecha es una fracción, se puede usar una interpolación bilineal o cualquier otro procedimiento de interpolación adecuado. Como ejemplo, se va a considerar un bloque 4×4 . Para el modo de intrapredicción 34, para la muestra (2, 2), la muestra de referencia correspondiente sería $P_{-1,6}$. Por tanto, para el modo 34, la predicción HEVC sería $T_{2,2} = P_{-1,5}$. En cambio, en este ejemplo, $T_{2,2} = P_{1,3}$.

[0102] De forma similar, si para una dirección de predicción específica y una posición específica se hubieran usado como valores de predicción unas muestras de referencia $P_{i,-1}$, $M \leq i \leq 2M$ (o su interpolación bilineal), entonces se usan los valores de muestra originales de la fila inferior (es decir, $P_{M-1,j}$, $0 \leq j \leq N - 1$) en su lugar. Si la posición donde el ángulo de predicción intercepta la fila inferior es una fracción, se puede usar una interpolación bilineal u otra técnica de interpolación adecuada. Como ejemplo, se va a considerar un bloque 4×4 . En este ejemplo, para el modo de intrapredicción 2, para la muestra (2, 1), la muestra de referencia correspondiente es $P_{4,-1}$. Por tanto, la predicción HEVC para el modo 2 sería $T_{2,1} = P_{4,-1}$. En cambio, en este ejemplo, $T_{2,1} = P_{3,0}$.

[0103] En algunos ejemplos adicionales de esta divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden realizar una codificación que usa uno o más valores de muestra originales dentro de un bloque de datos de vídeo para realizar la predicción de otros valores de muestra dentro del bloque. Los valores de muestra originales

pueden corresponder a una última fila y una última columna del bloque (por ejemplo, la fila inferior y la columna más a la derecha del bloque). Como otro ejemplo, los valores de muestra originales pueden corresponder a una primera fila y una primera columna del bloque (por ejemplo, la fila superior y la columna más a la izquierda). La FIG. 4 ilustra un ejemplo de valores de muestra usados para realizar la predicción de otros valores de muestra. En este ejemplo, el

[0104] Los procedimientos de modulación por codificación de impulsos (DPCM) diferencial se proponen en el documento de Lee *et al.*, "AHG7: Residual DPCM for HEVC lossless coding", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 12.^a reunión, Ginebra, CH, 14-23 de enero de 2013, documento n.º JCTVC-L0117 (a continuación en el presente documento, "JCTVC-L0117"). El documento JCTVC-L0117 propuso una mejora en la intrapredicción para los modos horizontal (por ejemplo, modo de intrapredicción 10) y vertical (por ejemplo, modo de intrapredicción 26) en HEVC para la codificación sin pérdida. Esta mejora se denota por DPCM residual. En JCTVC-L0117, la DPCM residual se aplica cuando una CU se codifica sin pérdida. La idea básica de la DPCM residual es usar el píxel de la fila superior para predecir el píxel actual para el modo vertical y usar el píxel de la columna izquierda para predecir el píxel actual para el modo vertical.

[0105] La DPCM residual, como se describe en JCTVC-L0117, se puede describir como sigue. Se va a considerar un bloque de tamaño M (filas) $\times$ N (cols.). Se va a suponer que $r_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$, es la predicción residual después de realizar una intrapredicción como se especifica en HEVC Working Draft 10. Esto se muestra en las FIGS. 5A y 5B. El bloque podría representar cualquier componente (por ejemplo, luma, croma, R, G, B, etc.). En el procedimiento propuesto en JCTVC-L0117, la DPCM residual se aplica a las muestras residuales, de modo que la matriz $M \times N$ modificada $\tilde{R}$ con los elementos $\tilde{r}_{i,j}$ se obtiene como sigue cuando el modo de intrapredicción es vertical:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - r_{(i-1),j}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (23)$$

Cuando el modo de intrapredicción es horizontal, $\tilde{r}_{i,j}$ se obtiene como sigue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - r_{i,(j-1)}, & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (24)$$

En lugar de las muestras residuales originales R , se señalizan las muestras residuales modificadas de $\tilde{R}$ al decodificador de vídeo. Esto puede ser equivalente a usar $P_{i-1,j}$ como predicción para $P_{i,j}$ para el modo de predicción vertical y usar $P_{i,j-1}$ como predicción para $P_{i,j}$ para el modo de predicción horizontal.

[0106] Además, en el procedimiento propuesto en JCTVC-L0117, en el lado del decodificador, cuando el modo de intrapredicción es un modo vertical, las muestras residuales originales se pueden reconstruir después de analizar las muestras residuales modificadas como sigue:

$$r_{i,j} = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1), \quad (25)$$

Cuando el modo de intrapredicción es un modo horizontal, las muestras residuales originales se pueden reconstruir después de analizar las muestras residuales modificadas como sigue:

$$r_{i,j} = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1). \quad (26)$$

[0107] Además, esta divulgación analiza procedimientos para ampliar las técnicas DPCM residuales propuestas en JCTVC-L0117 para modos de intrapredicción horizontal y vertical a otros modos de intrapredicción angular para codificación sin pérdida. Debido a que la codificación es sin pérdida, las muestras vecinas originales (en orden de codificación causal), así como los residuos de predicción correspondientes están disponibles para una predicción (porque se omiten la transformada y la cuantificación).

[0108] En un primer ejemplo, la técnica DPCM residual se puede ampliar a otros modos de intrapredicción angulares. La FIG. 3 muestra las direcciones de intrapredicción para diferentes modos de predicción angulares (de 2 a 34). Ahora se va a considerar un modo entre 22 y 30. Para cada uno de estos modos, la dirección de predicción se puede considerar cercana a la vertical (casi vertical). Los números 22 y 30 son solo ejemplos. También se pueden

5 elegir otros intervalos (por ejemplo, modos de intrapredicción entre 24 y 28). Ahora se va a considerar que el residuo $r_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$, se calcula como se especifica en HEVC (por ejemplo, HEVC Working Draft 10). Es decir, la predicción se realiza de acuerdo con el modo angular y se resta de los valores de muestra originales para obtener el residuo $r_{i,j}$. La DPCM residual se puede aplicar a las muestras residuales exactamente igual que en JCTVC-L0117 para obtener la matriz $M \times N$ modificada $\tilde{R}$ con los elementos $\tilde{r}_{i,j}$ como sigue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - r_{(i-1),j}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (27)$$

10 **[0109]** De forma similar, en este primer ejemplo, se va a considerar un modo de intrapredicción direccional entre 6 y 14. Para cada uno de estos modos de intrapredicción, la dirección de predicción se puede considerar cercana a la horizontal (es decir, casi horizontal). Los números 6 y 14 son solo ejemplos. En otros ejemplos, se pueden usar otros intervalos (por ejemplo, modos de intrapredicción entre 8 y 12). Ahora, se va a considerar que el residuo $r_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$ se calcula como se especifica en HEVC (por ejemplo, HEVC Working Draft 10). Es decir, la predicción se realiza de acuerdo con el modo angular y la predicción se resta de los valores de la muestra originales para determinar el residuo $r_{i,j}$. La DPCM residual se aplica a las muestras residuales exactamente igual que en JCTVC-L0117 para obtener la matriz $M \times N$ modificada $\tilde{R}$ con los elementos $\tilde{r}_{i,j}$ como sigue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - r_{i,(j-1)}, & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (28)$$

20 Como en el caso de JCTVC-L0117, las muestras residuales modificadas de $\tilde{R}$ se señalizan al decodificador.

25 **[0110]** Además, en este primer ejemplo, en el lado del decodificador, las muestras residuales originales se pueden reconstruir después de que se analicen las muestras residuales modificadas como sigue. Cuando el modo de intrapredicción es casi vertical (por ejemplo, modos 22 a 30, ambos incluidos), las muestras residuales originales se pueden reconstruir como:

$$r_{i,j} = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1), \quad (29)$$

30 Cuando el modo de intrapredicción es casi horizontal (por ejemplo, modos 6 a 14, inclusive), las muestras residuales originales se pueden reconstruir como:

$$r_{i,j} = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1). \quad (30)$$

35 Una vez que el residuo $r_{i,j}$ se ha calculado, el residuo $r_{i,j}$ se añade a la predicción realizada de acuerdo con el modo angular para obtener las muestras originales. Cuando se obtiene la predicción para los modos horizontal y vertical, puede ser posible habilitar o inhabilitar la adición de un término de gradiente a la predicción para la primera columna (para el modo vertical) o la primera fila (para el modo horizontal).

40 **[0111]** En un segundo ejemplo donde las técnicas DPCM residuales se amplían a los modos de intrapredicción angular para codificación sin pérdida, para el modo de intrapredicción 18 o los modos de intrapredicción con direcciones de predicción cercanas a la dirección de predicción diagonal hacia abajo y hacia la derecha del modo de intrapredicción 18 (véase la FIG. 3), el residuo se puede modificar de la siguiente manera. En primer lugar, el residuo $r_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$, se puede calcular como se especifica en HEVC Working Draft 10. Es decir, la predicción se puede realizar de acuerdo con el modo angular y la predicción se puede restar de los valores de muestra originales para obtener el residuo $r_{i,j}$. A continuación, se puede determinar una matriz $M \times N$ modificada $\tilde{R}$ con elementos $\tilde{r}_{i,j}$ como sigue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j = 0 \\ r_{i,j} - r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (31)$$

50 **[0112]** En este segundo ejemplo, las muestras residuales modificadas de $\tilde{R}$ se señalizan al decodificador (por ejemplo, el decodificador de vídeo 30). En el lado del decodificador, el decodificador de vídeo puede reconstruir muestras residuales originales después de que el decodificador de vídeo analice las muestras residuales modificadas

como sigue. Cuando la dirección de intrapredicción es cercana a la dirección de predicción diagonal hacia abajo y hacia la derecha, el decodificador de vídeo puede reconstruir las muestras residuales originales como:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j = 0 \\ \tilde{r}_{i,j} + r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (32)$$

[0113] Además, en este segundo ejemplo, una vez que se ha calculado el residuo $r_{i,j}$, $r_{i,j}$ se puede añadir a la predicción realizada de acuerdo con el modo angular para obtener las muestras originales. Al calcular el residuo $r_{i,j}$, puede que $r_{i-1,j-1}$ necesite estar disponible. Esto puede ser siempre así si los verdaderos residuos se calculan fila por fila o columna por columna. Por tanto, puede ser posible calcular el verdadero residuo $r_{i,j}$ para todas las muestras de una fila (o columna) en paralelo.

[0114] En un tercer ejemplo, donde las técnicas DPCM residuales se amplían a los modos de intrapredicción angular para la codificación sin pérdida, para el modo de intrapredicción 34 o los modos de intrapredicción con direcciones de predicción cercanas a la dirección de predicción diagonal hacia abajo y hacia la izquierda del modo de intrapredicción 34 (véase la FIG. 3), el residuo se puede modificar de la siguiente manera. En primer lugar, el residuo $r_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$, se puede calcular como se especifica en HEVC (por ejemplo, HEVC Working Draft 10). Es decir, la predicción se puede realizar de acuerdo con el modo angular y la predicción se puede restar de los valores de muestra originales para determinar el residuo $r_{i,j}$. A continuación, se puede determinar una matriz $M \times N$ modificada $\tilde{R}$ con elementos $\tilde{r}_{i,j}$ como sigue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j = (N-1) \\ r_{i,j} - r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases} \quad (33)$$

[0115] Además, en este tercer ejemplo, las muestras residuales modificadas de $\tilde{R}$ se pueden señalar a un decodificador de vídeo (por ejemplo, el decodificador de vídeo 30). En el lado del decodificador, el decodificador de vídeo puede reconstruir muestras residuales originales después de que el decodificador de vídeo analice las muestras residuales modificadas como sigue. Cuando la dirección de intrapredicción es cercana a la dirección de predicción diagonal hacia abajo y hacia la izquierda (modo 34), el decodificador de vídeo puede reconstruir las muestras residuales originales usando la siguiente ecuación:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j = (N-1) \\ \tilde{r}_{i,j} + r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases} \quad (34)$$

[0116] En este tercer ejemplo, una vez que se ha calculado el residuo $r_{i,j}$, $r_{i,j}$ se puede añadir a la predicción realizada de acuerdo con el modo angular para obtener las muestras originales. Al calcular el residuo $r_{i,j}$, puede que $r_{i-1,j+1}$ necesite estar disponible. Esto siempre puede ser así si los verdaderos residuos se calculan fila por fila. Por tanto, puede ser posible calcular el verdadero residuo $r_{i,j}$ para todas las muestras de una fila en paralelo.

[0117] En un cuarto ejemplo donde las técnicas de DPCM residual se amplían a los modos de intrapredicción angulares para la codificación sin pérdida, para el modo de intrapredicción 2 o los modos de intrapredicción con direcciones de predicción cercanas a la dirección de predicción diagonal hacia arriba y hacia la derecha del modo 2 (véase la FIG. 3), el residuo se puede modificar de la siguiente manera. En primer lugar, el residuo $r_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$ se puede calcular como se especifica en HEVC (por ejemplo, HEVC Working Draft 10). Es decir, la predicción se puede realizar de acuerdo con el modo angular y la predicción se puede restar de los valores de muestra originales para determinar el residuo $r_{i,j}$. A continuación, se puede determinar una matriz $M \times N$ modificada $\tilde{R}$ con elementos $\tilde{r}_{i,j}$ como sigue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = (M-1), 0 \leq j \leq (N-1) \\ r_{i,j}, & 0 \leq i \leq (M-2), j = 0 \\ r_{i,j} - r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases} \quad (35)$$

[0118] En este cuarto ejemplo, las muestras residuales modificadas de $\tilde{R}$ se pueden señalar a un descodificador de vídeo (por ejemplo, el descodificador de vídeo 30). En el lado del descodificador, el descodificador de vídeo puede reconstruir las muestras residuales originales después de que el descodificador de vídeo analice las muestras residuales modificadas como sigue. Cuando la dirección de intrapredicción es cercana a la dirección de predicción diagonal hacia arriba y hacia la derecha (modo 2), el descodificador de vídeo puede reconstruir las muestras residuales originales de acuerdo con la siguiente ecuación:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i = (M-1), 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 0 \leq i \leq (M-2), j = 0 \\ \tilde{r}_{i,j} + r_{i-1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases} \quad (36)$$

[0119] Además, en este cuarto ejemplo, una vez que se ha calculado el residuo $r_{i,j}$, el residuo $r_{i,j}$ se puede añadir a la predicción realizada de acuerdo con el modo angular para obtener las muestras originales. Al calcular el residuo $r_{i,j}$, puede que $r_{i+1,j+1}$ necesite estar disponible. Esto siempre puede así si los verdaderos residuos se calculan columna por columna. Por tanto, puede ser posible calcular el verdadero residuo $r_{i,j}$ para todas las muestras de una columna en paralelo. Los ejemplos primero, segundo, tercero y cuarto donde las técnicas de DPCM residual se amplían a modos de intrapredicción angulares para codificación sin pérdida se pueden emplear simultáneamente siempre que el intervalo de modos de predicción para cada modo de realización no se solape.

[0120] Como se describe en un quinto ejemplo donde las técnicas de DPCM residual se amplían a los modos de intrapredicción angulares para codificación sin pérdida, en el procedimiento de DPCM residual propuesto en JCTVC-L0117, para el modo vertical, el residuo para la primera fila no se modifica. Una observación similar es cierta para el modo horizontal. En ese caso, el residuo para la primera columna no se modifica. Una observación similar es cierta para los modos casi vertical y casi horizontal en el caso del primer ejemplo donde las técnicas de DPCM residual se amplían a modos de intrapredicción angulares para codificación sin pérdida. El quinto ejemplo donde las técnicas de DPCM residual se amplían a los modos de intrapredicción angulares para codificación sin pérdida propone ampliar el concepto de DPCM residual a la primera fila para un modo de intrapredicción vertical o casi vertical y a la primera columna para un modo de intrapredicción horizontal o casi horizontal. Se va a considerar un modo vertical o casi vertical. En este caso, en primer lugar, el residuo $r_{i,j}$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$ se puede calcular como se especifica en HEVC (por ejemplo, HEVC Working Draft 10). Es decir, la predicción se puede realizar de acuerdo con el modo angular y la predicción se puede restar de los valores de muestra originales para determinar el residuo $r_{i,j}$. A continuación, se puede determinar una matriz $M \times N$ modificada $\tilde{R}$ con elementos $\tilde{r}_{i,j}$ como sigue:

$$\tilde{r}_{i,j} = r_{i,j} - r_{(i-1),j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1). \quad (37)$$

En la ecuación (37), $r_{i,j}$ se refiere al residuo del bloque superior. Si el bloque superior no está disponible o si el bloque superior pertenece a una LCU diferente, puede que no sea posible no realizar una DPCM residual en la primera fila.

