

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 019 082**

51 Int. Cl.:

<b>H04N 19/119</b>	(2014.01)
<b>H04N 19/96</b>	(2014.01)
<b>H04N 19/157</b>	(2014.01)
<b>H04N 19/70</b>	(2014.01)
<b>H04N 19/186</b>	(2014.01)
<b>H04N 19/103</b>	(2014.01)
<b>H04N 19/134</b>	(2014.01)
<b>H04N 19/176</b>	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **14.08.2020 PCT/US2020/046501**
- 87 Fecha y número de publicación internacional: **18.02.2021 WO21030747**
- 96 Fecha de presentación y número de la solicitud europea: **14.08.2020 E 20853492 (5)**
- 97 Fecha y número de publicación de la concesión europea: **26.02.2025 EP 4005215**

54 Título: **Restricción del tamaño de los bloques de crominancia pequeños en la codificación de video**

30 Prioridad:

**15.08.2019 US 201962887650 P**  
**13.09.2019 US 201962900424 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**20.05.2025**

73 Titular/es:

**BEIJING DAJIA INTERNET INFORMATION TECHNOLOGY CO., LTD. (100.00%)**  
**Room 101D1-7, 1st Floor, Building 1 No.6, Shangdi West Road, Haidian District Beijing 100085, CN**

72 Inventor/es:

**CHEN, YI-WEN;**  
**XIU, XIAOYU;**  
**MA, TSUNG-CHUAN;**  
**JHU, HONG-JHENG;**  
**WANG, XIANGLIN y**  
**YU, BING**

74 Agente/Representante:

**RUIZ GALLEGOS, Natalia**

**ES 3 019 082 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

Restricción del tamaño de los bloques de crominancia pequeños en la codificación de vídeo

## 5 CAMPO TÉCNICO

La presente solicitud en general se refiere a la codificación y compresión de datos de vídeo y, en particular, a un método y sistema de restricción del tamaño de los bloques disponibles más pequeños en la codificación de vídeo.

## 10 ANTECEDENTES

El vídeo digital es compatible con una gran variedad de dispositivos electrónicos, tales como televisiones digitales, ordenadores portátiles o de escritorio, tabletas, cámaras digitales, dispositivos digitales de grabación, reproductores digitales multimedia, consolas de videojuegos, teléfonos inteligentes, dispositivos de teleconferencia de vídeo, dispositivos de transmisión de vídeo, etc. Los dispositivos electrónicos transmiten, reciben, codifican, decodifican y/o almacenan datos de vídeo digital mediante la implementación de estándares de compresión/descompresión de vídeo según se definen por MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, la Codificación Avanzada de Vídeo (AVC), la Codificación de Vídeo de Alta Eficiencia (HEVC) y la Codificación de Vídeo Versátil (VVC). La compresión de vídeo normalmente incluye realizar la predicción espacial (intra-fotograma) y/o la predicción temporal (inter-fotograma) para reducir o eliminar la redundancia inherente en los datos de vídeo. Para la codificación de vídeo basada en bloques, un fotograma de vídeo se divide en uno o más segmentos y cada segmento tiene múltiples bloques de vídeo, los cuales también pueden ser referidos como unidades de codificación en árbol (CTU). Cada CTU puede contener una unidad de codificación (CU) o dividirse de forma recursiva en unidades de codificación más pequeñas hasta alcanzar el tamaño mínimo predefinido de unidad de codificación. Cada CU (también denominada CU en hoja) contiene una o múltiples unidades de transformada (TU) y cada CU también contiene una o múltiples unidades de predicción (PU). Cada CU puede codificarse en los modos intra, inter o IBC. Los bloques de vídeo en un segmento intracodificado (I) de un fotograma de vídeo se codifican mediante la predicción espacial con respecto a las muestras de referencia en los bloques contiguos dentro del mismo fotograma de vídeo. Los bloques de vídeo en un segmento intercodificado (P o B) de un fotograma de vídeo pueden utilizar la predicción espacial con respecto a las muestras de referencia en los bloques contiguos dentro del mismo fotograma de vídeo o la predicción temporal con respecto a las muestras de referencia en otros fotogramas de vídeo de referencia previos y/o futuros.

La predicción espacial o temporal con base en un bloque de referencia que se ha codificado previamente, por ejemplo, un bloque contiguo, da como resultado un bloque predictivo para la codificación de un bloque de vídeo actual. El proceso de encontrar el bloque de referencia puede realizarse mediante un algoritmo de coincidencia de bloques. Los datos residuales que representan diferencias de píxeles entre el bloque actual que se va a codificar y el bloque predictivo son referidos como bloque residual o errores de predicción. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta hacia un bloque de referencia en un fotograma de referencia que forma el bloque predictivo, y el bloque residual. El proceso de determinar el vector de movimiento normalmente es referido como estimación de movimiento. Un bloque intracodificado se codifica de acuerdo con un modo de intra-predicción y con el bloque residual. Para una mayor compresión, el bloque residual se transforma, del dominio de píxeles a un dominio de transformada, por ejemplo, un dominio de frecuencia, lo que da como resultado coeficientes residuales de transformada que posteriormente pueden cuantificarse. Los coeficientes de transformada cuantificados, inicialmente dispuestos en una matriz bidimensional, pueden explorarse para producir un vector unidimensional de coeficientes de transformada para después codificarse por entropía en un flujo de bits de vídeo y así obtener una compresión más uniforme.

A continuación, el flujo de bits de vídeo codificados se guarda en un medio de almacenamiento legible por ordenador (por ejemplo, una memoria flash) a la que se puede acceder a través de otro dispositivo electrónico con capacidad de vídeo digital o transmitirse directamente al dispositivo electrónico de forma inalámbrica o por cable. El dispositivo electrónico luego realiza la descompresión de vídeo (que es un proceso opuesto a la compresión de vídeo antes descrita) al, por ejemplo, analizar sintácticamente el flujo de bits de vídeo codificados para obtener elementos de sintaxis del flujo de bits y reconstruir los datos de vídeo digital a su formato original a partir del flujo de bits de vídeo codificados con base, por lo menos en parte, en los elementos de sintaxis obtenidos del flujo de bits, y presenta los datos de vídeo digital reconstruidos en una pantalla del dispositivo electrónico.

Z-Y LIN ET AL: "CE3-related: Constrained partitioning of chromaintra CBs", 126. MPEG MEETING; 20190325 - 20190329; GENEVA;n.º m46738, JVET-N0082-v1 13 de marzo de 2019 (13-03-2019) describe un método para no permitir croma intra CB menores de 16 muestras de croma al restringir la partición de croma intra CB. Un decodificador de vídeo configurable para determinar que un bloque de los datos de vídeo está formateado de acuerdo con un 4: 4:4 se conoce a partir del documento WO 2021/026564 A1.

Con una calidad de vídeo digital que va desde una alta definición hasta 4Kx2K o incluso hasta 8Kx4K, la cantidad de datos de vídeo que se codificarán/decodificarán aumenta exponencialmente. Se trata de un desafío constante en

términos de cómo pueden codificarse/decodificarse los datos de vídeo de forma más eficaz, al mismo tiempo que se conserva la calidad de la imagen de los datos de vídeo decodificados.

## SUMARIO

5

La presente solicitud describe implementaciones relacionadas con la codificación y decodificación de datos de vídeo y, de manera más particular, con un sistema y método de restricción del tamaño de los bloques disponibles más pequeños en la codificación de vídeo.

10

De acuerdo con un primer aspecto de la presente solicitud, se describe un método para decodificar datos de vídeo de acuerdo con las características de la reivindicación 1. El método incluye recibir, del flujo de bits, datos que codifican un nodo de codificación en árbol, en donde el nodo de codificación en árbol se asocia con un método de división predefinido; determinar un tipo de modo del nodo de codificación en árbol, incluyendo: de acuerdo con una determinación de que el nodo de codificación en árbol está codificado en un formato de submuestreo de crominancia 4:4:4 o 4:0:0, establecer el nodo de codificación en árbol en un primer tipo de modo; de acuerdo con una determinación de que el nodo de codificación en árbol está codificado en un formato de submuestreo de crominancia 4:2:2 o 4:2:0 y que el nodo de codificación en árbol es una unidad de intra-predicción de crominancia más pequeña (SCIPU): establecer el nodo de codificación en árbol en un segundo tipo de modo si un bloque de luminancia del nodo de codificación en árbol y el método de división cumplen con un primer criterio; y establecer el nodo de codificación en árbol en un tercer tipo de modo si el bloque de luminancia del nodo de codificación en árbol y el método de división cumplen con un segundo criterio; y decodificar la unidad de codificación en árbol con base en el tipo de modo, incluyendo: de acuerdo con una determinación de que el nodo de codificación en árbol es el segundo modo: decodificar cada bloque de codificación en el nodo de codificación en árbol mediante el modo de intra-predicción, de acuerdo con una determinación de que el nodo de codificación en árbol es el primer modo o el tercer modo: recibir, del flujo de bits, un elemento de sintaxis que indica un modo de predicción asociado con el nodo de codificación en árbol: de acuerdo con una determinación de que el elemento de sintaxis tiene un primer valor: decodificar cada bloque de codificación en el nodo de codificación en árbol mediante el modo de inter-predicción; y, de acuerdo con una determinación de que el elemento de sintaxis tiene un segundo valor: decodificar cada bloque de codificación en el nodo de codificación en árbol mediante el modo de intra-predicción.

30

De acuerdo con un segundo aspecto de la presente solicitud, un aparato electrónico incluye una o más unidades de procesamiento, memoria y una pluralidad de programas almacenados en la memoria. Los programas, cuando son ejecutados por la una o más unidades de procesamiento, hacen que el aparato electrónico realice el método de decodificación de datos de vídeo como se describió anteriormente.

35

De acuerdo con un tercer aspecto de la presente solicitud, un medio de almacenamiento no transitorio y legible por ordenador almacena una pluralidad de programas para su ejecución a través de un aparato electrónico que tiene una o más unidades de procesamiento. Los programas, cuando son ejecutados por la una o más unidades de procesamiento, hacen que el aparato electrónico realice el método de decodificación de datos de vídeo como se describió anteriormente.

40

## BREVE DESCRIPCIÓN DE LOS DIBUJOS

45

Los dibujos adjuntos, que se incluyen para proporcionar una comprensión adicional de las implementaciones y se incorporan a la presente y constituyen una parte de la memoria descriptiva, ilustran las implementaciones descritas y, junto con la descripción, sirven para explicar los principios subyacentes. Números de referencia similares hacen referencia a las partes correspondientes.

50

La figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo de acuerdo con algunas implementaciones de la presente divulgación.

La figura 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo de acuerdo con algunas implementaciones de la presente divulgación.

55

La figura 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo de acuerdo con algunas implementaciones de la presente divulgación.

60

Las figuras 4A a 4E son diagramas de bloques que ilustran la manera en que un fotograma se divide de forma recursiva en múltiples bloques de vídeo de diferentes tamaños y formas de acuerdo con algunas implementaciones de la presente divulgación.

La figura 5 es un diagrama de bloques que ilustra ejemplos de unidades de intra-predicción de crominancia más pequeñas (SCIPU) de acuerdo con algunas implementaciones de la presente divulgación.

Las figuras 6A a 6C son diagramas de bloques que ilustran ejemplos de formatos de crominancia que son compatibles con la VVC de acuerdo con algunas implementaciones de la presente divulgación.

5 La figura 7 es un diagrama de bloques que ilustra ejemplos de unidades de intra-predicción de crominancia más pequeñas (SCIPU) en diversos formatos de crominancia de acuerdo con algunas implementaciones de la presente divulgación.

10 La figura 8 es un diagrama de flujo que ilustra un proceso ejemplar mediante el cual un decodificador de vídeo implementa las técnicas de decodificar una unidad de intra-predicción de crominancia más pequeña (SCIPU) de acuerdo con algunas implementaciones de la presente divulgación.

### DESCRIPCIÓN DETALLADA

15 A continuación, se hará referencia detallada a las implementaciones específicas, cuyos ejemplos se ilustran en los dibujos adjuntos. En la siguiente descripción detallada, se expone una serie de detalles específicos no limitativos con el objetivo de ayudar a comprender la materia objeto que se presenta en el presente documento. Pero será evidente para un experto en la técnica que pueden utilizarse diversas alternativas sin apartarse del alcance de las reivindicaciones y que la materia objeto puede practicarse sin estos detalles específicos. Por ejemplo, será evidente para un experto en la técnica que la materia objeto que se presenta en este documento puede implementarse en distintos tipos de dispositivos electrónicos con capacidades de vídeo digital.

20 La figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema 10 para la codificación y decodificación de bloques de vídeo en paralelo de acuerdo con algunas implementaciones de la presente divulgación. Tal como se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que genera y codifica datos de vídeo para que un dispositivo de destino 14 los decodifique posteriormente. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia variedad de dispositivos electrónicos, incluyendo ordenadores o portátiles, tablet, teléfonos inteligentes, convertidores de señal de TV, televisiones digitales, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo o similares. En algunas implementaciones, el dispositivo de origen 12 y el dispositivo de destino 14 están equipados con capacidades de comunicación inalámbrica.

25 En algunas implementaciones, el dispositivo de destino 14 puede recibir los datos de vídeo codificados que se decodificarán a través de un enlace 16. El enlace 16 puede incluir cualquier tipo de medio de comunicación o dispositivo capaz de trasladar los datos de vídeo codificados del dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita los datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como el Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

30 En otras implementaciones, los datos de vídeo codificados pueden transmitirse de la interfaz de salida 22 a un dispositivo de almacenamiento 32. Posteriormente, el dispositivo de destino 14 puede acceder a los datos de vídeo codificados del dispositivo de almacenamiento 32 a través de la interfaz de entrada 28. El dispositivo de almacenamiento 32 puede incluir cualquiera de una diversidad de medios de almacenamiento de datos distribuidos o a los que se accede localmente, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 32 puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que pueda contener los datos de vídeo codificados, generados por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento 32 mediante transmisión por secuencias o descarga. El servidor de archivos puede ser cualquier tipo de ordenador 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 un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, módem por cable, etc.) o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados, almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 32 puede ser una transmisión por secuencias, una transmisión por descarga o una combinación de ambas.

5 Tal como se muestra en la figura 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. La fuente de vídeo 18 puede incluir una fuente, tal como un dispositivo de captura de vídeo, por ejemplo una cámara de vídeo, un archivo de vídeo que contenga vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo y/o un sistema de gráficos de ordenador para generar datos de gráficos de ordenador como el vídeo de origen, o una combinación de tales fuentes. Como ejemplo, si la fuente de vídeo 18 es una cámara de vídeo de un sistema de vigilancia de seguridad, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar teléfonos con cámara o teléfonos con vídeo. Sin embargo, las implementaciones descritas en la presente solicitud pueden ser aplicables a la codificación de vídeo en general y pueden aplicarse a aplicaciones inalámbricas y/o por cable.