[0121] En este quinto ejemplo, las muestras residuales modificadas de $\tilde{R}$ se pueden señalar al descodificador de vídeo. En el lado del descodificador, el descodificador de vídeo puede reconstruir las muestras residuales originales después de que el descodificador de vídeo analice las muestras residuales modificadas como sigue. Cuando el modo de intrapredicción es vertical o casi vertical, las muestras residuales originales se pueden reconstruir como sigue:

$$r_{i,j} = r_{-1,j} + \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1). \quad (38)$$

Se puede usar una estrategia similar para los modos horizontal y casi horizontal donde:

$$\tilde{r}_{i,j} = r_{i,j} - r_{i,(j-1)}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1), \text{ y} \quad (39)$$

$$r_{i,j} = r_{i,-1} + \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1). \quad (40)$$

Este enfoque se puede ampliar de manera similar a los ejemplos segundo, tercero y cuarto donde las técnicas de DPCM residual se amplían a los modos de intrapredicción angulares para una codificación sin pérdida como se describe anteriormente.

[0122] En un sexto ejemplo donde las técnicas de DPCM residual se amplían a los modos de intrapredicción angulares para codificación sin pérdida, el procedimiento de DPCM residual propuesto en JCTVC-L0117 también se puede aplicar al modo intra DC (por ejemplo, modo 1 en HEVC) y al modo plano (por ejemplo, modo 0 en HEVC). Por

ejemplo, la predicción de residuo vertical (u horizontal) se puede aplicar después de realizarse la intrapredicción DC. Además, se puede aplicar la predicción de residuo tanto vertical como horizontal: en primer lugar, aplicando la DPCM vertical (horizontal) y, a continuación, aplicando la DPCM horizontal (vertical) a la salida de la primera DPCM.

[0123] Un séptimo ejemplo donde las técnicas de DPCM residual se amplían a los modos de intrapredicción angulares para codificación sin pérdida es similar al sexto ejemplo donde las técnicas de DPCM residual se amplían a los modos de predicción angulares para codificación sin pérdida en los que se pueden aplicar dos o más DPCM al residuo. Por ejemplo, las dos DPCM diagonales descritas en los ejemplos tercero y cuarto donde las técnicas de DPCM residual se amplían a modos de intrapredicción angulares para codificación sin pérdida y al modo plano, y entonces a la DPCM horizontal y vertical.

[0124] Diversas técnicas descritas en esta divulgación también se pueden ampliar a la intracodificación con pérdida cuando se omite la transformada. Por ejemplo, en HEVC (por ejemplo, HEVC Working Draft 10), la transformada se puede omitir para bloques 4x4. En esta divulgación, el término "bloque de omisión de transformada" se aplica a un bloque para el que se omite la aplicación de la transformada. Sin embargo, las técnicas propuestas descritas anteriormente donde las técnicas de DPCM residual se amplían a modos de intrapredicción angulares para codificación sin pérdida se pueden usar para tamaños de bloque mayores si se omite la transformada. Por ejemplo, JCTVC-E145 analizó una forma de ampliar estas técnicas a la intracodificación con pérdida cuando se omite la transformada. En general, la idea propuesta en JCTVC-E145 es que en lugar de usar las muestras de referencia para la intrapredicción, como se muestra en la FIG. 2, los valores de muestra reconstruidos de vecinos causales se pueden usar para realizar una intrapredicción. Sin embargo, esto puede tener un alto coste computacional ya que es necesario repetir el proceso de predicción para cada muestra. Además, no está claro cómo funcionaría esto para los modos de intrapredicción DC y plano.

[0125] Una o más técnicas de esta divulgación están relacionadas con la DPCM residual para codificación con pérdida. Por ejemplo, en algunos ejemplos de esta divulgación, se consideran los modos de intrapredicción casi horizontal y casi vertical para un bloque para el que se omite la transformada. En esta divulgación, un modo casi vertical se puede definir como un modo de intrapredicción donde la dirección de predicción es casi vertical. Todos los modos intrapredicción entre el 22 y el 30 mostrados en la FIG. 3 pueden ser ejemplos de modos casi verticales. De forma similar, un modo casi horizontal se puede definir como un modo de intrapredicción donde la dirección de predicción es casi horizontal. Todos los modos de intrapredicción entre el 6 y el 14 mostrados en la FIG. 3 pueden ser ejemplos de modos de intrapredicción casi horizontal. Además, se va a considerar un bloque de tamaño M (filas) x N (cols.). Se va a suponer que $r_{i,j}$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, es la predicción residual después de realizar la intrapredicción como se especifica en HEVC (por ejemplo, HEVC Working Draft 10). Esto se muestra en las FIGS. 5A y 5B. El bloque puede representar cualquier componente (por ejemplo, luma, croma, R, G, B, etc.).

[0126] Específicamente, la FIG. 5A muestra una dirección de DPCM residual para modos casi verticales. La FIG. 5B muestra una dirección de DPCM residual para modos casi horizontales. Cada cuadrado respectivo de la FIG. 5A y 5B corresponde a una muestra residual respectiva $r_{i,j}$. Las flechas verticales de cada columna de la FIG. 5A muestran la dirección de DPCM residual para los modos casi verticales. Las flechas horizontales de cada fila de la FIG. 5B muestran la dirección de DPCM residual para los modos casi horizontales. Una dirección de DPCM es una dirección (por ejemplo, horizontal o vertical) a lo largo de la cual un codificador de vídeo procesa muestras cuando aplica una DPCM residual.

[0127] Además, se va a considerar un modo casi vertical y se va a suponer que $Q(r_{i,j})$, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$ denotan la versión cuantificada del residuo $r_{i,j}$. El bloque de valores residuales se puede denotar por R . En otras palabras, el residuo $r_{i,j}$ se ha sometido a cuantificación y cuantificación inversa. La DPCM residual se puede aplicar a las muestras residuales para obtener una matriz $M \times N$ $\tilde{R}$, como sigue. La matriz $M \times N$ modificada $\tilde{R}$ con elementos $\tilde{r}_{i,j}$ se puede obtener como sigue cuando el modo de intrapredicción es vertical:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, \quad 0 \leq j \leq (N - 1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), \quad 0 \leq j \leq (N - 1) \end{cases}, \quad (41)$$

Cuando el modo de intrapredicción es horizontal, $\tilde{r}_{i,j}$ se puede obtener como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M - 1), \quad j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}), & 0 \leq i \leq (M - 1), \quad 1 \leq j \leq (N - 1) \end{cases}. \quad (42)$$

[0128] Por tanto, en el ejemplo anterior, cuando el modo de intrapredicción es vertical o casi vertical, los valores residuales modificados de la columna más a la izquierda de $\tilde{R}$ son iguales a los valores residuales correspondientes de la columna más a la izquierda de R . Para cada valor residual respectivo (es decir, $r_{i,j}$) de cada columna de R a la derecha de la columna más a la izquierda de R , el codificador de vídeo 20 puede hacer que un valor residual

modificado correspondiente $\tilde{r}_{i,j}$ sea igual al valor residual respectivo (es decir, $r_{i,j}$) menos un valor residual reconstruido correspondiente a un valor residual que aparece justo a la izquierda del valor residual respectivo (es decir, $Q(r_{(i-1),j})$). De forma similar, cuando el modo de intrapredicción es horizontal o casi horizontal, los valores residuales modificados de la fila superior de $\tilde{R}$ son iguales a los valores residuales correspondientes de la fila superior de R . Sin embargo, para cada valor residual respectivo (es decir, $r_{i,j}$) de cada fila de R de debajo de la fila superior de R , el codificador de vídeo 20 puede hacer que el valor residual modificado correspondiente (es decir, $\tilde{r}_{i,j}$) sea igual al valor residual respectivo (es decir, $r_{i,j}$) menos un valor residual reconstruido correspondiente a un valor residual que aparece justo encima del valor residual respectivo (es decir, $Q(r_{i,(j-1)})$).

[0129] Además, en este ejemplo, la muestra residual modificada, $\tilde{r}_{i,j}$, se cuantifica para generar una versión cuantificada de la muestra residual modificada, $Q(\tilde{r}_{i,j})$. Cuando el modo de intrapredicción es vertical, la muestra residual reconstruida $Q(r_{i,j})$ se calcula como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (43)$$

Esto también se puede escribir como:

$$Q(r_{i,j}) = \sum_{k=0}^i Q(\tilde{r}_{k,j}), \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1).$$

De forma similar, cuando el modo es horizontal, la muestra residual reconstruida $Q(r_{i,j})$ se calcula como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j = 0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}), & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (44)$$

Esto también se puede escribir como:

$$Q(r_{i,j}) = \sum_{k=0}^j Q(\tilde{r}_{i,k}), \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1).$$

[0130] Por tanto, cuando el modo de intrapredicción es vertical o casi vertical, los valores residuales reconstruidos correspondientes a los valores residuales de una columna más a la izquierda de R son iguales a las versiones cuantificadas correspondientes de unos valores residuales modificados de la columna más a la izquierda de $\tilde{R}$. Sin embargo, para cada valor residual respectivo de cada columna de R a la derecha de la columna más a la izquierda de R , el codificador de vídeo 20 hace que el valor residual reconstruido correspondiente sea igual a la suma de la versión cuantificada del valor residual modificado correspondiente de $\tilde{R}$ (es decir, $Q(\tilde{r}_{i,j})$) y el valor residual reconstruido correspondiente al valor residual que aparece justo a la izquierda del valor residual respectivo (es decir, $Q(r_{(i-1),j})$). De forma similar, cuando el modo de intrapredicción es horizontal o casi horizontal, los valores residuales reconstruidos correspondientes a unos valores residuales de la fila superior de R son iguales a unas versiones cuantificadas correspondientes de unos valores residuales modificados de la fila superior de $\tilde{R}$. Sin embargo, para cada valor residual respectivo de R de cada fila de R por debajo de la fila superior de R , el codificador de vídeo 20 puede hacer que el valor residual reconstruido correspondiente sea igual a una suma de la versión cuantificada del valor residual modificado correspondiente de $\tilde{R}$ (es decir, $Q(\tilde{r}_{i,j})$) y un valor residual reconstruido correspondiente al valor residual de R que aparece justo encima del valor residual respectivo (es decir, $Q(r_{i,(j-1)})$).

[0131] En el lado del descodificador, los cálculos (43) y (44) se repiten para generar la muestra residual reconstruida $Q(r_{i,j})$, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$. El descodificador de vídeo 30 puede añadir valores de muestra residual reconstruidos a los valores de predicción originales para generar valores de muestra reconstruidos. Por ejemplo, el descodificador de vídeo 30 puede añadir las muestras residuales reconstruidas $Q(r_{i,j})$ a unas muestras correspondientes de un bloque predictivo para reconstruir unos valores de muestra de un bloque actual (por ejemplo, una CU actual). El proceso de determinar valores residuales reconstruidos a partir de valores residuales que se codifican usando DPCM se puede denominar en el presente documento "RDPCM inversa". Normalmente, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 recortan las muestras reconstruidas hasta una profundidad de bits apropiada. Sin embargo, en este caso, la operación de recorte se realiza después del proceso de RDPCM inversa para reconstruir los residuos cuantificados.

[0132] Los ejemplos anteriores también se pueden ampliar de una manera fácil a las direcciones diagonales. Por ejemplo, en un ejemplo, cuando la dirección de intrapredicción es cercana a una dirección diagonal hacia abajo y hacia la derecha (por ejemplo, el modo de intrapredicción 18), $\tilde{r}_{i,j}$ se puede determinar como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j = 0 \\ r_{i,j} - Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (45)$$

La muestra residual reconstruida $Q(r_{i,j})$ se puede determinar como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N-1) \\ Q(\tilde{r}_{i,j}), & 1 \leq i \leq (M-1), j = 0 \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (46)$$

Otros modos casi diagonales (por ejemplo, los modos cercano a la diagonal hacia abajo y hacia la izquierda (modo 34) y cercano a la diagonal hacia arriba y hacia la derecha (modo 2)) se pueden tratar de forma similar.

[0133] Además, en algunos ejemplos, en lugar de aplicar la DPCM a unos residuos reconstruidos, la DPCM se puede aplicar a unas versiones con desplazamiento de bits de los residuos reconstruidos. Por ejemplo, para bloques de omisión de transformada, el decodificador de vídeo 30 puede realizar, para cada muestra, una operación de descuantificación, un desplazamiento a la izquierda de 7 y luego un desplazamiento a la derecha (por ejemplo, después de añadir un desplazamiento) de 20 bits de profundidad. En este contexto, la profundidad de bits se puede referir al número de bits usados para representar un valor de muestra o un coeficiente de transformada. En una operación de descuantificación, hay un desplazamiento a la derecha de bdShift después de añadir un desplazamiento.

[0134] Por ejemplo, la sección 8.6.2 de JCTVC-M1005_v2 (es decir, un borrador de la especificación de ampliación de alcance para HEVC) describe un proceso de escalado y transformada que determina una matriz de muestras residuales en base a una matriz de niveles de coeficientes de transformada de un bloque de transformada de una TU. En la sección 8.6.2 de JCTVC-M1005_v2, la matriz de niveles de coeficientes de transformada puede contener valores de muestra residuales si la transformada y la cuantificación no se aplican al bloque de transformada. A continuación, se reproduce una parte de la sección 8.6.2 de JCTVC-M1005_v2.

[0135] La matriz (nTbS)x(nTbS) de muestras residuales r se deriva como sigue:

- Si cu_transquant_bypass_flag es igual a 1, se hace que la matriz (nTbS)x(nTbS) r sea igual a la matriz (nTbS)x(nTbS) de coeficientes de transformada TransCoeffLevel[xTbY][yTbY][cldx].

- De lo contrario, se aplican las siguientes etapas ordenadas:

1.El proceso de escalado para coeficientes de transformada como se especifica en la subcláusula 8.6.3 se invoca con la ubicación del bloque de transformada (xTbY, yTbY), el tamaño del bloque de transformada nTbS, la variable de componente de color cldx y el parámetro de cuantificación qp como entradas, y la salida es una matriz (nTbS)x(nTbS) de coeficientes de transformada escalados d.

2.La matriz (nTbS)x(nTbS) de muestras residuales r se deriva como sigue:

- Si transform_skip_flag[xTbY][yTbY][cldx] es igual a 1, los valores de la matriz de muestras residuales r[x][y] con x = 0..nTbS - 1, y = 0..nTbS - 1 se derivan como sigue:

$$r[x][y] = (d[x][y] \ll 7) \quad (8-267)$$

- De lo contrario (si transform_skip_flag[xTbY][yTbY][cldx] es igual a 0), el proceso de transformación para coeficientes de transformada escalados como se especifica en la subcláusula 8.6.4 se invoca con la ubicación del bloque de transformada (xTbY, yTbY), el tamaño del bloque de transformada nTbS, la variable de componente de color cldx y la matriz (nTbS)x(nTbS) de coeficientes de transformada escalados d como entradas, y la salida es una matriz (nTbS)x(nTbS) de muestras residuales r.

3. La variable bdShift se deriva como sigue:

$$bdShift = (cldx == 0) ? 20 - BitDepth_Y : 20 - BitDepth_C \quad (8-268)$$

4. Los valores de muestra residual $[x][y]$ con $x = 0..nTbS - 1$ e $y = 0..nTbS - 1$ se modifican como sigue:

$$r[x][y] = (r[x][y] + (1 \ll (bdShift - 1))) \gg bdShift \quad (8-269)$$

[0136] Como se muestra anteriormente, la sección 8.6.2 de JCTVC-M1005_v2 especifica que la sección 8.6.3 de JCTVC-M1005_v2 se invoca si la transformada y la cuantificación se aplican al bloque de transformada. La sección 8.6.3 de JCTVC-M1005_v2 describe un proceso de escalado para coeficientes de transformada. A continuación, se reproduce la sección 8.6.3 de JCTVC-M1005_v2.

[0137] Las entradas para este proceso son:

- una ubicación de luma ($xTbY$, $yTbY$) que especifica la muestra superior izquierda del bloque de transformada de luma actual en relación con la muestra de luma superior izquierda de la imagen actual,
- una variable $nTbS$ que especifica el tamaño del bloque de transformada actual,
- una variable $cldx$ que especifica el componente de color del bloque actual,
- una variable qP que especifica el parámetro de cuantificación.

[0138] La salida de este proceso es la matriz $(nTbS) \times (nTbS)$ d de coeficientes de transformada escalados con elementos $d[x][y]$.

[0139] La variable $bdShift$ se deriva como sigue:

- Si $cldx$ es igual a 0,

$$bdShift = BitDepth_Y + \log_2(nTbS) - 5 \quad (8-270)$$

- De lo contrario,

$$bdShift = BitDepth_C + \log_2(nTbS) - 5 \quad (8-271)$$

La lista $levelScale[]$ se especifica como $levelScale[k] = \{ 40, 45, 51, 57, 64, 72 \}$ con $k = 0..5$.

[0140] Para la derivación de los coeficientes de transformada escalados $d[x][y]$ con

$x = 0..nTbS - 1$, $y = 0..nTbS - 1$, se aplica lo siguiente:

- El factor de escala $m[x][y]$ se deriva como sigue:
- Si $scaling_list_enabled$ flag es igual a 0,

$$m[x][y] = 16 \quad (8-272)$$

- De lo contrario (si $scalimg_list_enabled_flag$ es igual a 1),

$$m[x][y] = ScalingFactor[sizeId][matrixId][x][y] \quad (8-273)$$

Donde $sizeId$ se especifica en la tabla 7-3 para el tamaño de la matriz de cuantificación igual a $(nTbS) \times (nTbS)$ y $matrixId$ se especifica en la tabla 7-4 para $sizeId$,

[0141] $CuPredMode[xTbY][yTbY]$ y $cldx$, respectivamente.

- El coeficiente de transformada escalado $d[x][y]$ se deriva como sigue:

$$d[x][y] = \text{Clip3}(-32768, 32767, ((\text{TransCoeffLevel}[xTbY][yTbY][cIdx][x][y] * m[x][y] * \text{levelScale}[qP \% 6] << (qP / 6)) + (1 << (bdShift - 1))) >> bdShift)$$

(8-274)

[0142] Como se indica en la sección 8.6.3 de JCTVC-M1005_v2, el decodificador de vídeo 30 puede determinar un coeficiente de transformada escalado, $d[x][y]$, usando la ecuación 8-274 de JCTVC-M1005_v2. En la ecuación 8-274, la función $\text{Clip3}(\dots)$ se define como:

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; z < x \\ y & ; z > y \\ z & ; \text{en otros casos} \end{cases}.$$

[0143] De acuerdo con una o más técnicas de esta divulgación, en lugar de aplicar la DPCM inversa a los residuos reconstruidos, la DPCM inversa se puede aplicar a los valores de muestra descuantificados antes de aplicar el desplazamiento a la izquierda. De forma similar, en lugar de una transformada inversa, se puede aplicar un desplazamiento a la izquierda de 7 después de la descuantificación. Por ejemplo, una matriz de muestras residuales $r[x][y]$ con $x = 0..nTbS-1$, $y = 0..nTbS-1$ se puede obtener como se muestra en la ecuación 8-267 de JCTVC-M1005_v2.

[0144] En algunos ejemplos, la DPCM inversa se aplica a las muestras después de la descuantificación o después del desplazamiento a la izquierda de 7. De forma similar, se aplica un desplazamiento a la derecha (una profundidad de 20 bits) después de añadir un desplazamiento, justo al final del proceso para reconstruir los residuos. Por ejemplo, las muestras residuales $r[x][y]$, con $x = 0..nTbS-1$ e $y = 0..nTbS-1$, se pueden modificar como se muestra en la ecuación 8-269 de JCTVC-M1005_v2. En ese caso, la DPCM inversa se puede aplicar después del desplazamiento a la derecha en una cantidad menor que una profundidad de 20 bits para retener una mejor precisión, antes de aplicar el desplazamiento a la derecha restante para que el desplazamiento a la derecha total sea equivalente a 20 bits de profundidad.

[0145] En algunos casos, el codificador de vídeo 20 no aplica una transformada a las muestras residuales de un bloque de transformada, pero sí cuantifica las muestras residuales del bloque de transformada. En otras palabras, el codificador de vídeo 20 puede aplicar una forma de codificación con pérdida en la que se omite la transformada. En dichos casos, el decodificador de vídeo 30 puede determinar, a partir de los elementos de sintaxis de un flujo de bits, muestras residuales cuantificadas del bloque de transformada y descuantificar las muestras residuales cuantificadas del bloque de transformada para reconstruir las muestras residuales del bloque de transformada sin aplicar una transformada inversa al bloque de transformada.

[0146] De forma alternativa, el filtrado de gradiente se puede modificar como se describe en el presente documento usando valores de muestra reconstruidos (cuantificados) en lugar de valores de muestra originales. Cuando en HEVC (por ejemplo, HEVC Working Draft 10) se obtiene la predicción para los modos horizontal y vertical, se añade un término de gradiente a la predicción para la primera columna (para el modo vertical) o la primera fila (para el modo horizontal). Esto se puede denominar filtrado de gradiente. La adición del término de gradiente se podría habilitar o inhabilitar cuando se usa DPCM residual. Esta divulgación propone mejoras en el filtrado de gradiente para los modos de intrapredicción horizontal y vertical cuando se usa la predicción residual.

[0147] Por ejemplo, se va a considerar que se usa DPCM residual como se propone en JCTVC-L0117. En HEVC Working Draft 10, para componentes de luma y para tamaños de bloque inferiores a 32, para el modo de intrapredicción vertical, la predicción para la muestra $P_{i,0}$, $0 \leq i \leq M-1$ viene dada por:

$$\text{Clip}\left(P_{-1,0} + \left((P_{i,-1} - P_{-1,-1}) \gg 1\right)\right),$$

En la ecuación anterior, $\gg$ representa un desplazamiento a la derecha a nivel de bit y la operación Clip recorta los valores de predicción hasta el intervalo de valores de muestra. Debido a que en la DPCM residual, para el modo de intrapredicción vertical, la predicción para la muestra $P_{i,j}$ es $P_{i+1,j}$, el término de gradiente para las muestras de la primera columna se modifica como $((P_{i,-1} - P_{i+1,-1}) \gg 1)$. En consecuencia, la predicción para la muestra $P_{i,0}$, $0 \leq i \leq M-1$ es:

$$\text{Clip}\left(P_{i-1,0} + \left((P_{i,-1} - P_{i-1,-1}) \gg 1\right)\right).$$

[0148] Además, el filtrado de gradiente se puede ampliar a otras columnas para el modo de intrapredicción vertical como sigue. Debido a que las muestras originales izquierda y superior izquierda están disponibles en modo sin pérdida, el término de gradiente se puede añadir a la predicción para cualquier columna en el modo de intrapredicción vertical. Por tanto, para el modo de intrapredicción vertical, la predicción para la muestra $P_{i,j}$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ se puede modificar como:

$$\text{Clip}\left(P_{i-1,j} + \left((P_{i,j-1} - P_{i-1,j-1}) \gg 1\right)\right).$$

El filtrado de gradiente modificado para el modo de intrapredicción vertical se puede aplicar a cualquier componente y cualquier tamaño de bloque.

[0149] Este concepto se puede ampliar también al modo de intrapredicción horizontal. En HEVC Working Draft 10, para componentes de luma y para tamaños de bloque inferiores a 32, para el modo de intrapredicción horizontal, la predicción para la muestra $P_{0,j}$, $0 \leq j \leq N-1$ viene dada por:

$$\text{Clip}\left(P_{0,-1} + \left((P_{-1,j} - P_{-1,-1}) \gg 1\right)\right),$$

En la ecuación anterior, $\gg$ representa un desplazamiento a la derecha a nivel de bit y la operación Clip recorta los valores de predicción hasta el intervalo de valores de muestra. Debido a que en la DPCM residual, para el modo de intrapredicción horizontal, la predicción para la muestra $P_{i,j}$ es $P_{i,j-1}$, el término de gradiente para las muestras de la primera fila se puede modificar como $((P_{-1,j} - P_{-1,j-1}) \gg 1)$. La predicción para la muestra $P_{0,j}$, $0 \leq j \leq N-1$ se puede determinar mediante:

$$\text{Clip}\left(P_{0,j-1} + \left((P_{-1,j} - P_{-1,j-1}) \gg 1\right)\right).$$

[0150] El filtrado de gradiente se puede ampliar a otras columnas para el modo de intrapredicción horizontal como sigue. Debido a que las muestras originales superior y superior izquierda siempre están disponibles en modo sin pérdida, el término de gradiente se puede añadir a la predicción para cualquier fila en el modo de intrapredicción horizontal. Por tanto, para el modo de intrapredicción horizontal, la predicción para la muestra $P_{i,j}$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ puede venir dada por:

$$\text{Clip}\left(P_{i,j-1} + \left((P_{i-1,j} - P_{i-1,j-1}) \gg 1\right)\right).$$

[0151] Como se describe en la sección 7.3.2.3 de JCTVC-M1005_v2, un PPS puede incluir un elemento de sintaxis habilitado para ocultación de datos de signo (por ejemplo, `sign_data_hiding_enabled_flag`). Además, como se muestra en la sección 7.3.8.11 de JCTVC-M1005_v2, cuando el elemento de sintaxis habilitado para ocultación de datos de signo tiene un valor en particular (por ejemplo, 1), los elementos de sintaxis (por ejemplo, `coeff_sign_flag`) que indican signos positivos/negativos de coeficientes de transformada se pueden omitir de una TU. Por tanto, el codificador de vídeo 20 puede señalar, en un flujo de bits, una indicación explícita (por ejemplo, `sign_data_hiding_enabled_flag`) de que la ocultación de datos de signo está habilitada para una imagen actual y, por consiguiente, un bloque actual dentro de la imagen actual. Del mismo modo, el decodificador de vídeo 30 puede obtener, de un flujo de bits, una indicación explícita (por ejemplo, `sign_data_hiding_enabled_flag`) de que la ocultación de datos de signo está habilitada para una imagen actual y, por consiguiente, un bloque actual de la imagen actual. A continuación, se reproduce la sección 7.3.8.11 de JCTVC-M1005_v2.

<code>residual_coding(x0, y0, log2TrafoSize, cldx) {</code>	Descriptor
<code>if(transform_skip_enabled_flag && !cu_transquant_bypass_flag && (log2TrafoSize == 2))</code>	
<code>transform_skip_flag[x0][y0][cldx]</code>	<code>ae(v)</code>
<code>last_sig_coeff_x_prefix</code>	<code>ae(v)</code>
<code>last_sigcoeff_y_prefix</code>	<code>ae(v)</code>
<code>if(last_sig_coeff_x_prefix > 3)</code>	
<code>last_sig_coeff_x_suffix</code>	<code>ae(v)</code>
<code>if(last_sig_coeff_y_prefix > 3)</code>	

residual_coding(x0, y0, log2TrafoSize, cldx) {	Descriptor
last_sig_coeffy_suffix	ae(v)
lastScanPos = 16	
lastSubBlock = (1 << (log2TrafoSize - 2)) * (1 << (log2TrafoSize - 2)) - 1	
do {	
if(lastScanPos == 0) {	
lastScanPos = 16	
lastSubBlock- -	
}	
lastScanPos- -	
xS = ScanOrder[log2TrafoSize - 2][scanIdx][lastSubBlock][0]	
yS = ScanOrder[log2TrafoSize - 2][scanIdx][lastSubBlock][1]	
xC = (xS << 2) + ScanOrder[2][scanIdx][lastScanPos][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
for(i = lastSubBlock; i >= 0; i- -) {	
xS = ScanOrder[log2TrafoSize - 2][scanIdx][i][0]	
yS = ScanOrder[log2TrafoSize - 2][scanIdx][i][1]	
inferSbDcSigCoeffFlag = 0	
if((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag[xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
for(n = (i == lastSubBlock) ? lastScanPos - 1 : 15; n >= 0; n- -) {	
xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
}	
}	
firstSigScanPos = 16	
lastSigScanPos = -1	
numGreater1Flag = 0	
lastGreater1ScanPos = -1	
for(n = 15; n >= 0; n- -) {	
xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]	
if(sig_coeff_flag[xC][yC]) {	
if(numGreater1Flag < 8) {	
coeff_abs_level_greater1_flag[n]	ae(v)
numGreater1Flag++	
if(coeff_abs_level_greater1_flag[n] && lastGreater1ScanPos == -1)	

residual_coding(x0, y0, log2TrafoSize, cldx) {	Descriptor
lastGreater1ScanPos = n	
}	
if(lastSigScanPos == -1)	
lastSigScanPos = n	
firstSigScanPos = n	
}	
}	
signHidden = (lastSigScanPos - firstSigScanPos > 3 && !cu_transquant_bypass_flag)	
if(lastGreater1ScanPos != -1)	
coeff_abs_level_greater2_flag [lastGreater1ScanPos]	ae(v)
for(n = 15; n >= 0; n- -) {	
xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]	
if(sig_coeff_flag[xC][yC] && (!sign_data_hiding_enabled_flag !signHidden (n != firstSigScanPos)))	
coeff_sign_flag [n]	ae(v)
}	
numSigCoeff = 0	
sumAbsLevel = 0	
for(n = 15; n >= 0; n- -) {	
xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]	
if(sig_coeff_flag[xC][yC]){	
baseLevel = 1 + coeff_abs_level_greater1_flag[n] + coeff_abs_level_greater2_flag[n]	
if(baseLevel == ((numSigCoeff < 8) ? ((n == lastGreater1ScanPos) ? 3 : 2) : 1))	
coeff_abs_level_remaining [n]	ae(v)
TransCoeffLevel[x0][y0][cldx][xC][yC]	=
(coeff_abs_level_remaining[n]+baseLevel) * (1 - 2 * coeff_sign_flag[n])	
if(sign_data_hiding_enabled_flag && signHidden) {	
sumAbsLevel += (coeff_abs_level_remaining[n] + baseLevel)	
if((n == firstSigScanPos) && ((sumAbsLevel % 2) == 1))	
TransCoeffLevel[x0][y0][cldx][xC][yC]	=
TransCoeffLevel[x0][y0][cldx][xC][yC]	-
}	
numSigCoeff++	
}	
}	
}	
}	

[0152] Como se especifica en la sección 7.4.3.3 de JCTVC-M1005_v2, sign_data_hiding_enabled_flag igual a 0 especifica que la ocultación de bits de signo (es decir, la ocultación de datos de signo) está inhabilitada. sign_data_hiding_enabled_flag igual a 1 especifica que la ocultación de bits de signo está habilitada. Esta divulgación puede usar los términos "ocultación de bit de signo" y "ocultación de datos de signo" de manera intercambiable. Además, un codificador de vídeo puede organizar elementos de sintaxis que representan coeficientes de transformada de acuerdo con un orden de barrido de subbloques. En el orden de barrido de subbloques, el codificador de vídeo

divide un bloque de coeficientes de transformada en subbloques, cada uno con una anchura de 4 coeficientes de transformada y una altura de 4 coeficientes de transformada. El codificador de vídeo puede determinar una variable de signo oculto (es decir, `signHidden`) para cada uno de los subbloques. Cuando la diferencia en las posiciones entre un primer coeficiente de transformada significativo (es decir, no cero) en un subbloque 4x4 de un bloque de coeficientes de transformada y un último coeficiente de transformada significativo en el subbloque 4x4 del bloque de coeficientes de transformada es mayor que 3 (es decir, `lastSignScanPos - firstSigScanPos > 3`) y las etapas de transformada y cuantificación no se excluyen (es decir, se omiten) para la CU actual (es decir, `!cu_transquant_bypass_flag`), el codificador de vídeo puede determinar que la variable de signo oculto (es decir, `signHidden`) para el subbloque 4x4 es igual a 1. Si la transformada se omite o se excluye, los coeficientes de transformada en realidad son residuos de predicción.