10 El vídeo capturado, precapturado o generado por ordenador puede codificarse a través del codificador de vídeo 20. Los datos de vídeo codificados pueden transmitirse directamente al dispositivo de destino 14 a través de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados también (o de manera alternativa) pueden almacenarse en el dispositivo de almacenamiento 32 para un posterior acceso por el dispositivo de destino 14 u otros dispositivos, para la decodificación y/o reproducción. La interfaz de salida 22 también puede incluir un módem y/o un transmisor.

20 El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 34. La interfaz de entrada 28 puede incluir un receptor y/o un módem y recibir los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados, comunicados a través del enlace 16 o proporcionados en el dispositivo de almacenamiento 32, pueden incluir una diversidad de elementos de sintaxis generados por el codificador de vídeo 20 para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo. Tales elementos de sintaxis pueden incluirse en los datos de vídeo codificados, ya sea transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

25 En algunas implementaciones, el dispositivo de destino 14 puede incluir un dispositivo de visualización 34, el cual puede ser un dispositivo de visualización integrado y un dispositivo de visualización externo que se configure para comunicarse con el dispositivo de destino 14. El dispositivo de visualización 34 muestra los datos de vídeo decodificados a un usuario y puede comprender cualquiera de una diversidad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo emisor de luz orgánico (OLED) u otro tipo de dispositivo de visualización.

35 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con estándares de propiedad exclusiva o de la industria, tales como los estándares VVC, HEVC, MPEG-4, Parte 10, la Codificación de Vídeo Avanzada (AVC) o extensiones de tales estándares. Debe entenderse que la presente solicitud no se limita a un estándar específico de codificación/decodificación de vídeo y que puede aplicarse a otros estándares de codificación/decodificación de vídeo. Por lo general, se contempla que el codificador de vídeo 20 del dispositivo de origen 12 pueda configurarse para codificar datos de vídeo de acuerdo con cualquiera de estos estándares actuales o futuros. De igual manera, por lo general también se contempla que el decodificador de vídeo 30 del dispositivo de destino 14 pueda configurarse para decodificar datos de vídeo de acuerdo con cualquiera de estos estándares actuales o futuros.

45 Cada uno del codificador de vídeo 20 y el decodificador de vídeo 30 puede implementarse como cualquiera de una diversidad de circuitos de codificador adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables de campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando se implementa parcialmente en el software, un dispositivo electrónico puede almacenar instrucciones para el software en un medio no transitorio y legible por ordenador que sea adecuado y ejecutar las instrucciones en el hardware a través de uno o más procesadores para realizar las operaciones de codificación/decodificación de vídeo que se dan a conocer en la presente divulgación. Cada uno del codificador de vídeo 20 y el decodificador de vídeo 30 pueden incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo.

55 La figura 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 de acuerdo con algunas implementaciones descritas en la presente solicitud. El codificador de vídeo 20 puede realizar la intra-codificación y la inter-predicción predictiva de bloques de vídeo dentro de los fotogramas de vídeo. La intra-codificación predictiva se basa en la predicción espacial para reducir o eliminar la redundancia espacial en los datos de vídeo dentro de una imagen o fotograma de vídeo determinado. La inter-codificación predictiva se basa en la predicción temporal para reducir o eliminar la redundancia temporal en los datos de vídeo dentro de imágenes o fotogramas de vídeo adyacentes de una secuencia de vídeo.

60 Tal como se muestra en la figura 2, el codificador de vídeo 20 incluye una memoria de datos de vídeo 40, una unidad de procesamiento de predicciones 41, una memoria intermedia de imágenes decodificadas (DPB) 64, una sumadora 50, una unidad de procesamiento de transformada 52, una unidad de cuantificación 54 y una unidad de codificación

de entropía 56. La unidad de procesamiento de predicciones 41 también incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44, una unidad de división 45, una unidad de procesamiento de intra-predicciones 46 y una unidad de copia intra-bloque (BC) 48. En algunas implementaciones, el codificador de vídeo 20 también incluye una unidad de cuantificación inversa 58, una unidad de procesamiento de transformada inversa 60 y una sumadora 62 para la reconstrucción de bloques de vídeo. Un filtro de desbloqueo (no se muestra) puede situarse entre la sumadora 62 y la DPB 64 para filtrar los límites de bloques y eliminar los artefactos de bloqueo del vídeo reconstruido. También se puede utilizar un filtro en bucle (no se muestra) además del filtro de desbloqueo para filtrar el resultado de la sumadora 62. El codificador de vídeo 20 puede adoptar la forma de una unidad de hardware fija o programable o puede dividirse entre una o más de las unidades de hardware fijas o programables ilustradas.

La memoria de datos de vídeo 40 puede almacenar datos de vídeo para codificarse mediante los componentes del codificador de vídeo 20. Los datos de vídeo de la memoria de datos de vídeo 40 pueden obtenerse, por ejemplo, de la fuente de vídeo 18. La DPB 64 es una memoria intermedia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo a través del codificador de vídeo 20 (por ejemplo, en los modos intra-codificación o inter-codificación predictiva). La memoria de datos de vídeo 40 y la DPB 64 pueden conformarse por cualquiera de una diversidad de dispositivos de memoria. En diversos ejemplos, la memoria de datos de vídeo 40 puede estar en chip con otros componentes del codificador de vídeo 20, o fuera de chip en relación con esos componentes.

Tal como se muestra en la figura 2, después de recibir los datos de vídeo, la unidad de división 45 dentro de la unidad de procesamiento de predicciones 41 divide los datos de vídeo en bloques de vídeo. Esta división también puede incluir dividir un fotograma de vídeo en segmentos, mosaicos u otras unidades de codificación (CU) más grandes de acuerdo con estructuras de división predefinidas, tal como una estructura de árbol cuaternario asociada con los datos de vídeo. El fotograma de vídeo puede dividirse en múltiples bloques de vídeo (o conjuntos de bloques de vídeo referidos como mosaicos). La unidad de procesamiento de predicciones 41 puede seleccionar uno de una pluralidad de posibles modos de codificación predictiva, tal como uno de una pluralidad de modos de intra-codificación predictiva o uno de una pluralidad de modos de inter-codificación predictiva, para el bloque de vídeo actual con base en los resultados de error (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicciones 41 puede proporcionar el bloque de predicción intracodificado o intercodificado resultante a la sumadora 50 para generar un bloque residual y a la sumadora 62 para reconstruir el bloque codificado para su uso posterior como parte de un fotograma de referencia. La unidad de procesamiento de predicciones 41 también proporciona elementos de sintaxis, tales como vectores de movimiento, indicadores de modo intra, información de divisiones y otra información de sintaxis, a la unidad de codificación de entropía 56.

Para seleccionar un modo adecuado de intra-codificación predictiva para el bloque de vídeo actual, la unidad de procesamiento de intra-predicciones 46 dentro de la unidad de procesamiento de predicciones 41 puede realizar la intra-codificación predictiva del bloque de vídeo actual en relación con uno o más bloques contiguos en el mismo fotograma que el bloque actual que se va a codificar para proporcionar la predicción espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 dentro de la unidad de procesamiento de predicciones 41 realizan la inter-codificación predictiva del bloque de vídeo actual en relación con uno o más bloques predictivos en uno o más fotogramas de referencia para proporcionar una predicción temporal. El codificador de vídeo 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación adecuado para cada bloque de datos de vídeo.