[0153] Para cada coeficiente de transformada respectivo del subbloque 4x4, un elemento de sintaxis de signo (por ejemplo, `coeff_sign_flag[n]`) para el coeficiente de transformada respectivo se señala en el flujo de bits si el coeficiente de transformada respectivo es significativo (es decir, distinto de cero) y si `sign_data_hiding_enabled_flag` especifica que la ocultación de bits de signo no está habilitada, la variable de signo oculto para el subbloque 4x4 no es igual a 1 o el coeficiente de transformada respectivo no es el primer coeficiente de transformada significativo del subbloque 4x4. Por ejemplo, el signo de un coeficiente de transformada significativo no se señala si `sign_data_hiding_enabled_flag` indica que la ocultación de bits de signo está habilitada, la diferencia de posición entre el primer coeficiente significativo del subbloque 4x4 y el último coeficiente significativo del subbloque 4x4 es mayor que 3, el subbloque 4x4 no se generó usando codificación de exclusión de transformada y cuantificación (por ejemplo, codificación sin pérdida), y el coeficiente de transformada respectivo es el primer coeficiente de transformada significativo del subbloque 4x4.

[0154] Como se especifica en la sección 7.4.3.3 de JCTVC-M1005_v2, `coeff_sign_flag[n]` especifica el signo de un nivel de coeficiente de transformada para la posición de barrido `n` como sigue. Si `coeff_sign_flag[n]` es igual a 0, el nivel del coeficiente de transformada correspondiente tiene un valor positivo. De lo contrario (si `coeff_sign_flag[n]` es igual a 1), el nivel del coeficiente de transformada correspondiente tiene un valor negativo. Cuando `coeff_sign_flag[n]` no está presente, se deduce que `coeff_sign_flag[n]` es igual a 0.

[0155] Cuando se habilita la ocultación de datos de signo, el codificador de vídeo 20 puede incluir datos que indican el signo de un coeficiente de transformada/muestra residual en el propio valor del coeficiente de transformada/valor de muestra residual. Por ejemplo, el codificador de vídeo 20 puede modificar uno o más bits que representan un coeficiente de transformada/muestra residual de modo que el decodificador de vídeo 30 puede usar la información de paridad para determinar si el primer coeficiente significativo de un subbloque 4 × 4 es positivo o negativo. Por ejemplo, si la suma de los valores absolutos de los coeficientes de transformada de un subbloque 4x4 es par, se deduce que el signo es positivo; de lo contrario, se deduce que el signo es negativo. Típicamente, el codificador de vídeo 20 puede cambiar un bit menos significativo de la representación de uno de los coeficientes de transformada/las muestras residuales cuantificados en un subbloque 4x4. En consecuencia, un usuario puede no ser capaz de percibir ninguna pérdida de calidad visual debido a que el codificador de vídeo 20 modifica un bit de representación del coeficiente de transformada/de la muestra residual para indicar el signo del coeficiente de transformada/de la muestra residual.

[0156] Cuando se usan las técnicas descritas de esta divulgación para codificación con pérdida, puede ser difícil para un codificador de vídeo implementar la ocultación de datos de signo para bloques a los que se aplican las técnicas descritas. Estos son bloques intra para los que se omite la transformada y los modos son planos y/o DC y/o los modos para los que se aplica DPCM residual. El codificador de vídeo 20 puede introducir errores cambiando valores de bits en la representación del coeficiente de transformada/de la muestra residual para indicar el signo del coeficiente de transformada/de la muestra residual. Además, usando estas técnicas, el error de una muestra residual modificada se puede propagar a muestras residuales posteriores. Dichos errores se pueden agravar cuando se aplica una DPCM residual porque la DPCM residual puede depender del decodificador de vídeo 30 que suma entre sí múltiples coeficientes de transformada/muestras residuales para determinar coeficientes de transformada/muestras residuales. Por tanto, la ocultación de datos de signo en realidad puede dar como resultado una degradación del rendimiento. En dichos casos, la ocultación de datos de signo se puede inhabilitar normativamente. Esto significa que, aunque se indique en un flujo de bits (o como una opción predeterminada) que se está usando la ocultación de datos de signo, la ocultación de datos de signo se inhabilita para determinados bloques.

[0157] En un ejemplo, un codificador de vídeo puede inhabilitar la ocultación de datos de signo para un bloque si el codificador de vídeo no aplica una transformada al bloque, si el bloque se intrapredice usando un modo de intrapredicción plano o un modo de intrapredicción DC, o si el bloque se intrapredice usando un modo de intrapredicción para el que se aplica una DPCM residual. Por ejemplo, en algunos ejemplos, la ocultación de datos de signo se puede inhabilitar cuando se omite la transformada y el bloque se intracodifica y el modo intra es una intrapredicción plana. Además, en algunos ejemplos, la ocultación de datos de signo se puede inhabilitar cuando se omite la transformada y el bloque se intracodifica y el modo intra es una intrapredicción DC. Además, en algunos ejemplos, la ocultación de datos de signo se puede inhabilitar cuando se omite la transformada y el bloque se intracodifica y el modo intra es un modo para el que se aplica una DPCM residual.

[0158] Por ejemplo, el codificador de vídeo 20 puede determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera usando codificación con pérdida sin la aplicación de una transformada (por ejemplo, una transformada de coseno discreta, transformada direccional u otra transformada) a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una DPCM residual. Cuando la ocultación de datos de signo está inhabilitada para el bloque actual, el codificador de vídeo 20 puede incluir, en el flujo de bits, para cada valor significativo respectivo del bloque actual, un elemento de sintaxis respectivo que indica si el valor significativo respectivo es positivo o negativo. De forma similar, en este ejemplo, el descodificador de vídeo 30 puede determinar que la ocultación de datos de signo está inhabilitada para el bloque actual si el bloque actual se genera usando codificación con pérdida sin la aplicación de una transformada (por ejemplo, una transformada de coseno discreta, una transformada direccional u otra transformada) a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una DPCM residual. En dichos ejemplos, cuando la ocultación de datos de signo está inhabilitada para el bloque actual, el descodificador de vídeo 30 obtiene, del flujo de bits, para cada valor significativo respectivo del bloque, un elemento de sintaxis respectivo que indica si el valor significativo respectivo es positivo o negativo.

[0159] En otros ejemplos, la ocultación de datos de signo puede estar inhabilitada para un bloque cuando se omite la transformada, el bloque se intracodifica o para todos los bloques con omisión de transformada. Por tanto, en este ejemplo, la ocultación de datos de signo puede estar inhabilitada independientemente del modo de intrapredicción que se usa. Por ejemplo, un codificador de vídeo puede determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se codifica sin la aplicación de la transformada a unos datos residuales del bloque actual, y el bloque actual se intracodifica usando un modo de intrapredicción DC o un modo de intrapredicción plano.

[0160] Cabe destacar que esta divulgación ha analizado técnicas para DPCM residual y modificaciones a los modos DC y plano. Se puede usar una cualquiera o una combinación de estas técnicas.

[0161] La FIG. 6 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas de esta divulgación. La FIG. 6 se proporciona con propósitos explicativos y no se debería considerar limitativa de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

[0162] En el ejemplo de la FIG. 6, el codificador de vídeo 20 incluye una unidad de procesamiento de predicción 100, una unidad de generación de residuos 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de procesamiento de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 114, una memoria intermedia de imágenes descodificadas 116 y una unidad de codificación entrópica 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de interpredicción 120 y una unidad de procesamiento de intrapredicción 126. La unidad de procesamiento de interpredicción 120 incluye una unidad de estimación de movimiento 122 y una unidad de compensación de movimiento 124. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

[0163] El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en un sector de una imagen de los datos de vídeo. Cada una de las CTU puede estar asociada a bloques de árbol de codificación (CTB) de luma de igual tamaño y a unos CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar una división de árbol cuaternario para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de las CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un CTB asociado a una CTU en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subbloques de igual tamaño, y así sucesivamente.

[0164] El codificador de vídeo 20 puede codificar las CU de una CTU para generar representaciones codificadas de las CU (es decir, CU codificadas). Como parte de la codificación de una CU, la unidad de procesamiento de predicción 100 puede dividir los bloques de codificación asociados a la CU entre una o más PU de la CU. Por tanto, cada PU puede estar asociada a un bloque de predicción de luma y a unos bloques correspondientes de predicción de croma. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir PU que tienen diversos tamaños. Como se indica anteriormente, el tamaño de una CU se puede referir al tamaño del bloque de codificación de luma de la CU, y el tamaño de una PU se puede referir al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es de $2N \times 2N$, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir tamaños de PU de $2N \times 2N$ o $N \times N$ para intrapredicción, y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para interpredicción. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir una división asimétrica para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para interpredicción.

[0165] La unidad de procesamiento de interpredicción 120 puede generar datos predictivos para una PU realizando una interpredicción en cada PU de una CU. Los datos predictivos para la PU pueden incluir bloques de muestras predictivos (es decir, bloques predictivos) de la PU e información de movimiento para la PU. La unidad de

interpredicción 121 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un sector I, en un sector P o en un sector B. En un sector I, todas las PU se intrapredicen. Por lo tanto, si la PU está en un sector I, la unidad de interpredicción 121 no realiza interpredicción en la PU. Por tanto, para bloques codificados en modo I, el bloque predictivo se forma usando predicción espacial a partir de bloques vecinos previamente codificados dentro de la misma trama.

[0166] Si una PU está en un sector P, la unidad de estimación de movimiento 122 puede buscar en las imágenes de referencia de una lista de imágenes de referencia (por ejemplo, "RefPicList0") una región de referencia para la PU. La región de referencia para la PU puede ser una región, dentro de una imagen de referencia, que contiene muestras (por ejemplo, bloques de muestras) que se corresponden más estrechamente con los bloques de muestras de la PU. La unidad de estimación de movimiento 122 puede generar un índice de referencia que indica una posición en RefPicList0 de la imagen de referencia que contiene la región de referencia para la PU. Además, la unidad de estimación de movimiento 122 puede generar un vector de movimiento que indica un desplazamiento espacial entre un bloque de codificación de la PU y una ubicación de referencia asociada con la región de referencia. Por ejemplo, el vector de movimiento puede ser un vector bidimensional que proporciona un desplazamiento desde las coordenadas de la imagen actual hasta las coordenadas de una imagen de referencia. La unidad de estimación de movimiento 122 puede facilitar el índice de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar los bloques de muestras predictivos de la PU en base a unas muestras reales o interpoladas en la ubicación de referencia indicada por el vector de movimiento de la PU.

[0167] Si una PU está en un sector B, la unidad de estimación de movimiento 122 puede realizar una unipredicción o una bipredicción para la PU. Para realizar una unipredicción para la PU, la unidad de estimación de movimiento 122 puede buscar en las imágenes de referencia de RefPicList0 o en una segunda lista de imágenes de referencia ("RefPicList1") una región de referencia para la PU. La unidad de estimación de movimiento 122 puede facilitar, como la información de movimiento de la PU, un índice de referencia que indica una posición en RefPicList0 o RefPicList1 de la imagen de referencia que contiene la región de referencia, un vector de movimiento que indica un desplazamiento espacial entre un bloque de muestras de la PU y una ubicación de referencia asociada con la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en RefPicList0 o en RefPicList1. La unidad de compensación de movimiento 124 puede generar los bloques de muestras predictivos de la PU en base, al menos en parte, a muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

[0168] Para realizar una interpredicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede buscar en las imágenes de referencia de RefPicList0 una región de referencia para la PU y también puede buscar en las imágenes de referencia de RefPicList1 otra región de referencia para la PU. La unidad de estimación de movimiento 122 puede generar índices de imagen de referencia (es decir, índices de referencia) que indican posiciones en RefPicList0 y RefPicList1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento 122 puede generar vectores de movimiento que indican desplazamientos espaciales entre la ubicación de referencia asociada a las regiones de referencia y un bloque de muestras de la PU. La información de movimiento de la PU puede incluir los índices de referencia y los vectores de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar los bloques de muestras predictivos de la PU en base, al menos en parte, a muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

[0169] De acuerdo con una o más técnicas de esta divulgación, una o más unidades dentro de la unidad de procesamiento de predicción 100 (tales como la unidad de procesamiento de intrapredicción 126) pueden realizar una o más de las técnicas descritas en el presente documento como parte de un proceso de codificación de vídeo.

[0170] La unidad de procesamiento de intrapredicción 126 puede generar datos predictivos para una PU realizando una intrapredicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos para la PU y diversos elementos de sintaxis. La unidad de procesamiento de intrapredicción 126 puede realizar una intrapredicción en las PU de sectores I, sectores P y sectores B.

[0171] Para realizar una intrapredicción en una PU, la unidad de procesamiento de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos predictivos para la PU. Para usar algunos modos de intrapredicción para generar un conjunto de datos predictivos para la PU, la unidad de procesamiento de intrapredicción 126 puede ampliar muestras de bloques de muestras de PU vecinas a través de los bloques de muestras de la PU en una dirección asociada al modo de intrapredicción. Las PU vecinas pueden estar encima, encima y a la derecha, encima y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba abajo para las PU, CU y CTU. La unidad de procesamiento de intrapredicción 126 puede usar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccionales. En algunos ejemplos, el número de modos de intrapredicción puede depender del tamaño de la región asociada a la PU. Además, como se describe en el presente documento, la unidad de procesamiento de intrapredicción 126 puede implementar modos de codificación sin pérdida y las modificaciones descritas en el presente documento para mejorar dichos modos de codificación. De acuerdo con algunas técnicas de esta divulgación, la unidad de procesamiento de

intrapredicción 126 puede usar uno o más valores de muestra originales dentro de un bloque para realizar una intrapredicción DC de otros valores de muestra dentro del bloque.

[0172] La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para las PU de una CU de entre los datos predictivos generados por la unidad de procesamiento de interpretación 120 para las PU, o los datos predictivos generados por la unidad de procesamiento de intrapredicción 126 para las PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos predictivos para las PU de la CU en base a unas métricas de velocidad/distorsión de los conjuntos de datos predictivos. Los bloques de muestras predictivos de los datos predictivos seleccionados se pueden denominar en el presente documento bloques de muestras predictivos seleccionados.

[0173] La unidad de generación residual 102 puede generar, en base a unos bloques de codificación (por ejemplo, los bloques de codificación de luma, Cb y Cr) de una CU y los bloques predictivos seleccionados (por ejemplo, los bloques predictivos de luma, Cb y Cr) de las PU de la CU, bloques residuales (por ejemplo, bloques residuales de luma, Cb y Cr) para la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de modo que cada muestra de los bloques residuales tiene un valor igual a una diferencia entre una muestra de un bloque de codificación de la CU y una muestra correspondiente de un bloque predictivo seleccionado correspondiente de una PU de la CU.

[0174] La unidad de procesamiento de transformada 104 puede realizar una división de árbol cuaternario para dividir los bloques residuales asociados a una CU en bloques de transformada asociados a las TU de la CU. Por tanto, una TU puede estar asociada a un bloque de transformada de luma y a dos bloques de transformada de croma. Los tamaños y las posiciones de los bloques de transformada de luma y croma de las TU de una CU pueden o no estar basados en los tamaños y las posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.

[0175] La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas a un bloque de transformada asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar a un bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada se puede tratar como un bloque de coeficientes de transformada.

[0176] La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada de un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o a la totalidad de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de n bits se puede redondear a la baja hasta un coeficiente de transformada de m bits durante la cuantificación, donde n es mayor que m . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado a una TU de una CU en base a un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor de QP asociado a la CU. La cuantificación puede introducir pérdida de información; por tanto, los coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.

[0177] De acuerdo con algunos ejemplos de esta divulgación, se puede realizar lo siguiente para $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, donde M es una altura de un bloque y N es la anchura de un bloque. En dichos ejemplos, el bloque es un bloque residual que incluye valores residuales que indican diferencias entre valores de muestra de un bloque predictivo generado usando intrapredicción y valores de muestra originales. Además, en dichos ejemplos, el bloque es un bloque con omisión de transformada. La unidad de generación residual 102 puede determinar un valor residual modificado $\tilde{r}_{i,j}$ para un valor residual $r_{i,j}$. Si el bloque se codifica usando un modo de intrapredicción vertical, $\tilde{r}_{i,j}$ se define como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases}$$

En la ecuación anterior, $Q(r_{(i-1),j})$ denota un valor residual reconstruido para un valor residual $r_{i-1,j}$ una columna a la izquierda del valor residual $r_{i,j}$. Si el bloque se codifica usando un modo de intrapredicción horizontal, $\tilde{r}_{i,j}$ se define como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M - 1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M - 1), 1 \leq j \leq (N - 1) \end{cases}$$

En la ecuación anterior, $Q(r_{i,j-1})$ denota un valor residual reconstruido para un valor residual $r_{i,j-1}$ una fila encima del valor residual $r_{i,j}$. La unidad de cuantificación 106 puede cuantificar el valor residual modificado $\tilde{r}_{i,j}$ para generar un valor residual modificado cuantificado $Q(\tilde{r}_{i,j})$.

[0178] La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar una cuantificación inversa y transformadas inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede añadir el bloque residual reconstruido a las muestras correspondientes de uno o más bloques predictivos generados por la unidad de procesamiento de predicción 100 para generar un bloque de transformada reconstruido asociado a una TU. Reconstruyendo bloques de transformada para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

[0179] La unidad de filtro 114 puede realizar una o más operaciones de eliminación de bloques para reducir los artefactos de bloque en los bloques de codificación asociados a una CU. La memoria intermedia de imágenes descodificadas 116 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 114 realice la una o más operaciones de eliminación de bloques en los bloques de codificación reconstruidos. La unidad de procesamiento de interpredicción 120 puede usar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar una interpredicción en las PU de otras imágenes. Además, la unidad de procesamiento de intrapredicción 126 puede usar bloques de codificación reconstruidos de la memoria intermedia de imágenes descodificadas 116 para realizar una intrapredicción en otras PU de la misma imagen que la CU.

[0180] La unidad de codificación entrópica 118 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación entrópica 118 puede recibir bloques de coeficientes desde la unidad de cuantificación 106 y puede recibir elementos de sintaxis desde la unidad de procesamiento de predicción 100. La unidad de codificación entrópica 118 puede realizar una o más operaciones de codificación entrópica en los datos para generar datos codificados entrópicamente. Por ejemplo, la unidad de codificación entrópica 118 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación entrópica de división en intervalos de probabilidad (PIPE), una operación de codificación Golomb exponencial u otro tipo de operación de codificación entrópica en los datos. El codificador de vídeo 20 puede facilitar un flujo de bits que incluye datos codificados entrópicamente generados por la unidad de codificación entrópica 118. Por ejemplo, el flujo de bits puede incluir datos que representan un RQT para una CU.

[0181] De acuerdo con algunos ejemplos de esta divulgación, la unidad de codificación entrópica 118 puede determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una DPCM residual. Cuando la ocultación de datos de signo está inhabilitada para el bloque actual, la unidad de codificación entrópica 118 puede incluir, en el flujo de bits, un elemento de sintaxis que indica si un valor del bloque actual es positivo o negativo.

[0182] El elemento 130 en la FIG. 6 puede representar un conmutador (o un conmutador conceptual) para seleccionar entre codificación sin pérdida y codificación con pérdida. La señal de control 132 puede representar una señal de la unidad de procesamiento de predicción 100 que determina la codificación sin pérdida o con pérdida, y el elemento 134 puede representar un bucle de descodificación que excluye la transformada inversa y los procesos de cuantificación inversa. En algunos ejemplos, la codificación sin pérdida elimina las transformadas y la cuantificación. En otros ejemplos, la codificación sin pérdida realiza transformadas y elimina solo el proceso de cuantificación. En otros ejemplos más, la codificación sin pérdida se puede implementar con el uso de transformadas y cuantificación, pero el parámetro de cuantificación se puede seleccionar para evitar cualquier pérdida de datos de cuantificación. Estos y otros ejemplos están dentro del alcance de esta divulgación.

[0183] Los elementos 136 y 138 representan conmutadores (o conmutadores conceptuales) que se pueden usar para implementar un modo de omisión de transformada. En los modos de omisión de transformada, los datos residuales no se transforman mediante la unidad de procesamiento de transformada 104, sino que se cuantifican mediante la unidad de cuantificación 106. Por tanto, las líneas discontinuas del elemento 136 representan dos trayectos de datos posibles. En un trayecto de datos, los datos residuales se cuantifican mediante la unidad de cuantificación 106 y, en el otro trayecto de datos, los datos residuales no se cuantifican mediante la unidad de cuantificación 106. De forma similar, en el bucle de descodificación del codificador de vídeo 20, los datos residuales se cuantifican inversamente mediante la unidad de cuantificación inversa 108, pero no se transforman mediante la unidad de procesamiento de transformada inversa 110. Por tanto, las líneas discontinuas del elemento 138 representan un trayecto de datos alternativo donde los datos residuales se cuantifican inversamente mediante la unidad de cuantificación inversa 108, pero no se transforman mediante la unidad de procesamiento de transformada inversa 110.

[0184] La FIG. 7 es un diagrama de bloques que ilustra un descodificador de vídeo 30 de ejemplo que está configurado para implementar las técnicas de esta divulgación. La FIG. 7 se proporciona con propósitos explicativos

y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

[0185] En el ejemplo de la FIG. 7, el descodificador de vídeo 30 incluye una unidad de descodificación entrópica 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de procesamiento de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y una memoria intermedia de imágenes descodificadas 162. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 164 y una unidad de procesamiento de intrapredicción 166. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

[0186] El descodificador de vídeo 30 puede recibir un flujo de bits. La unidad de descodificación entrópica 150 puede analizar el flujo de bits para descodificar los elementos de sintaxis del flujo de bits. La unidad de descodificación entrópica 150 puede descodificar entrópicamente los elementos de sintaxis codificados entrópicamente del flujo de bits. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de procesamiento de transformada inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 160 pueden generar datos de vídeo descodificados en base a los elementos de sintaxis extraídos del flujo de bits.

[0187] El flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de sector codificado. Como parte de la descodificación del flujo de bits, la unidad de descodificación entrópica 150 puede extraer y descodificar entrópicamente elementos de sintaxis de las unidades NAL de sector codificado. Cada uno de los sectores codificados puede incluir una cabecera de sector y unos datos de sector. La cabecera de sector puede contener elementos de sintaxis pertenecientes a un sector. Los elementos de sintaxis de la cabecera de sector pueden incluir un elemento de sintaxis que identifica un PPS asociado a una imagen que contiene el sector.

[0188] De acuerdo con algunos ejemplos de esta divulgación, la unidad de descodificación entrópica 150 determina que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una DPCM residual. En dichos ejemplos, cuando la ocultación de datos de signo está inhabilitada para el bloque actual, la unidad de descodificación entrópica 150 obtiene, del flujo de bits, para cada valor significativo respectivo del bloque, un elemento de sintaxis respectivo que indica si el valor significativo respectivo es positivo o negativo.

[0189] Además de descodificar (es decir, obtener) elementos de sintaxis del flujo de bits, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir bloques residuales de la CU.

[0190] Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede realizar la cuantificación inversa, es decir, la descuantificación, de los bloques de coeficientes asociados a la TU. La unidad de cuantificación inversa 154 puede usar un valor de QP asociado a la CU de la TU para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para que la unidad de cuantificación inversa 154 lo aplique. Es decir, la relación de compresión, es decir, la relación del número de bits usados para representar la secuencia original y la comprimida, se puede controlar ajustando el valor del QP usado al cuantificar los coeficientes de transformada. La relación de compresión también puede depender del procedimiento de codificación entrópica empleado.

[0191] Después de que la unidad de cuantificación inversa 154 haya realizado la cuantificación inversa de un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes para generar un bloque residual asociado a la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada entera inversa, una transformada inversa de Karhunen-Loeve (KLT), una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.

[0192] Si se codifica una PU usando intrapredicción, la unidad de procesamiento de intrapredicción 166 puede realizar una intrapredicción para generar bloques predictivos para la PU. La unidad de procesamiento de intrapredicción 166 puede usar un modo de intrapredicción para generar los bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para la PU en base a los bloques de predicción de unas PU espacialmente vecinas. La unidad de procesamiento de intrapredicción 166 puede determinar el modo de intrapredicción para la PU en base a uno o más elementos de sintaxis descodificados a partir del flujo de bits. De acuerdo con algunas técnicas de esta divulgación, la unidad de procesamiento de intrapredicción 166 puede usar uno o más valores de muestra originales dentro de un bloque para realizar una intrapredicción DC de otros valores de muestra dentro del bloque. Es decir, la unidad de procesamiento de intrapredicción 166 puede generar un bloque predictivo. Como parte de la generación del bloque predictivo, la unidad de procesamiento de intrapredicción 166 puede usar al menos una de una muestra

reconstruida sin pérdida a la izquierda de una muestra actual en una fila actual de un bloque predictivo y una muestra reconstruida sin pérdida para una fila del bloque predictivo de encima de la fila actual para predicción DC de la muestra actual

[0193] La unidad de procesamiento de predicción 152 puede construir una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) en base a unos elementos de sintaxis extraídos del flujo de bits. Además, si una PU se codifica usando interpredicción, la unidad de descodificación entrópica 150 puede extraer información de movimiento para la PU. La unidad de compensación de movimiento 164 puede determinar, en base a la información de movimiento de la PU, una o más regiones de referencia para la PU. La unidad de compensación de movimiento 164 puede generar, en base a unos bloques de muestras en el uno o más bloques de referencia para la PU, unos bloques predictivos (por ejemplo, bloques de luma, Cb y Cr) para la PU. De acuerdo con una o más técnicas de esta divulgación, una o más unidades dentro de la unidad de procesamiento de predicción 152 (tales como la unidad de procesamiento de intrapredicción 166) pueden realizar unas técnicas descritas en el presente documento como parte de un proceso de descodificación de vídeo.

[0194] La unidad de reconstrucción 158 puede usar el bloque de transformada (por ejemplo, los bloques de transformada de luma, Cb y Cr) asociado con las TU de una CU y los bloques predictivos (por ejemplo, los bloques de luma, Cb y Cr) de las PU de la CU, es decir, ya sean datos de intrapredicción o bien datos de interpredicción, según corresponda, para reconstruir los bloques de codificación (por ejemplo, los bloques de codificación de luma, Cb y Cr) de la CU. Por ejemplo, la unidad de reconstrucción 158 puede añadir muestras de los bloques de transformada de luma, Cb y Cr a muestras correspondientes de los bloques predictivos de luma, Cb y Cr para reconstruir los bloques de codificación de luma, Cb y Cr de la CU.

[0195] De acuerdo con algunos ejemplos de esta divulgación, la unidad de descodificación entrópica 150 puede generar un bloque de valores residuales. Este bloque puede ser un bloque de omisión de transformada. Además, el bloque puede ser un bloque residual que incluye valores residuales que indican diferencias entre los valores de muestra originales y los valores de muestra de un bloque predictivo generado usando intrapredicción. Además, para $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, donde M es una altura del bloque y N es la anchura del bloque, la unidad de cuantificación inversa 154 puede calcular un valor residual reconstruido $Q(r_{i,j})$ para un valor residual $r_{i,j}$ del bloque. Si el bloque se codifica usando un modo de intrapredicción vertical (o, en algunos ejemplos, un modo de intrapredicción casi vertical), $Q(r_{i,j})$ se define como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N - 1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases}$$

En la ecuación anterior, $Q(\tilde{r}_{i,j})$ denota una versión cuantificada de un valor residual modificado $\tilde{r}_{i,j}$, el valor residual modificado $\tilde{r}_{i,j}$ es una versión modificada del valor residual $r_{i,j}$ y $Q(r_{i-1,j})$ es un valor residual reconstruido para (es decir, correspondiente a) un valor residual una columna a la izquierda del valor residual $r_{i,j}$. La unidad de descodificación entrópica 150 puede haber determinado previamente $Q(r_{i-1,j})$ de la misma manera que la unidad de descodificación entrópica 150 determina $Q(r_{i,j})$. Si el bloque se codifica usando un modo de intrapredicción horizontal, $Q(r_{i,j})$ se define como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M - 1), j = 0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M - 1), 1 \leq j \leq (N - 1) \end{cases}$$

En la ecuación anterior, $Q(r_{i,j-1})$ es un valor residual reconstruido para un valor residual una fila encima del valor residual $r_{i,j}$. La unidad de descodificación entrópica 150 puede haber determinado previamente $Q(r_{i,j-1})$ de la misma manera que la unidad de descodificación entrópica 150 determina $Q(r_{i,j})$. La unidad de reconstrucción 158 puede añadir el valor residual reconstruido $Q(r_{i,j})$ a un valor de predicción para reconstruir un valor de muestra.

[0196] La unidad de filtro 160 puede realizar una operación de eliminación de bloques para reducir los artefactos de bloque asociados con los bloques de codificación (por ejemplo, los bloques de codificación de luma, Cb y Cr) de la CU. El descodificador de vídeo 30 puede almacenar los bloques de codificación (por ejemplo, los bloques de codificación de luma, Cb y Cr) de la CU en la memoria intermedia de imágenes descodificadas 162. La memoria intermedia de imágenes descodificadas 162 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, en base a los bloques (por ejemplo, los bloques de luma, Cb y Cr) de la memoria intermedia de imágenes descodificadas 162, operaciones de intrapredicción o de interpredicción en las PU de otras CU. De esta manera, el descodificador de vídeo 30 puede extraer, del flujo de bits, niveles de coeficientes de transformada del bloque de coeficientes de luma significativos, realizar la cuantificación inversa de los niveles de coeficientes de transformada, aplicar una transformada a los niveles de coeficientes de transformada para generar un bloque de transformada, generar, en base al menos en parte al bloque de transformada, un bloque de codificación, y facilitar el bloque de codificación para su visualización.

[0197] El elemento 170 puede representar un trayecto de codificación normal para compresión con pérdida, y el elemento 172 puede representar un trayecto de codificación de exclusión que excluye la transformada inversa y los procesos de cuantificación inversa. Estos diferentes trayectos son meramente ejemplares y la codificación sin pérdida se puede realizar sin ninguna exclusión. En algunos ejemplos, la codificación sin pérdida elimina las transformadas y la cuantificación. En otros ejemplos, la codificación sin pérdida realiza transformadas y elimina solo el proceso de cuantificación. En otros ejemplos más, la codificación sin pérdida se puede implementar con el uso de transformadas y cuantificación, pero el parámetro de cuantificación se puede seleccionar para evitar cualquier pérdida de datos de cuantificación. El elemento 174 representa un ejemplo de trayecto que se puede usar para un modo de omisión de transformada. En un modo de omisión de transformada, los datos residuales se pueden cuantificar inversamente mediante la unidad de cuantificación inversa 154, pero la transformada inversa de la unidad de procesamiento de transformada inversa 156 se puede omitir. Estos y otros ejemplos están dentro del alcance de esta divulgación.

[0198] La FIG. 8A es un diagrama de flujo que ilustra un ejemplo de operación del codificador de vídeo 20 de acuerdo con una o más técnicas de esta divulgación. En otros ejemplos de esta divulgación, operaciones similares a la operación de la FIG. 8A pueden incluir más, menos o diferentes acciones. Además, en algunos ejemplos, se pueden omitir o reorganizar una o más acciones de la operación de la FIG. 8A. Por ejemplo, en la FIG. 8A y la FIG. 8B, las líneas discontinuas indican acciones no realizadas en algunos ejemplos.