En algunas implementaciones, la unidad de estimación de movimiento 42 determina el modo de inter-predicción para un fotograma de vídeo actual mediante la generación de un vector de movimiento, el cual indica el desplazamiento de una unidad de predicción (PU) de un bloque de vídeo dentro del fotograma de vídeo actual en relación con un bloque predictivo dentro de un fotograma de vídeo de referencia, de acuerdo con un patrón predeterminado dentro de una secuencia de fotogramas de vídeo. La estimación del movimiento, realizada a través de la unidad de estimación del movimiento 42, es el proceso de generar vectores de movimiento, los cuales estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de un fotograma de vídeo o imagen actual en relación con un bloque predictivo dentro de un fotograma de referencia (u otra unidad codificada) en relación con el bloque actual que se codifica dentro del fotograma actual (u otra unidad codificada). El patrón predeterminado puede designar fotogramas de vídeo en la secuencia como fotogramas P o fotogramas B. La unidad intra-bloque BC 48 puede determinar vectores, por ejemplo, vectores de bloque, para la intra-codificación de BC, de forma similar a la determinación de vectores de movimiento a través de la unidad de estimación de movimiento 42 para la inter-predicción, o puede utilizar la unidad de estimación de movimiento 42 para determinar el vector de bloque.

Un bloque predictivo es un bloque de un fotograma de referencia que se considera que coincide estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxeles, el cual puede determinarse mediante la suma de las diferencias absolutas (SAD), la suma de las diferencias cuadradas (SSD) u otras métricas de diferencia. En algunas implementaciones, el codificador de vídeo 20 puede calcular valores para posiciones de píxeles subenteros de fotogramas de referencia almacenados en la DPB 64. Por ejemplo, el codificador de vídeo 20 puede

interpolarse valores de posiciones de un cuarto de píxel, posiciones de un octavo píxel u otras posiciones de píxeles fraccionarios del fotograma de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completos y las posiciones de píxeles fraccionarios y generar un vector de movimiento con una precisión de píxel fraccionario.

5

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fotograma de predicción intercodificado al comparar la posición de la PU con la posición de un bloque predictivo de un fotograma de referencia seleccionado de una primera lista de fotogramas de referencia (Lista 0) o de una segunda lista de fotogramas de referencia (Lista 1), cada una de las cuales identifica uno o más fotogramas de referencia almacenados en la DPB 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de compensación de movimiento 44 y luego a la unidad de codificación de entropía 56.

10

La compensación de movimiento, realizada a través de la unidad de compensación de movimiento 44, puede implicar recolectar o generar el bloque predictivo con base en el vector de movimiento determinado por la unidad de estimación de movimiento 42. Una vez que recibe el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar un bloque predictivo al que apunta el vector de movimiento en una de las listas de fotogramas de referencia, recuperar el bloque predictivo de la DPB 64 y reenviar el bloque predictivo a la sumadora 50. Después, la sumadora 50 forma un bloque de vídeo residual de valores de diferencia de píxeles al restar los valores de píxeles del bloque predictivo proporcionado por la unidad de compensación de movimiento 44 de los valores de píxeles del bloque de vídeo actual que se está codificando. Los valores de diferencia de píxeles que forman el bloque de vídeo residual pueden incluir componentes de diferencia de luminancia o crominancia o ambos. La unidad de compensación de movimiento 44 también puede generar elementos de sintaxis asociados con los bloques de vídeo de un fotograma de vídeo para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fotograma de vídeo. Los elementos de sintaxis pueden incluir, por ejemplo, elementos de sintaxis que definen el vector de movimiento utilizado para identificar el bloque predictivo, cualquier indicador que indique el modo de predicción o cualquier otra información de sintaxis descrita en el presente documento. Es importante señalar que la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar totalmente integradas, pero se ilustran por separado para fines conceptuales.

15

20

25

30

En algunas implementaciones, la unidad intra-bloque BC 48 puede generar vectores y recolectar bloques predictivos de forma similar a la descrita anteriormente en relación con la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, pero los bloques predictivos están en el mismo fotograma que el bloque actual que se está codificando y los vectores son referidos como vectores de bloque en vez de vectores de movimiento. En particular, la unidad intra-bloque BC 48 puede determinar un modo intra-predicción que se utilizará para codificar un bloque actual. En algunos ejemplos, la unidad intra-bloque BC 48 puede codificar un bloque actual con el uso de distintos modos de intra-predicción, por ejemplo, durante pasadas de codificación separadas, y probar su rendimiento mediante un análisis de distorsión de la velocidad. Después, la unidad intra-bloque BC 48 puede seleccionar, entre los distintos modos de intra-predicción probados, un modo de intra-predicción adecuado para utilizarse y generar un indicador de modo intra en consecuencia. Por ejemplo, la unidad intra-bloque BC 48 puede calcular los valores de distorsión de la velocidad mediante un análisis de distorsión de la velocidad para los distintos modos de intra-predicción probados y seleccionar el modo de intra-predicción que tenga las mejores características de distorsión de la velocidad, de entre los modos probados, como el modo de intra-predicción adecuado que se va a utilizar. El análisis de distorsión de la velocidad por lo general determina un grado de distorsión (o de error) entre un bloque codificado y un bloque original sin codificar que se codificó para producir el bloque codificado, así como una velocidad de bits (es decir, un número de bits) utilizada para producir el bloque codificado. La unidad intra-bloque BC 48 puede calcular las proporciones a partir de las distorsiones y las velocidades para los distintos bloques codificados a fin de determinar cuál modo de intra-predicción muestra el mejor valor de distorsión de la velocidad para el bloque.

35

40

45

50

En otros ejemplos, la unidad intra-bloque BC 48 puede utilizar la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, en su totalidad o en parte, para realizar tales funciones para la intra-predicción de BC de acuerdo con las implementaciones descritas en el presente documento. En cualquier caso, para la copia intra-bloque, un bloque predictivo puede ser un bloque que se considere que coincide estrechamente con el bloque que se va a codificar, en términos de diferencia de píxeles, el cual puede determinarse mediante la suma de las diferencias absolutas (SAD), la suma de las diferencias cuadradas (SSD) u otras métricas de diferencia, y la identificación del bloque predictivo puede incluir el cálculo de valores para las posiciones de píxeles subenteros.

55

Ya sea que el bloque predictivo proceda del mismo fotograma de acuerdo con la intra-predicción, o de un fotograma diferente de acuerdo con la inter-predicción, el codificador de vídeo 20 puede formar un bloque de vídeo residual al restar los valores de píxeles del bloque predictivo de los valores de píxeles del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxeles. Los valores de diferencia de píxeles que forman el bloque de vídeo residual pueden incluir diferencias de componentes tanto de luminancia como de crominancia.

60

La unidad de procesamiento de intra-predicciones 46 puede intra-predecir un bloque de vídeo actual, como alternativa a la inter-predicción realizada a través de la unidad de estimación de movimiento 42 y la unidad de compensación de

movimiento 44, o la predicción de copia intra-bloque realizada a través de la unidad de intra-bloque BC 48, tal como se describió anteriormente. En particular, la unidad de procesamiento de intra-predicciones 46 puede determinar un modo de intra-predicción que se utilizará para codificar un bloque actual. Para ello, la unidad de procesamiento de intra-predicciones 46 puede codificar un bloque actual con el uso de distintos modos de intra-predicción, por ejemplo, durante pasadas de codificación separadas, y la unidad de procesamiento de intra-predicciones 46 (o una unidad de selección de modo, en algunos ejemplos) puede seleccionar un modo de intra-predicción adecuado para utilizarse a partir de los modos de intra-predicción probados. La unidad de procesamiento de intra-predicciones 46 puede proporcionar información indicativa del modo de intra-predicción seleccionado para el bloque a la unidad de codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica el modo de intra-predicción seleccionado en el flujo de bits.

Una vez que la unidad de procesamiento de predicciones 41 determina el bloque predictivo para el bloque de vídeo actual mediante la inter-predicción o la intra-predicción, la sumadora 50 forma un bloque de vídeo residual al restar el bloque predictivo del bloque de vídeo actual. Los datos de vídeo residuales del bloque residual pueden incluirse en una o más unidades de transformada (TU) y se proporcionan a la unidad de procesamiento de transformada 52. La unidad de procesamiento de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales mediante una transformada, tal como una transformada discreta del coseno (DCT) o una transformada conceptualmente similar.

La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir aún más la velocidad de bits. El proceso de cuantificación también puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede modificarse al ajustar un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 después puede realizar una exploración de una matriz que incluya los coeficientes de transformada cuantificados. De manera alternativa, la unidad de codificación de entropía 56 puede realizar la exploración.

Posterior a la cuantificación, la unidad de codificación de entropía 56 codifica por entropía los coeficientes de transformada cuantificados en un flujo de bits de vídeo con el uso, por ejemplo, de la codificación de longitud variable adaptativa según el contexto (CAVLC), la codificación aritmética binaria adaptativa según el contexto (CABAC), la codificación aritmética binaria adaptativa según el contexto basada en sintaxis (SBAC), la codificación de entropía por división de intervalos de probabilidad (PIPE) u otra técnica o metodología de codificación de entropía. Posteriormente, el flujo de bits codificados puede transmitirse al decodificador de vídeo 30, o archivarse en el dispositivo de almacenamiento 32 para su posterior transmisión o recuperación mediante el decodificador de vídeo 30. La unidad de codificación de entropía 56 también puede codificar por entropía los vectores de movimiento y los demás elementos de sintaxis para el fotograma de vídeo actual que se está codificando.

La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque de vídeo residual en el dominio de píxeles para generar un bloque de referencia para la predicción de otros bloques de vídeo. Tal como se señaló anteriormente, la unidad de compensación de movimiento 44 puede generar un bloque predictivo con compensación de movimiento a partir de uno o más bloques de referencia de los fotogramas almacenados en la DPB 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque predictivo para calcular los valores de píxeles subenteros para su uso en la estimación de movimiento.

La sumadora 62 añade el bloque residual reconstruido al bloque predictivo con compensación de movimiento producido por la unidad de compensación de movimiento 44 para producir un bloque de referencia para su almacenamiento en la DPB 64. Después, la unidad intra-bloque BC 48, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden utilizar el bloque de referencia como un bloque predictivo para inter-predicir otro bloque de vídeo en un fotograma de vídeo posterior.

La figura 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 de acuerdo con algunas implementaciones de la presente solicitud. El decodificador de vídeo 30 incluye una memoria de datos de vídeo 79, una unidad de decodificación de entropía 80, una unidad de procesamiento de predicciones 81, una unidad de cuantificación inversa 86, una unidad de procesamiento de transformada inversa 88, una sumadora 90 y una DPB 92. La unidad de procesamiento de predicciones 81 también incluye una unidad de compensación de movimiento 82, una unidad de procesamiento de intra-predicciones 84 y una unidad intra-bloque BC 85. El decodificador de vídeo 30 puede realizar un proceso de decodificación generalmente recíproco al proceso de codificación antes descrito con respecto al codificador de vídeo 20 en relación con la figura 2. Por ejemplo, la unidad de compensación de movimiento 82 puede generar datos de predicción con base en vectores de movimiento recibidos de la unidad de decodificación de entropía 80, mientras que la unidad de intra-predicciones 84 puede generar datos de predicción con base en indicadores de modo de intra-predicción recibidos de la unidad de decodificación de entropía 80.

En algunos ejemplos, a una unidad de decodificador de vídeo 30 se le puede asignar la tarea de realizar las

- implementaciones de la presente solicitud. Asimismo, en algunos ejemplos, las implementaciones de la presente divulgación pueden dividirse entre una o más de las unidades del decodificador de vídeo 30. Por ejemplo, la unidad de intra-bloque BC 85 puede realizar las implementaciones de la presente solicitud, sola o en combinación con otras unidades del decodificador de vídeo 30, tal como la unidad de compensación de movimiento 82, la unidad de procesamiento de intra-predicciones 84 y la unidad de decodificación de entropía 80. En algunos ejemplos, es posible que el decodificador de vídeo 30 no incluya la unidad intra-bloque BC 85 y la funcionalidad de la unidad intra-bloque BC 85 puede ser realizada por otros componentes de la unidad de procesamiento de predicciones 81, tal como la unidad de compensación de movimiento 82.
- 10 La memoria de datos de vídeo 79 puede almacenar datos de vídeo, tal como un flujo de bits de vídeo codificados, para su decodificación a través de los demás componentes del decodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 79 pueden obtenerse, por ejemplo, del dispositivo de almacenamiento 32, de una fuente de vídeo local, tal como una cámara, mediante la comunicación de datos de vídeo por red inalámbrica o por cable, o al acceder a medios de almacenamiento de datos físicos (por ejemplo, una unidad flash o un disco duro). La memoria de datos de vídeo 79 puede incluir una memoria intermedia de imágenes codificadas (CPB) que almacena datos de vídeo codificados de un flujo de bits de vídeo codificados. La memoria intermedia de imágenes decodificadas (DPB) 92 del decodificador de vídeo 30 almacena datos de vídeo de referencia para su uso en la decodificación de datos de vídeo a través del codificador de vídeo 30 (por ejemplo, en los modos de intra-codificación o inter-codificación predictiva). La memoria de datos de vídeo 79 y la DPB 92 puede conformarse por cualquiera de una variedad de dispositivos de memoria, tal como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magneto-resistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. Para fines ilustrativos, la memoria de datos de vídeo 79 y la DPB 92 se representan como dos componentes distintos del decodificador de vídeo 30 en la figura 3. Sin embargo, será evidente para un experto en la técnica que la memoria de datos de vídeo 79 y la DPB 92 puede proporcionarse en el mismo dispositivo de memoria o en dispositivos de memoria separados. En algunos ejemplos, la memoria de datos de vídeo 79 puede estar en chip con otros componentes del decodificador de vídeo 30, o fuera de chip en relación con esos componentes.
- 30 Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificados que representa los bloques de vídeo de un fotograma de vídeo codificado y elementos de sintaxis asociados. El decodificador de vídeo 30 puede recibir los elementos de sintaxis a nivel de fotograma de vídeo y/o a nivel de bloque de vídeo. La unidad de decodificación de entropía 80 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intra-predicción, y otros elementos de sintaxis. Después, la unidad de decodificación de entropía 80 reenvía los vectores de movimiento y otros elementos de sintaxis a la unidad de procesamiento de predicciones 81.
- 35 Cuando el fotograma de vídeo se codifica como un fotograma predictivo intracodificado (I) o para bloques predictivos intracodificados en otros tipos de fotogramas, la unidad de procesamiento de intra-predicciones 84 de la unidad de procesamiento de predicciones 81 puede generar datos de predicción para un bloque de vídeo del fotograma de vídeo actual con base en un modo de intra-predicción señalado y datos de referencia de los bloques previamente decodificados del fotograma actual.
- 40 Cuando el fotograma de vídeo se codifica como un fotograma predictivo intercodificado (es decir, B o P), la unidad de compensación de movimiento 82 de la unidad de procesamiento de predicciones 81 produce uno o más bloques predictivos para un bloque de vídeo del fotograma de vídeo actual con base en los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad de decodificación de entropía 80. Cada uno de los bloques predictivos puede producirse a partir de un fotograma de referencia dentro de una de las listas de fotogramas de referencia. El decodificador de vídeo 30 puede construir las listas de fotogramas de referencia, Lista 0 y Lista 1, con el uso de técnicas predeterminadas de construcción con base en fotogramas de referencia almacenadas en la DPB 92.
- 45 En algunos ejemplos, cuando el bloque de vídeo se codifica de acuerdo con el modo de copia intra-bloque que aquí se describe, la unidad intra-bloque (BC) 85 de la unidad de procesamiento de predicciones 81 produce bloques predictivos para el bloque de vídeo actual con base en vectores de bloque y otros elementos de sintaxis recibidos de la unidad de decodificación de entropía 80. Los bloques predictivos pueden estar dentro de una región reconstruida de la misma imagen que el bloque de vídeo actual definido por el codificador de vídeo 20.
- 55 La unidad de compensación de movimiento 82 y/o la unidad intra-bloque BC 85 determina la información de predicción para un bloque de vídeo del fotograma de vídeo actual al analizar sintácticamente los vectores de movimiento y otros elementos de sintaxis, y después utiliza la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 82 utiliza algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra-predicción o inter-predicción) utilizado para codificar los bloques de vídeo del fotograma de vídeo, un tipo de fotograma de inter-predicción (por ejemplo, B o P), información de construcción para una o más de las listas de fotogramas de referencia para el fotograma, vectores de movimiento para cada bloque de vídeo predictivo intercodificado del fotograma, estatus de inter-predicción para cada bloque de vídeo predictivo intercodificado del fotograma, así como otra información para
- 60