[0199] En el ejemplo de la FIG. 8A, el codificador de vídeo 20 puede generar un bloque predictivo (200). Como parte de la generación del bloque predictivo, el codificador de vídeo 20 puede usar al menos uno de: una muestra reconstruida sin pérdida a la izquierda de una muestra actual en una fila actual de un bloque predictivo y una muestra reconstruida sin pérdida para una fila del bloque predictivo encima de la fila actual para predicción DC de la muestra actual (202). En algunos ejemplos, el decodificador de vídeo 30 procesa muestras del bloque predictivo en un orden de barrido horizontal, un orden de barrido vertical, un orden de barrido diagonal o un orden de barrido en zig-zag. En dichos ejemplos, procesar las muestras del bloque predictivo puede comprender determinar predicciones DC para las muestras, así como reconstruir las muestras sin pérdida. Además, en el ejemplo de la FIG. 8A, el codificador de vídeo 20 puede generar muestras residuales que tienen valores iguales a una diferencia entre una muestra de un bloque de codificación y una muestra correspondiente del bloque predictivo.

[0200] La FIG. 8B es un diagrama de flujo que ilustra un ejemplo de operación del decodificador de vídeo 30, de acuerdo con una o más técnicas de esta divulgación. En otros ejemplos de esta divulgación, unas operaciones similares a la operación de la FIG. 8B pueden incluir más, menos o diferentes acciones. Además, en algunos ejemplos, se puede omitir o reorganizar una o más acciones de la operación de la FIG. 8B.

[0201] En el ejemplo de la FIG. 8B, el decodificador de vídeo 30 puede generar un bloque predictivo (250). Como parte de la generación del bloque predictivo, el decodificador de vídeo 30 puede usar al menos una de una muestra reconstruida sin pérdida a la izquierda de una muestra actual en una fila actual de un bloque predictivo y una muestra reconstruida sin pérdida para una fila del bloque predictivo encima de la fila actual para una predicción DC de la muestra actual (252). En algunos ejemplos, el decodificador de vídeo 30 procesa muestras del bloque predictivo en un orden de barrido horizontal, un orden de barrido vertical, un orden de barrido diagonal o un orden de barrido en zig-zag. En dichos ejemplos, el procesamiento de las muestras del bloque predictivo puede comprender determinar predicciones DC para las muestras. Además, en el ejemplo de la FIG. 8B, el decodificador de vídeo 30 puede reconstruir un bloque de codificación añadiendo muestras del bloque predictivo a las muestras residuales correspondientes (254).

[0202] La FIG. 9A es un diagrama de flujo que ilustra un ejemplo de operación del codificador de vídeo 20 de acuerdo con una o más técnicas de esta divulgación. En otros ejemplos de esta divulgación, operaciones similares a la operación de la FIG. 9A pueden incluir más, menos o diferentes acciones. Además, en algunos ejemplos, se pueden omitir o reorganizar una o más acciones de la operación de la FIG. 9A. El ejemplo de la FIG. 9A se explica con referencia a los componentes mostrados en la FIG. 6. Sin embargo, la operación de la FIG. 9A se puede realizar mediante componentes y tipos de codificadores de vídeo distintos de los mostrados en la FIG. 6.

[0203] Como se indica en el ejemplo de la FIG. 9A, la unidad de generación residual 102 del codificador de vídeo 20 puede generar un bloque de valores residuales (350). En este ejemplo, el bloque de valores residuales es un bloque de omisión de transformada. El bloque puede ser un bloque residual que incluye valores residuales que indican diferencias entre valores de muestra originales y valores de muestra de un bloque predictivo generado usando intrapredicción. El codificador de vídeo puede realizar las acciones restantes de la FIG. 9A para cada ubicación (i,j) del bloque, donde $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, M es la altura de un bloque y N es la anchura del bloque.

[0204] Además, la unidad de generación residual 102 del codificador de vídeo 20 puede determinar un valor residual modificado $\tilde{r}_{i,j}$ para un valor residual $r_{i,j}$ en el bloque (352). Si el bloque se codifica usando un modo de intrapredicción vertical, o en algunos ejemplos, un modo de intrapredicción entre 22 y 30 como se define en HEVC Working Draft 10, $\tilde{r}_{i,j}$ se define como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

$Q(r_{(i-1),j})$ denota un valor residual reconstruido para (es decir, correspondiente a) un valor residual $r_{i-1,j}$. Si el bloque se codifica usando un modo de intrapredicción horizontal o, en algunos ejemplos, un modo de intrapredicción entre 6 y 14, $\tilde{r}_{i,j}$ se define como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

$Q(r_{i,(j-1)})$ denota un valor residual reconstruido para un valor residual $r_{i,j-1}$.

[0205] Además, la unidad de cuantificación 106 del codificador de vídeo 20 puede cuantificar el valor residual modificado $\tilde{r}_{i,j}$ para generar un valor residual modificado cuantificado $Q(\tilde{r}_{i,j})$ (354). El codificador de vídeo 20 puede señalar el valor residual modificado cuantificado $Q(\tilde{r}_{i,j})$ en un flujo de bits (356). Por ejemplo, el codificador de vídeo 20 puede generar uno o más elementos de sintaxis que indican el valor residual modificado cuantificado $Q(\tilde{r}_{i,j})$. En este ejemplo, la unidad de codificación entrópica 118 del codificador de vídeo 20 puede codificar entrópicamente el uno o más elementos de sintaxis e incluir los datos resultantes en el flujo de bits.

[0206] Además, en el ejemplo de la FIG. 9A, el codificador de vídeo 20 puede calcular un valor residual reconstruido $Q(r_{i,j})$ (358). En algunos ejemplos, el codificador de vídeo 20 puede calcular el valor residual reconstruido $Q(r_{i,j})$ como parte de un bucle de retroalimentación para determinar valores de muestra reconstruidos para su uso en una intrapredicción o interpredicción adicionales. Si el bloque se codifica usando el modo de intrapredicción vertical (o en algunos ejemplos, un modo de intrapredicción casi vertical), el valor residual reconstruido $Q(r_{i,j})$ se puede definir como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

Si el bloque se codifica usando el modo de intrapredicción horizontal (o en algunos ejemplos, un modo de intrapredicción horizontal), $Q(r_{i,j})$ se puede definir como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j = 0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

[0207] Debido a que el bloque es un bloque de omisión de transformada, la unidad de procesamiento de transformada inversa 110 no aplica una transformada inversa al valor residual $r_{i,j}$. En consecuencia, la unidad de reconstrucción 112 del codificador de vídeo 20 puede añadir el valor residual reconstruido $Q(r_{i,j})$ a un valor de predicción para determinar un valor de muestra reconstruido (360). El valor de predicción puede ser una muestra de un bloque predictivo. La unidad de procesamiento de predicción 100 del codificador de vídeo 20 puede usar el valor de muestra reconstruido para la intrapredicción o interpredicción de otros bloques (362).

[0208] La FIG. 9B es un diagrama de flujo que ilustra una operación de ejemplo del decodificador de vídeo 30, de acuerdo con una o más técnicas de esta divulgación. En otros ejemplos de esta divulgación, unas operaciones similares a la operación de la FIG. 9B pueden incluir más, menos o diferentes acciones. Además, en algunos ejemplos, se pueden omitir o reorganizar una o más acciones de la operación de la FIG. 9B. El ejemplo de la FIG. 9B se explica con referencia a los componentes mostrados en la FIG. 7. Sin embargo, la operación de la FIG. 9B se puede realizar mediante componentes y tipos de decodificadores de vídeo distintos de los mostrados en la FIG. 7.

[0209] El decodificador de vídeo 30 puede realizar la operación de la FIG. 9B para cada ubicación (i,j) de un bloque de omisión de transformada, donde $0 \leq i \leq (M-1)$ y $0 \leq j \leq (N-1)$, M es una altura de un bloque y N es la anchura del bloque. El bloque puede ser un bloque residual que incluye valores residuales que indican diferencias entre valores de muestra originales y valores de muestra de un bloque predictivo generado usando intrapredicción. Como se indica en el ejemplo de la FIG. 9A, la unidad de decodificación entrópica 150 del decodificador de vídeo 30 puede obtener, de un flujo de bits, uno o más elementos de sintaxis que indican un valor residual cuantificado modificado $Q(\tilde{r}_{i,j})$ (400). La unidad de decodificación entrópica 150 puede decodificar entrópicamente algunos o la totalidad del uno o más elementos de sintaxis.

[0210] Además, en el ejemplo de la FIG. 9B, la unidad de cuantificación inversa 154 del decodificador de vídeo 30 puede calcular un valor residual reconstruido $Q(r_{i,j})$ para un valor residual $r_{i,j}$ (402). En algunos ejemplos, el valor

residual $r_{i,j}$ es un valor residual con desplazamiento de bits como se describe en otra parte de esta divulgación. Si el bloque se codifica usando un modo de intrapredicción vertical, $Q(r_{i,j})$ se define como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}.$$

$Q(\tilde{r}_{i,j})$ denota una versión cuantificada de un valor residual modificado $\tilde{r}_{i,j}$, el valor residual modificado $\tilde{r}_{i,j}$ es una versión modificada del valor residual $r_{i,j}$ y $Q(r_{i-1,j})$ es un valor residual reconstruido para un valor residual una columna a la izquierda del valor residual $r_{i,j}$. Si el bloque se codifica usando un modo de intrapredicción horizontal, $Q(r_{i,j})$ se define como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j = 0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}.$$

$Q(r_{i,j-1})$ es un valor residual reconstruido para un valor residual una fila encima del valor residual $r_{i,j}$.

[0211] Si el bloque se codifica usando el modo de intrapredicción vertical, el valor residual modificado $\tilde{r}_{i,j}$ se define como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{i-1,j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}.$$

Si el bloque se codifica usando el modo de intrapredicción horizontal, el valor residual modificado $\tilde{r}_{i,j}$ se define como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}.$$

[0212] Debido a que el bloque es un bloque de omisión de transformada, la unidad de procesamiento de transformada inversa 156 no aplica una transformada inversa al valor residual reconstruido $Q(r_{i,j})$. La unidad de reconstrucción 158 del decodificador de vídeo 30 puede añadir el valor residual reconstruido $Q(r_{i,j})$ a un valor de predicción para reconstruir un valor de muestra (406). El valor de predicción puede ser una muestra de un bloque predictivo.

[0213] La FIG. 10A es un diagrama de flujo que ilustra una operación de codificador de vídeo de ejemplo, de acuerdo con una o más técnicas de esta divulgación. En otros ejemplos de esta divulgación, operaciones similares a la operación de la FIG. 10A pueden incluir más, menos o diferentes acciones. Además, en algunos ejemplos, se pueden omitir o reorganizar una o más acciones de la operación de la FIG. 10A.

[0214] En el ejemplo de la FIG. 10A, el codificador de vídeo 20 genera un flujo de bits que incluye una secuencia de bits que forma una representación codificada de datos de vídeo (600). Como parte de la generación del flujo de bits, el codificador de vídeo 20 puede determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera sin la aplicación de una transformada a los datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de DPCM residual (602). En el contexto de la FIG. 10A, el bloque actual puede ser un subbloque 4x4 de un bloque de muestras residuales a las que se ha aplicado la técnica de DPCM residual. Posteriormente, en el ejemplo de la FIG. 10A, el codificador de vídeo 20 puede facilitar el flujo de bits (604).

[0215] En algunos ejemplos, cuando la ocultación de datos de signo está inhabilitada para el bloque actual, el codificador de vídeo 20 puede señalar, en el flujo de bits, para cada valor residual significativo respectivo del bloque actual, un elemento de sintaxis que indica si el valor residual significativo respectivo es positivo o negativo. En dichos ejemplos, cuando la ocultación de datos de signo no está inhabilitada para el bloque actual, el codificador de vídeo 20 puede no señalar, en el flujo de bits, un elemento de sintaxis que indica si un valor de al menos un valor residual significativo o un coeficiente de transformada del bloque actual son positivos o negativos.

[0216] La FIG. 10B es un diagrama de flujo que ilustra una operación de decodificador de vídeo de ejemplo para ocultación de datos de signo, de acuerdo con una o más técnicas de esta divulgación. En otros ejemplos de esta divulgación, unas operaciones similares a la operación de la FIG. 10B pueden incluir más, menos o diferentes

acciones. Además, en algunos ejemplos, se pueden omitir o reorganizar una o más acciones de la operación de la FIG. 10B.

[0217] En el ejemplo de la FIG. 10B, el descodificador de vídeo 30 obtiene elementos de sintaxis de un flujo de bits que incluye una secuencia de bits que forma una representación codificada de datos de vídeo (650). Como parte de la obtención de los elementos de sintaxis del flujo de bits, el descodificador de vídeo 30 puede determinar que la ocultación de datos de signo está inhabilitada para un bloque actual si el bloque actual se genera sin la aplicación de una transformada a unos datos residuales y el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de DPCM residual (652). En el contexto de la FIG. 10B, el bloque actual puede ser un subbloque 4x4 de un bloque de muestras residuales a las que se ha aplicado la técnica de DPCM residual. Posteriormente, en el ejemplo de la FIG. 10B, el descodificador de vídeo 30 puede reconstruir una imagen de los datos de vídeo en base, al menos en parte, a los elementos de sintaxis obtenidos del flujo de bits (654).

[0218] En algunos ejemplos, cuando la ocultación de datos de signo está inhabilitada para el bloque actual, el descodificador de vídeo 30 puede obtener, del flujo de bits, para cada valor residual significativo respectivo del bloque actual, un elemento de sintaxis que indica si el valor residual significativo respectivo es positivo o negativo. En dichos ejemplos, cuando la ocultación de datos de signo no está inhabilitada para el bloque actual, el descodificador de vídeo 30 puede no obtener del flujo de bits un elemento de sintaxis que indica si un valor de un valor residual significativo del bloque actual es positivo o negativo.

[0219] En uno o más ejemplos, las funciones descritas en el presente documento se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir a través de, un medio legible por ordenador como una o más instrucciones o código, y ejecutar 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 facilita la transferencia de un programa informático de 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) medios de almacenamiento tangibles legibles por ordenador que son 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 puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0220] A modo de ejemplo, y no de limitación, 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 se puede usar para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se puede acceder mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas 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 están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se dirigen a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, de los cuales los discos flexibles normalmente reproducen datos magnéticamente, mientras que los demás discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores deberían estar incluidos también dentro del alcance de los medios legibles por ordenador.

[0221] 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 pueden ejecutar las instrucciones. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir 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 se puede proporcionar dentro de módulos de hardware y/o de software dedicados configurados para codificar y descodificar, o incorporar en un códec combinado. Asimismo, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

[0222] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden

combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluye uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

- 5 **[0223]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de descodificación de datos de vídeo, comprendiendo el procedimiento:

5 obtener elementos de sintaxis de un flujo de bits que incluye una secuencia de bits que forman una representación codificada de los datos de vídeo,

10 en el que obtener los elementos de sintaxis del flujo de bits comprende determinar que una ocultación de datos de signo está inhabilitada para un bloque actual en base a (i) que el bloque actual se genera usando codificación con pérdida sin aplicación de una transformada a unos datos residuales y (ii) que el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación de codificación de impulsos diferencial, DPCM, residual; y

15 reconstruir una imagen de los datos de vídeo en base al menos en parte a los elementos de sintaxis obtenidos del flujo de bits.

2. El procedimiento de la reivindicación 1, en el que obtener los elementos de sintaxis del flujo de datos comprende además:

20 cuando una ocultación de datos de signo está inhabilitada para el bloque actual, obtener, del flujo de bits, para cada valor residual significativo respectivo del bloque actual, un elemento de sintaxis que indica si el valor residual significativo respectivo es positivo o negativo, y

25 cuando una ocultación de datos de signo no está inhabilitada para el bloque actual, no obtener del flujo de bits un elemento de sintaxis que indica si un valor de un valor residual significativo del bloque actual es positivo o negativo.

3. El procedimiento de la reivindicación 1, en el que obtener los elementos de sintaxis del flujo de bits comprende obtener, del flujo de bits, una indicación explícita de que una ocultación de datos de signo está habilitada para el bloque actual.

4. El procedimiento de la reivindicación 1, en el que determinar que una ocultación de datos de signo está inhabilitada para el bloque actual comprende determinar que una ocultación de datos de signo está inhabilitada para el bloque actual en base a:

35 que el bloque actual se codifica sin aplicación de la transformada a los datos residuales, y

que el bloque actual se intracodifica usando un modo de intrapredicción DC o un modo de intrapredicción plano.