decodificar los bloques de vídeo en el fotograma de vídeo actual.

5 Del mismo modo, la unidad intra-bloque BC 85 puede utilizar algunos de los elementos de sintaxis recibidos, por ejemplo, un indicador, para determinar que el bloque de vídeo actual se predijo mediante el modo de copia intra-bloque, información de construcción de cuáles bloques de vídeo del fotograma se encuentran dentro de la región reconstruida y deben almacenarse en la DPB 92, vectores de bloque para cada bloque de vídeo previsto mediante copia intra-bloque del fotograma, estatus de predicción de copia intra-bloque para cada bloque de vídeo previsto mediante copia intra-bloque del fotograma, y otra información para decodificar los bloques de vídeo en el fotograma de vídeo actual.

10 La unidad de compensación de movimiento 82 también puede realizar la interpolación con el uso de los filtros de interpolación que utiliza el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular los valores interpolados para píxeles subenteros de los bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación que utiliza el codificador de vídeo 20 a partir de los elementos de sintaxis recibidos y utilizar los filtros de interpolación para producir bloques predictivos.

15 La unidad de cuantificación inversa 86 cuantifica en forma inversa los coeficientes de transformada cuantificados que se proporcionan en el flujo de bits y se decodifican por entropía por la unidad de decodificación de entropía 80 con el uso del mismo parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo en el fotograma de vídeo para determinar un grado de cuantificación. La unidad de procesamiento de transformada inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada inversa de números enteros o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada para reconstruir los bloques residuales en el dominio de píxeles.

20 Una vez que la unidad de compensación de movimiento 82 o la unidad intra-bloque BC 85 genera el bloque predictivo para el bloque de vídeo actual con base en los vectores y otros elementos de sintaxis, la sumadora 90 reconstruye el bloque de vídeo decodificado para el bloque de vídeo actual al sumar el bloque residual de la unidad de procesamiento de transformada inversa 88 y un bloque predictivo correspondiente generado por la unidad de compensación de movimiento 82 y la unidad intra-bloque BC 85. Un filtro en bucle (no se muestra) puede situarse entre la sumadora 90 y la DPB 92 para procesar aún más el bloque de vídeo decodificado. Los bloques de vídeo decodificados en un fotograma determinado se almacenan después en la DPB 92, la cual almacena los fotogramas de referencia utilizados para la posterior compensación de movimiento de los siguientes bloques de vídeo. La DPB 92, o un dispositivo de memoria independiente de la DPB 92, también puede almacenar vídeo decodificado para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 34 de la figura 1.

30 En un proceso típico de codificación de vídeo, una secuencia de vídeo normalmente incluye un conjunto ordenado de fotogramas o imágenes. Cada fotograma puede incluir tres matrices de muestra, denotadas como SL, SCb y SCr. SL es una matriz bidimensional de muestras de luminancia. SCb es una matriz bidimensional de muestras de crominancia de Cb. SCr es una matriz bidimensional de muestras de crominancia de Cr. En otros casos, un fotograma puede ser monocromático y, por lo tanto, incluye sólo una matriz bidimensional de muestras de luminancia.

35 Tal como se muestra en la figura 4A, el codificador de vídeo 20 (o, más específicamente, la unidad de división 45) genera una representación codificada de un fotograma al primero dividir el fotograma en un conjunto de unidades de codificación en árbol (CTU). Un fotograma de vídeo puede incluir un número entero de CTU ordenadas de forma consecutiva en un orden de exploración de trama de izquierda a derecha y de arriba a abajo. Cada CTU es una unidad de codificación lógica más grande y la anchura y la altura de la CTU se señalan mediante el codificador de vídeo 20 en un conjunto de parámetros de secuencia, de tal modo que todas las CTU de una secuencia de vídeo tienen el mismo tamaño, como por ejemplo 128 x 128, 64 x 64, 32 x 32 y 16 x 16. Sin embargo, cabe señalar que la presente solicitud no necesariamente se limita a un tamaño específico. Tal como se muestra en la figura 4B, cada CTU puede comprender un bloque de codificación en árbol (CTB) de muestras de luminancia, dos bloques de codificación en árbol correspondientes de muestras de crominancia y elementos de sintaxis utilizados para codificar las muestras de los bloques de codificación en árbol. Los elementos de sintaxis describen las propiedades de los distintos tipos de unidades de un bloque codificado de píxeles y cómo se puede reconstruir la secuencia de vídeo en el decodificador de vídeo 30, incluyendo la inter-predicción o intra-predicción, el modo de intra-predicción, los vectores de movimiento y otros parámetros. En imágenes monocromáticas o imágenes con tres planos de color independientes, una CTU puede comprender un solo bloque de codificación en árbol y elementos de sintaxis utilizados para codificar las muestras del bloque de codificación en árbol. Un bloque de codificación en árbol puede ser un bloque NxN de muestras.

60 Para lograr un mejor rendimiento, el codificador de vídeo 20 puede realizar de forma recursiva divisiones en árbol, tales como divisiones en árbol binario, divisiones en árbol ternario, divisiones en árbol cuaternario o una combinación de ambos en los bloques de codificación en árbol de la CTU y dividir la CTU en unidades de codificación (CU) más pequeñas. Como se representa en la figura 4C, la CTU 400 de 64x64 se divide primero en cuatro CU más pequeñas, cada una con un tamaño de bloque de 32x32. Entre las cuatro CU más pequeñas, la CU 410 y la CU 420 se dividen,

5 cada una, en cuatro CU de 16x16 por tamaño de bloque. Cada una de las dos CU 430 y 440 de 16x16 se dividen en cuatro CU de 8x8 por tamaño de bloque. La figura 4D representa una estructura de datos de árbol cuádruple que ilustra el resultado final del proceso de partición de la CTU 400 como se representa en la figura 4C, cada nodo hoja del árbol cuádruple corresponde a una CU de un tamaño respectivo que varía de 32x32 a 8x8. Al igual que la CTU
   
 10 que se representa en la figura 4B, cada CU puede comprender un bloque de codificación (CB) de muestras de luminancia y dos bloques de codificación correspondientes de muestras de crominancia de un fotograma del mismo tamaño, y elementos de sintaxis utilizados para codificar las muestras de los bloques de codificación. En las imágenes monocromáticas o imágenes con tres planos de color independientes, una CU puede comprender un solo bloque de codificación y estructuras de sintaxis utilizadas para codificar las muestras del bloque de codificación. Cabe señalar
   
 15 que la división en árbol cuaternario que se describe en las figuras 4C y 4D es sólo para fines ilustrativos y que una CTU puede dividirse en varias CU para adaptarse a las diversas características locales con base en las divisiones en árbol cuaternario/ternario/binario. En la estructura de árbol de múltiples tipos, una CTU se divide en una estructura de árbol cuaternario y cada CU de hoja de árbol cuaternario también puede dividirse en una estructura de árbol binario y ternario. Tal como se muestra en la figura 4E, existen cinco tipos de división, es decir, la división cuaternaria, la división binaria horizontal, la división binaria vertical, la división ternaria horizontal y la división ternaria vertical.

20 En algunas implementaciones, el codificador de vídeo 20 también puede dividir un bloque de codificación de una CU en uno o más bloques de predicción (PB) MxN. Un bloque de predicción es un bloque rectangular (cuadrado o no cuadrado) de muestras en el que se aplica la misma inter-predicción o intra-predicción. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luminancia, dos bloques de predicción correspondientes de muestras de crominancia y elementos de sintaxis utilizados para predecir los bloques de predicción. En imágenes monocromáticas o imágenes con tres planos de color independientes, una PU puede comprender un solo bloque de predicción y estructuras de sintaxis utilizadas para predecir el bloque de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luminancia, Cb y Cr para los bloques de predicción de
   
 25 luminancia, Cb y Cr de cada PU de la CU.

30 El codificador de vídeo 20 puede utilizar la intra-predicción o la inter-predicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 utiliza la intra-predicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU con base en muestras decodificadas del fotograma asociado con la PU. Si el codificador de vídeo 20 utiliza la inter-predicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU con base en muestras decodificadas de uno o más fotogramas distintos del fotograma asociado con la PU.

35 Una vez que el codificador de vídeo 20 genera los bloques predictivos de luminancia, Cb y Cr para una o más PU de una CU, el codificador de vídeo 20 puede generar un bloque residual de luminancia para la CU al restar los bloques predictivos de luminancia de la CU de su bloque de codificación de luminancia original, de tal modo que cada muestra en el bloque residual de luminancia de la CU indica una diferencia entre una muestra de luminancia en uno de los bloques predictivos de luminancia de la CU y una muestra correspondiente en el bloque de codificación de luminancia original de la CU. Del mismo modo, el codificador de vídeo 20 puede generar un bloque residual de Cb y un bloque residual de Cr, respectivamente, para la CU, de tal modo que cada muestra en el bloque residual de Cb de la CU indica una diferencia entre una muestra de Cb en uno de los bloques predictivos de Cb de la CU y una muestra correspondiente en el bloque de codificación de Cb original de la CU y cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.
   
 45

Además, tal como se ilustra en la figura 4C, el codificador de vídeo 20 puede utilizar la división en árbol cuaternario para descomponer los bloques residuales de luminancia, Cb y Cr de una CU en uno o más bloques de transformada de luminancia, Cb y Cr. Un bloque de transformada es un bloque rectangular (cuadrado o no cuadrado) de muestras en el que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede comprender un bloque de transformada de muestras de luminancia, dos bloques de transformada correspondientes de muestras de crominancia y elementos de sintaxis utilizados para transformar las muestras de bloques de transformada. Por lo tanto, cada TU de una CU puede asociarse con un bloque de transformada de luminancia, un bloque de transformada de Cb y un bloque de transformada de Cr. En algunos ejemplos, el bloque de transformada de luminancia asociado con la TU puede ser un sub-bloque del bloque residual de luminancia de la CU. El bloque de transformada de Cb puede ser
   
 50 un sub-bloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un sub-bloque del bloque residual de Cr de la CU. En las imágenes monocromáticas o imágenes con tres planos de color independientes, una TU puede comprender un solo bloque de transformada y estructuras de sintaxis utilizadas para transformar las muestras del bloque de transformada.

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

a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luminancia, un bloque de coeficientes de Cb o un bloque de coeficientes de Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación por lo general se refiere a un proceso en el que se cuantifican los coeficientes de transformada para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes de transformada, proporcionando una mayor compresión. Una vez que el codificador de vídeo 20 cuantifica un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía los elementos de sintaxis que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar la codificación aritmética binaria adaptativa según el contexto (CABAC) en los elementos de sintaxis que indican los coeficientes de transformada cuantificados. Finalmente, el codificador de vídeo 20 puede generar un flujo de bits que incluye una secuencia de bits que forma una representación de los fotogramas codificados y datos asociados, los cuales se guardan en el dispositivo de almacenamiento 32 o se transmiten al dispositivo de destino 14.

Después de recibir un flujo de bits generado por el codificador de vídeo 20, el decodificador de vídeo 30 puede analizar sintácticamente el flujo de bits para obtener elementos de sintaxis del flujo de bits. El decodificador de vídeo 30 puede reconstruir los fotogramas de los datos de vídeo con base, por lo menos en parte, en los elementos de sintaxis obtenidos del flujo de bits. El proceso de reconstruir los datos de vídeo por lo general es recíproco al proceso de codificación realizado por el codificador de vídeo 20. Por ejemplo, el decodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficiente asociados con las TU de una CU actual para reconstruir los bloques residuales asociados con las TU de la CU actual. El decodificador de vídeo 30 también reconstruye los bloques de codificación de la CU actual al añadir las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de transformada de las TU de la CU actual. Después de reconstruir los bloques de codificación para cada CU de un fotograma, el decodificador de vídeo 30 puede reconstruir el fotograma.

Tal como se señaló anteriormente, la codificación de vídeo logra la compresión de vídeo al utilizar principalmente dos modos, es decir, la intra-predicción de fotograma (o intra-predicción) y la inter-predicción de fotograma (o inter-predicción). La codificación basada en paleta constituye otro esquema de codificación que muchos estándares de codificación de vídeo han adoptado. En la codificación basada en paleta, la cual puede ser particularmente adecuada para la codificación de contenido generado en pantalla, un codificador de vídeo (por ejemplo, un codificador de vídeo 20 o un decodificador de vídeo 30) forma una tabla de paleta de colores que representa los datos de vídeo de un bloque determinado. La tabla de paleta de colores incluye los valores de píxeles más dominantes (por ejemplo, los más utilizados) en el bloque determinado. Los valores de píxeles que no se representan con frecuencia en los datos de vídeo del bloque determinado no se incluyen en la tabla de paleta de colores o se incluyen en la tabla de paleta de colores como colores de escape.