40 5. El procedimiento de la reivindicación 1, en el que reconstruir la imagen comprende:

generar un bloque de valores residuales que incluye el bloque actual, en el que el bloque es un bloque de omisión de transformada; y

45 para $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, donde M es una altura del bloque y N es la anchura del bloque, el procedimiento comprende:

calcular un valor residual reconstruido $Q(r_{i,j})$ correspondiente a un valor residual $r_{i,j}$ del bloque,

50 en el que si el bloque se codifica usando un modo de intrapredicción vertical, $Q(r_{i,j})$ se define como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N - 1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases},$$

55 en el que $Q(\tilde{r}_{i,j})$ denota una versión cuantificada de un valor residual modificado $\tilde{r}_{i,j}$, el valor residual modificado $\tilde{r}_{i,j}$ es una versión modificada del valor residual $r_{i,j}$, y $Q(r_{i-1,j})$ es un valor residual reconstruido correspondiente a un valor residual una columna a la izquierda del valor residual $r_{i,j}$, y

en el que si el bloque se codifica usando un modo de intrapredicción horizontal, $Q(r_{i,j})$ se define como:

60
$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M - 1), j = 0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M - 1), 1 \leq j \leq (N - 1) \end{cases},$$

en el que $Q(r_{i,j-1})$ es un valor residual reconstruido correspondiente a un valor residual una fila encima del valor residual $r_{i,j}$; y añadir el valor residual reconstruido $Q(r_{i,j})$ a un valor de predicción para reconstruir un valor de muestra.

5 6. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:

generar un flujo de bits que incluye una secuencia de bits que forma una representación codificada de los datos de vídeo,

10 en el que generar el flujo de bits comprende determinar que una ocultación de datos de signo está inhabilitada para un bloque actual en base a (i) que el bloque actual se genera usando codificación con pérdida sin aplicación de una transformada a unos datos residuales y (ii) que el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación de codificación de impulsos diferencial, DPCM, residual; y

15 facilitar el flujo de bits.

7. El procedimiento de la reivindicación 6, en el que generar el flujo de bits comprende:

20 cuando una ocultación de datos de signo está inhabilitada para el bloque actual, señalar, en el flujo de bits, para cada valor residual significativo respectivo del bloque actual, un elemento de sintaxis que indica si el valor residual significativo respectivo es positivo o negativo, y

25 cuando una ocultación de datos de signo no está inhabilitada para el bloque actual, no señalar, en el flujo de bits, un elemento de sintaxis que indica si un valor de un valor residual significativo del bloque actual es positivo o negativo.

30 8. El procedimiento de la reivindicación 6, en el que generar el flujo de bits comprende señalar, en el flujo de bits, una indicación explícita de que una ocultación de datos de signo está habilitada para el bloque actual.

9. El procedimiento de la reivindicación 6, en el que determinar que una ocultación de datos de signo está inhabilitada para el bloque actual comprende determinar que una ocultación de datos de signo está inhabilitada para el bloque actual en base a:

35 que el bloque actual se codifica sin aplicación de la transformada a los datos residuales, y

que el bloque actual se intracodifica usando un modo de intrapredicción DC o un modo de intrapredicción plano.

40 10. El procedimiento de la reivindicación 1 o la reivindicación 6, en el que el modo de intrapredicción en el que se aplica una DPCM residual es un modo de intrapredicción horizontal o vertical.

11. El procedimiento de la reivindicación 6, que comprende además:

45 para $0 \leq i \leq (M - 1)$ y $0 \leq j \leq (N - 1)$, donde M es una altura de un bloque que incluye el bloque actual y N es la anchura del bloque, en el que el bloque es un bloque de omisión de transformada;

determinar un valor residual modificado $\tilde{r}_{i,j}$ para un valor residual $r_{i,j}$,

50 en el que si el bloque se codifica usando un modo de intrapredicción vertical, $\tilde{r}_{i,j}$ se define como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases},$$

55 en el que $Q(r_{(i-1),j})$ denota un valor residual reconstruido correspondiente a un valor residual $r_{i-1,j}$ una columna a la izquierda del valor residual $r_{i,j}$, y

si el bloque se codifica usando un modo de intrapredicción horizontal, $\tilde{r}_{i,j}$ se define como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M - 1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M - 1), 1 \leq j \leq (N - 1) \end{cases},$$

60 en el que $Q(r_{i,(j-1)})$ denota un valor residual reconstruido correspondiente a un valor residual $r_{i,j-1}$ una fila encima del valor residual $r_{i,j}$; y

cuantificar el valor residual modificado $\tilde{r}_{i,j}$ para generar un valor residual modificado cuantificado $Q(\tilde{r}_{i,j})$.

12. Un aparato de decodificación de vídeo que comprende:

- 5 medios para obtener elementos de sintaxis de un flujo de bits que incluye una secuencia de bits que forma una representación codificada de datos de vídeo,
- 10 en el que obtener los elementos de sintaxis del flujo de bits comprende determinar que una ocultación de datos de signo está inhabilitada para un bloque actual en base a (i) que el bloque actual se genera usando codificación con pérdida sin aplicación de una transformada a unos datos residuales y (ii) que el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación de codificación de impulsos diferencial, DPCM, residual; y
- 15 medios para reconstruir una imagen de los datos de vídeo en base al menos en parte a los elementos de sintaxis obtenidos del flujo de bits.

13. Un aparato de codificación de vídeo, que comprende:

- 20 medios para generar un flujo de bits que incluye una secuencia de bits que forma una representación codificada de datos de vídeo,
- 25 en el que generar el flujo de bits comprende determinar que una ocultación de datos de signo está inhabilitada para un bloque actual en base a (i) que el bloque actual se genera usando codificación con pérdida sin aplicación de una transformada a unos datos residuales y (ii) que el bloque actual se intrapredice usando un modo de intrapredicción en el que se usa una técnica de modulación de codificación de impulsos diferencial, DPCM, residual; y
- 30 medios para facilitar el flujo de bits.

14. Un medio de almacenamiento de datos legible por ordenador no transitorio que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que uno o más procesadores realicen el procedimiento de cualquiera de las reivindicaciones 1-11.

15. El aparato de codificación de vídeo de la reivindicación 12 o la reivindicación 13, que comprende además al menos uno de:

- un circuito integrado;
- 40 un microprocesador;
- un dispositivo de comunicación inalámbrica que incluye el uno o más procesadores; una pantalla configurada para visualizar una imagen descodificada que contiene el bloque actual; y una cámara configurada para capturar una imagen que contiene el bloque actual.

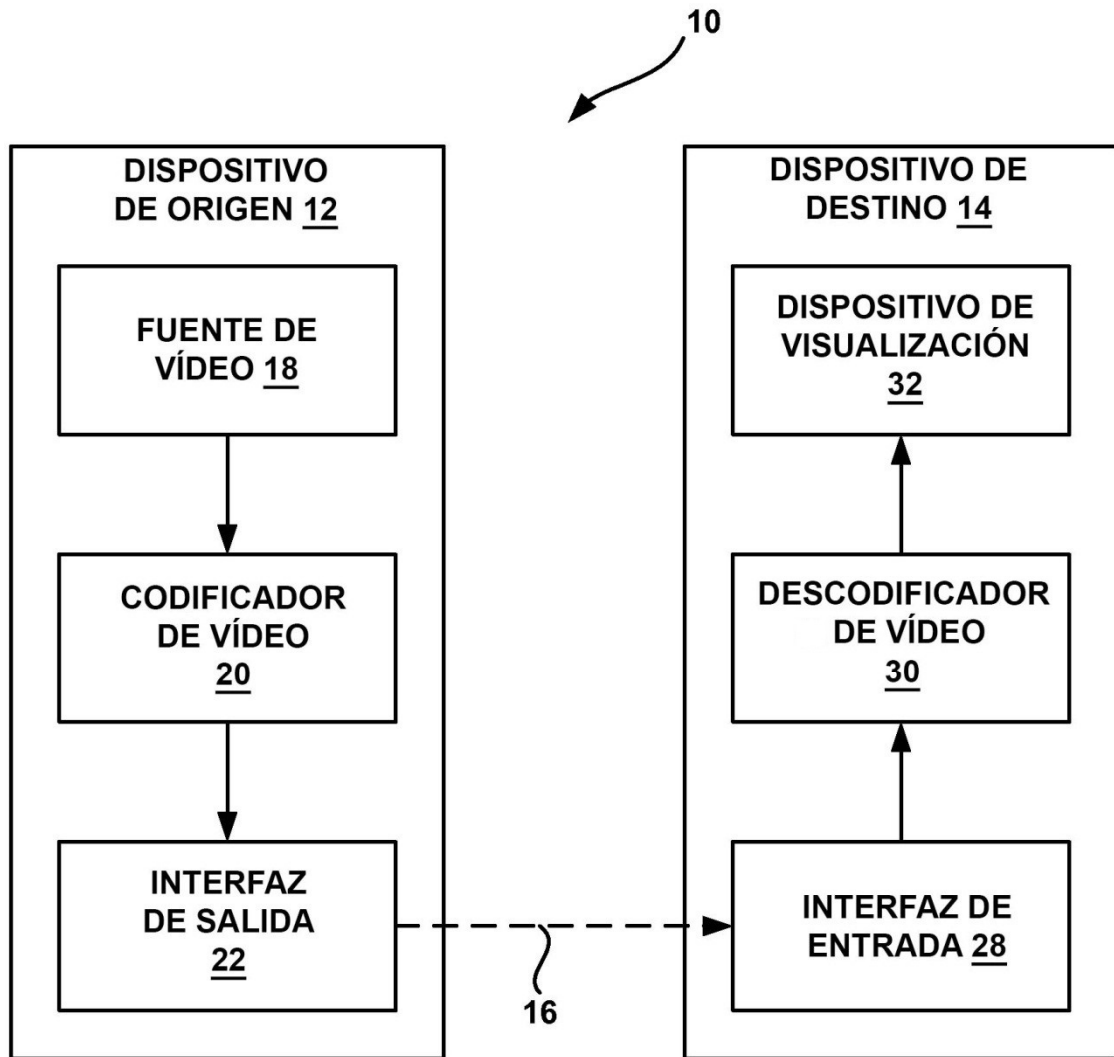


FIG. 1

$P_{-1,-1}$	$P_{-1,0}$	$P_{-1,1}$	$\dots$	$P_{-1,N-1}$	$P_{-1,N}$	$\dots$		$P_{-1,2N-1}$
$P_{0,-1}$	$P_{0,0}$	$P_{0,1}$	$\dots$	$P_{0,N-1}$				
$P_{1,-1}$	$P_{1,0}$			$\vdots$				
$\vdots$	$\vdots$							
$P_{M-1,-1}$	$P_{M-1,0}$	$\dots$		$P_{M-1,N-1}$				
$P_{M,-1}$								
$\vdots$								
$P_{2M-1,-1}$								

$M = N = 4$ en esta figura

FIG. 2

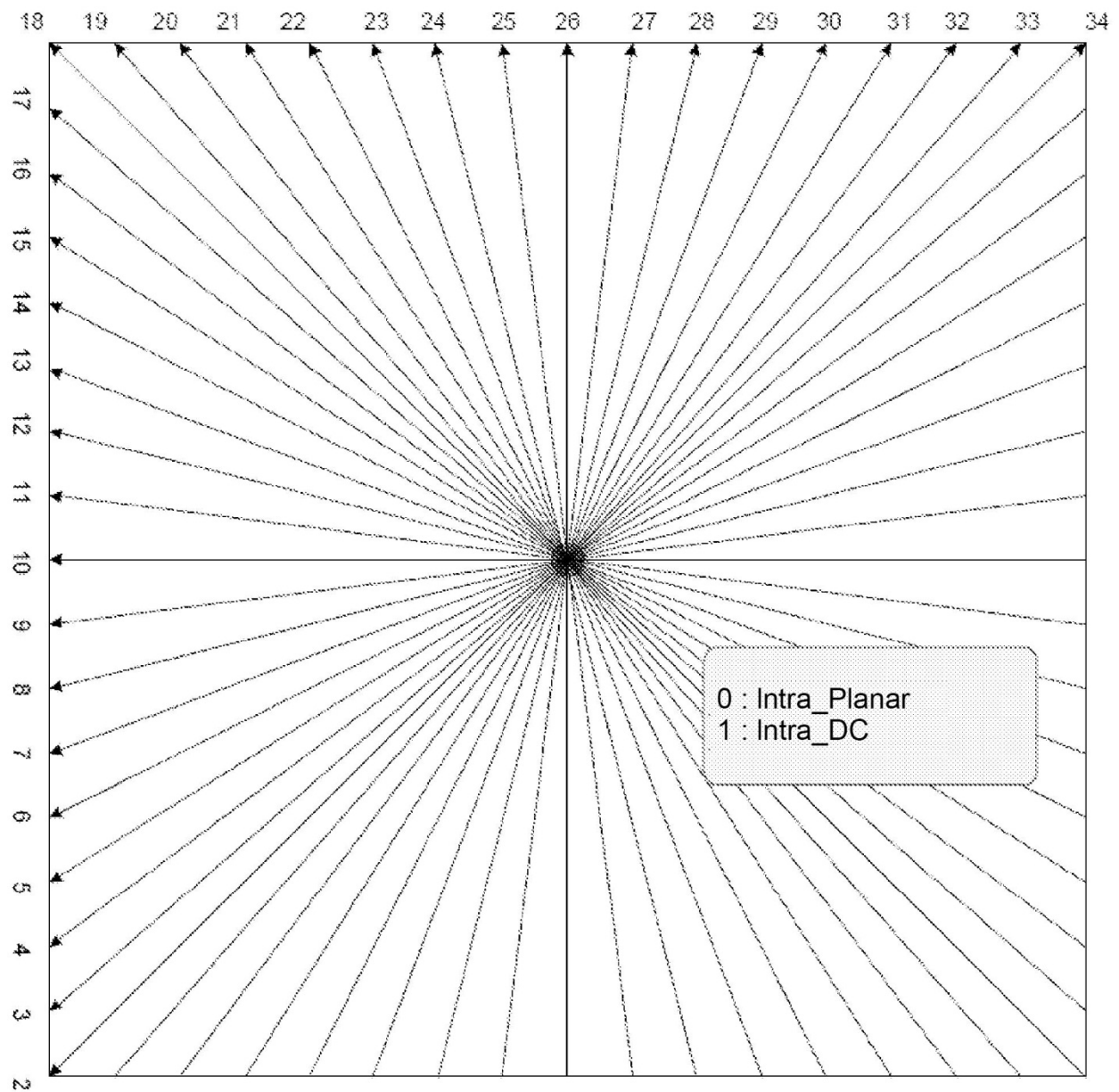


FIG. 3

$P_{-1,-1}$	$P_{-1,0}$	$P_{-1,1}$	■ ■ ■	$P_{-1,N-1}$
$P_{0,-1}$	$P_{0,0}$	$P_{0,1}$	■ ■ ■	$P_{0,N-1}$
$P_{1,-1}$	$P_{1,0}$			■ ■ ■
■ ■ ■	■ ■ ■			
$P_{M-1,-1}$	$P_{M-1,0}$	■ ■ ■		$P_{M-1,N-1}$

M = N = 4 en esta figura

FIG. 4

$r_{0,0}$	$r_{0,1}$	$r_{0,2}$	...	$r_{0,(N-1)}$
$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	...	$r_{1,(N-1)}$
$r_{2,0}$	$r_{2,1}$	$r_{2,2}$	...	$r_{2,(N-1)}$
...	...	...	...	...
$r_{(M-1),0}$	$r_{(M-1),1}$	$r_{(M-1),2}$	...	$r_{(M-1),(N-1)}$

FIG. 5A

$r_{0,0}$	$r_{0,1}$	$r_{0,2}$	...	$r_{0,(N-1)}$
$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	...	$r_{1,(N-1)}$
$r_{2,0}$	$r_{2,1}$	$r_{2,2}$	...	$r_{2,(N-1)}$
...	...	...	...	...
$r_{(M-1),0}$	$r_{(M-1),1}$	$r_{(M-1),2}$	...	$r_{(M-1),(N-1)}$

FIG. 5B

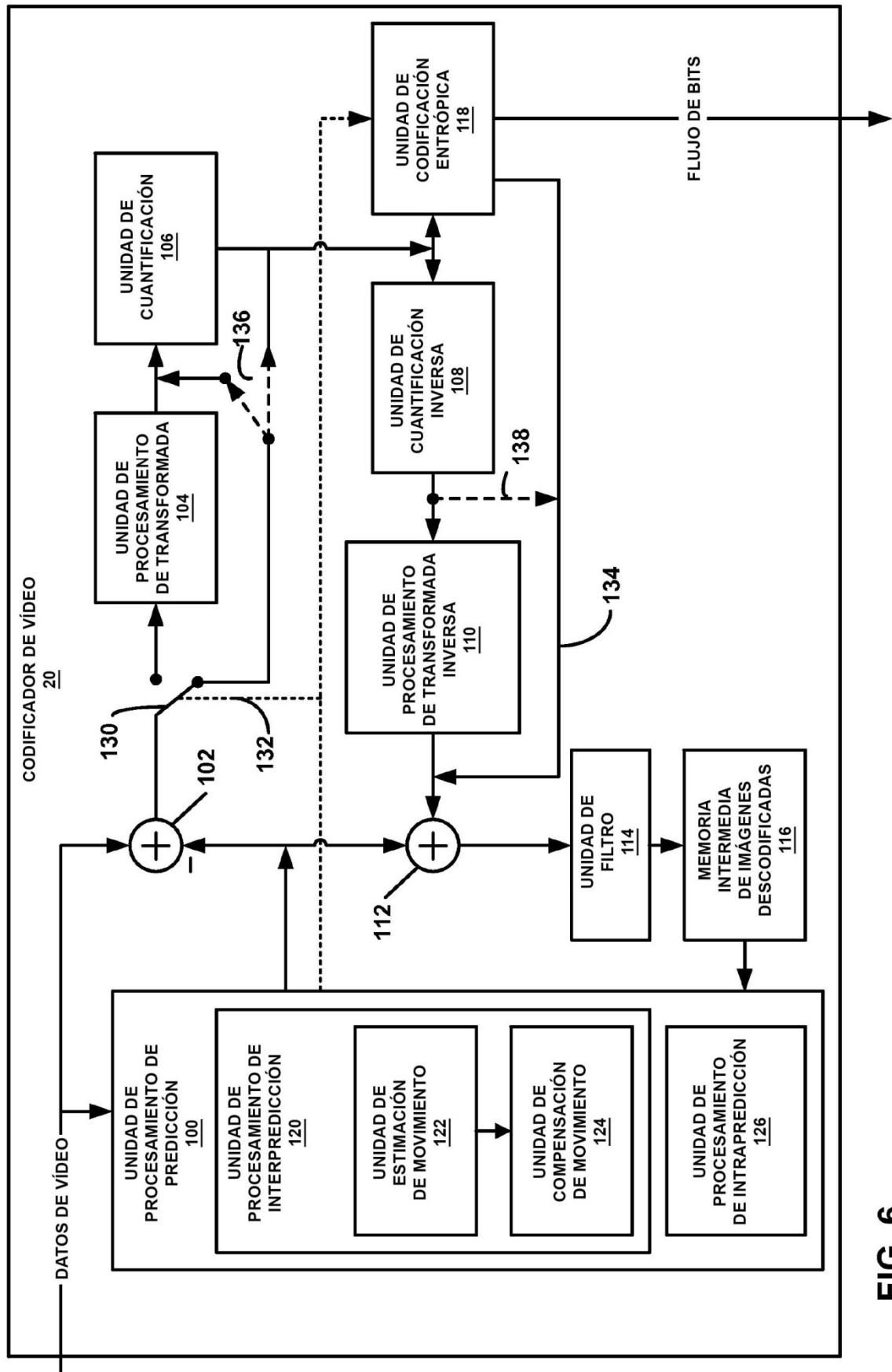


FIG. 6

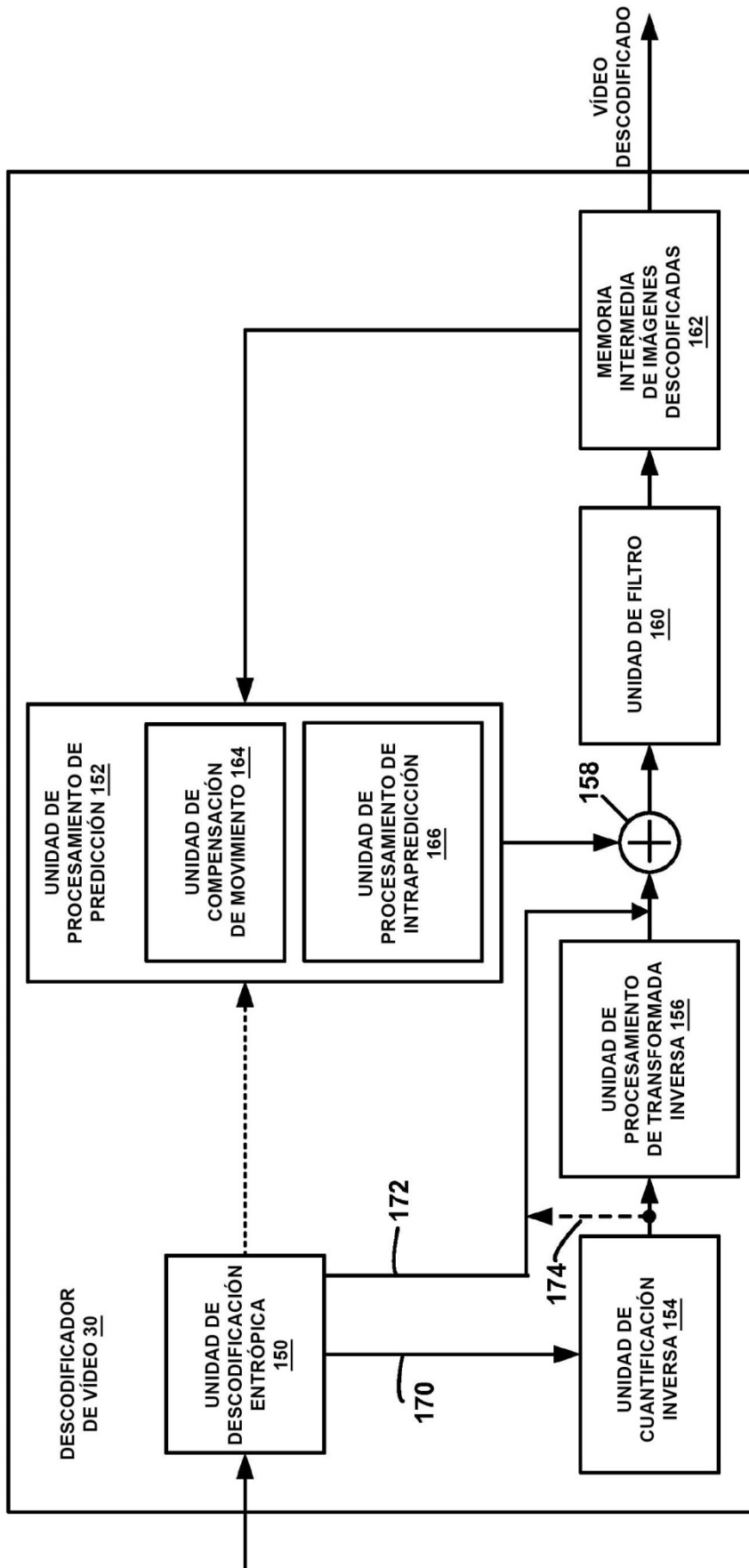


FIG. 7

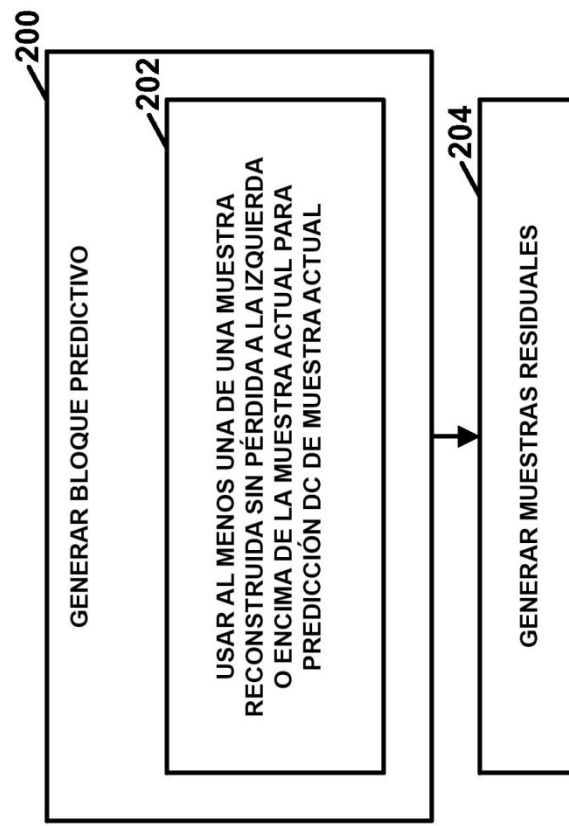


FIG. 8A

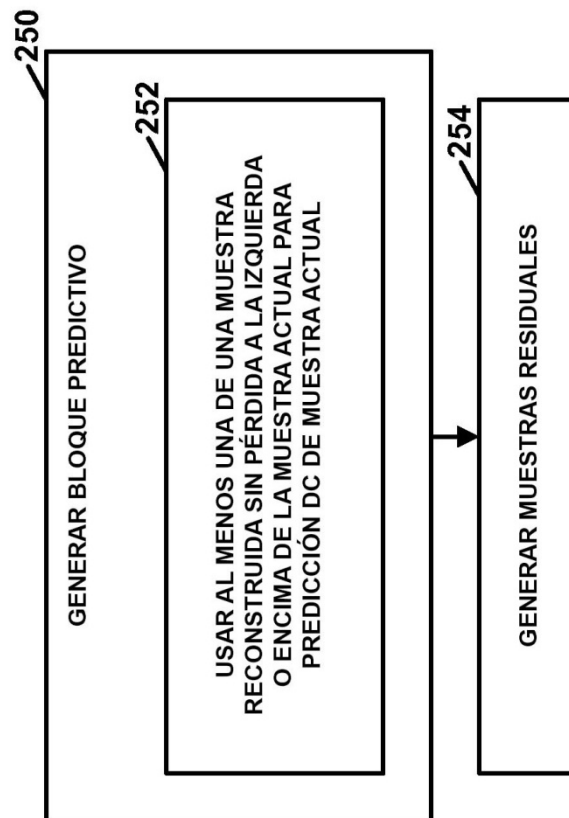


FIG. 8B

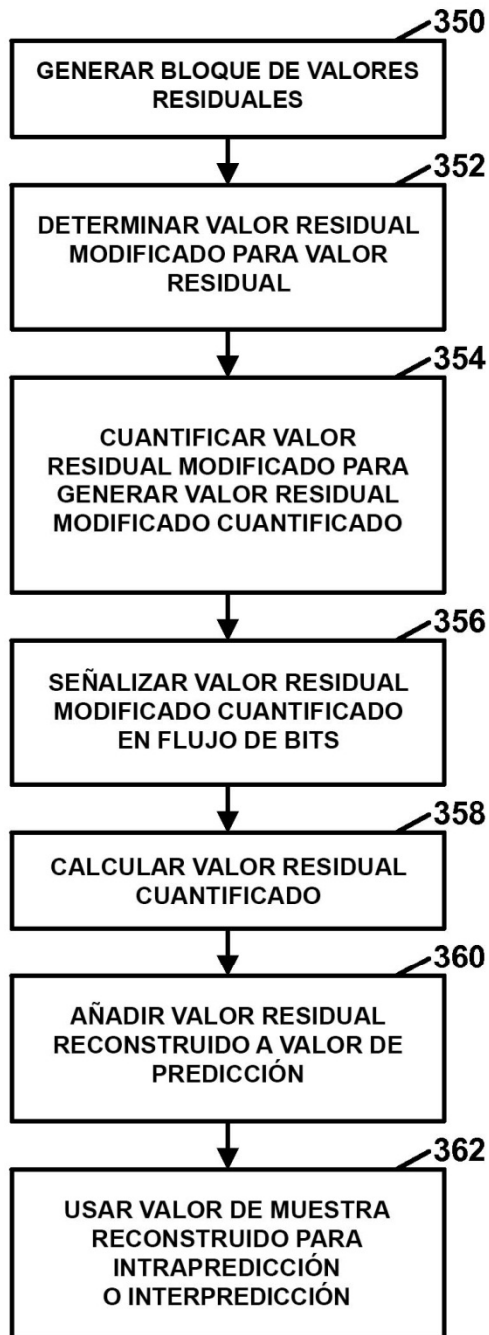


FIG. 9A

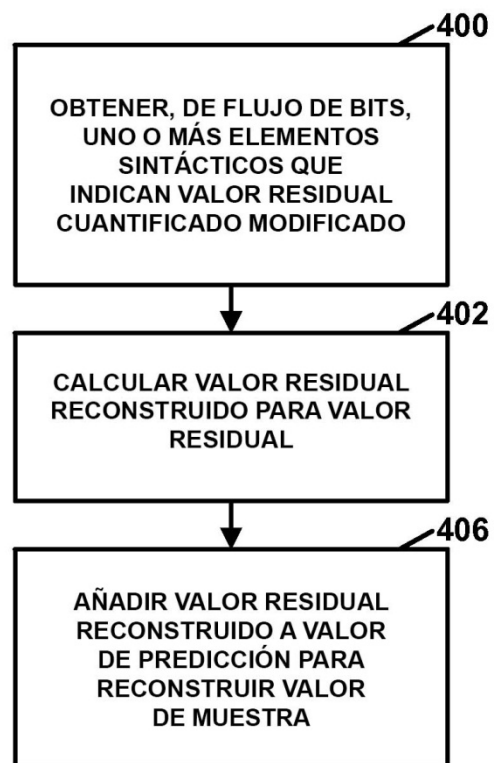


FIG. 9B

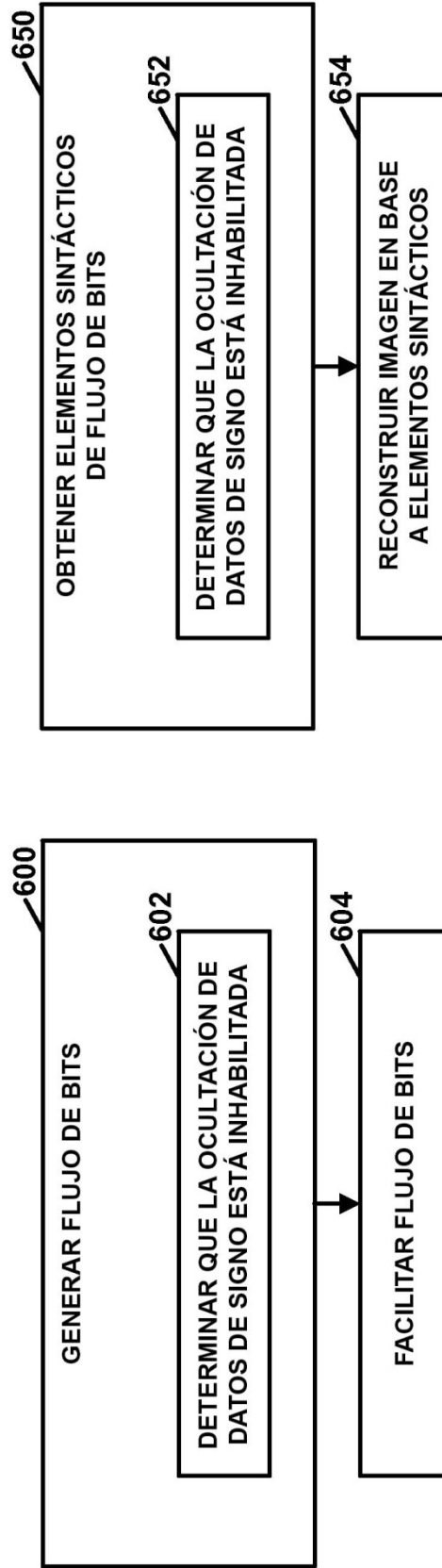


FIG. 10A

FIG. 10B