Cada entrada en la tabla de paleta de colores incluye un índice para un valor de píxel correspondiente que aparece en la tabla de paletas. Los índices de paleta para las muestras en el bloque pueden codificarse para indicar cuál entrada de la tabla de paleta de colores se va a utilizar para predecir o reconstruir qué muestra. Este modo de paleta comienza con el proceso de generar un predictor de paleta para un primer bloque de una imagen, segmento, mosaico u otra agrupación de bloques de vídeo. Tal como se explica a continuación, el predictor de paleta para los bloques de vídeo posteriores normalmente se genera al actualizar un predictor de paleta previamente utilizado. Para fines ilustrativos, se asume que el predictor de paleta se define a nivel de imagen. En otras palabras, una imagen puede incluir múltiples bloques de codificación, cada uno con su propia tabla de paleta de colores, pero existe un predictor de paleta para toda la imagen.

Para reducir los bits necesarios para señalar entradas de paleta en el flujo de bits de vídeo, un decodificador de vídeo puede utilizar un predictor de paleta para determinar nuevas entradas de paleta en la tabla de paleta de colores utilizada para reconstruir un bloque de vídeo. Por ejemplo, el predictor de paleta puede incluir entradas de paleta de una tabla de paleta previamente utilizada o incluso inicializarse con una tabla de paleta utilizada más recientemente al incluir todas las entradas de la tabla de paleta utilizada más recientemente. En algunas implementaciones, el predictor de paleta puede incluir menos de todas las entradas de la tabla de paleta utilizada más recientemente y, después, incorporar algunas entradas de otras tablas de paleta previamente utilizadas. El predictor de paleta puede tener el mismo tamaño que las tablas de paleta utilizadas para codificar diferentes bloques o puede ser más grande o más pequeño que las tablas de paleta utilizadas para codificar diferentes bloques. En un ejemplo, el predictor de paleta se implementa como una tabla (FIFO) primera en entrar, primera en salir que incluye 64 entradas de paleta.

Para generar una tabla de paleta para un bloque de datos de vídeo a partir del predictor de paleta, un decodificador de vídeo puede recibir, del flujo de bits de vídeo codificado, un indicador de un bit para cada entrada del predictor de paleta. El indicador de un bit puede tener un primer valor (por ejemplo, un valor binario) que indica que la entrada asociada del predictor de paleta se debe incluir en la tabla de paleta o un segundo valor (por ejemplo, un cero binario) que indica que la entrada asociada del predictor de paleta no se debe incluir en la tabla de paleta. Si el tamaño del predictor de paleta es mayor que la tabla de paleta utilizada para un bloque de datos de vídeo, entonces el decodificador de vídeo puede dejar de recibir más indicadores una vez que se alcance un tamaño máximo para la

tabla de paleta.

5 En algunas implementaciones, algunas entradas en una tabla de paleta pueden señalarse directamente en el flujo de bits de vídeo codificado en lugar de determinarse mediante el predictor de paleta. Para tales entradas, el decodificador de vídeo puede recibir, del flujo de bits de vídeo codificado, tres valores de m-bit separados que indican los valores de píxeles para la luminancia y dos componentes de crominancia asociados con la entrada, en donde m representa la profundidad de bit de los datos de vídeo. En comparación con los múltiples valores de m-bit necesarios para las entradas de paleta directamente señaladas, esas entradas de paleta derivadas del predictor de paleta sólo requieren un indicador de un bit. Por lo tanto, la señalización de algunas o todas las entradas de paleta mediante el predictor de paleta puede reducir significativamente el número de bits necesarios para señalar las entradas de una nueva tabla de paleta, mejorando así la eficacia de codificación general de la codificación en modo de paleta.

15 En muchos casos, el predictor de paleta para un bloque se determina con base en la tabla de paleta utilizada para codificar uno o más bloques previamente codificados. Sin embargo, al codificar la primera unidad de codificación en árbol de una imagen, un segmento o un mosaico, es posible que la tabla de paleta de un bloque previamente codificado no esté disponible. Por lo tanto, no se puede generar un predictor de paleta utilizando entradas de las tablas de paleta previamente utilizadas. En tal caso, una secuencia de inicializadores de predictor de paleta puede señalarse en un conjunto de parámetros de secuencia (SPS) y/o un conjunto de parámetros de imagen (PPS), los cuales son valores utilizados para generar un predictor de paleta cuando una tabla de paleta previamente utilizada no está disponible. Un SPS por lo general se refiere a una estructura de sintaxis de elementos de sintaxis que se aplican a una serie de imágenes de vídeo codificadas consecutivas denominada secuencia de vídeo codificada (CVS) que se determina por el contenido de un elemento de sintaxis encontrado en el PPS al que hace referencia un elemento de sintaxis encontrado en cada encabezado de fracción de segmento. Un PPS por lo general se refiere a una estructura de sintaxis de elementos de sintaxis que se aplican a una o más imágenes individuales dentro de un CVS que se determina por un elemento de sintaxis encontrado en cada encabezado de fracción de segmento. Por lo tanto, un SPS por lo general se considera como una estructura de sintaxis de nivel superior que un PPS, lo que significa que los elementos de sintaxis incluidos en el SPS por lo general cambian con menos frecuencia y se aplican a una porción más grande de datos de vídeo en comparación con los elementos de sintaxis incluidos en el PPS.

30 La figura 5 es un diagrama de bloques que ilustra ejemplos de unidades de intra-predicción de crominancia más pequeñas (SCIPU) de acuerdo con algunas implementaciones de la presente divulgación.

35 En los codificadores y decodificadores de vídeo de hardware típicos, el rendimiento de procesamiento disminuye cuando una imagen tiene un gran número de pequeños intra-bloques debido a la dependencia de datos de procesamiento de muestras entre los intra-bloques contiguos. La generación de predictor de un intra-bloque requiere muestras reconstruidas de límites superior e izquierdo de bloques contiguos. Por lo tanto, la intra-predicción debe procesarse de forma secuencial bloque por bloque.

40 En HEVC, la intra-CU más pequeña es de muestras de luminancia de 8x8. El componente de luminancia de la intra-CU más pequeña puede dividirse aún más en cuatro unidades de intra-predicción (PU) de luminancia de 4x4, pero los componentes de crominancia de la intra-CU más pequeña no se pueden dividir aún más. Por lo tanto, el peor caso de rendimiento en el procesamiento de hardware se produce cuando se procesan intra-bloques de crominancia de 4x4 o intra-bloques de luminancia de 4x4.

45 En VTM5.0, en el árbol dual, los bloques de crominancia de 2x2/4x2/2x4 ya están inhabilitados en el árbol dual. Sin embargo, en un solo árbol de codificación de VTM5, dado que las divisiones de crominancia siempre siguen a la luminancia y la intra-CU más pequeña son muestras de luminancia de 4x4, el intra-CB de crominancia más pequeño es de 2x2. Por lo tanto, en VTM5.0, los intra-CB de crominancia más pequeños en un solo árbol de codificación son de 2x2. El peor caso de rendimiento en el procesamiento de hardware para la decodificación VVC es tan sólo un 1/4 en comparación con el de la decodificación de HEVC. Además, el proceso de reconstrucción de un intra-CB de crominancia se vuelve mucho más complejo que el de HEVC después de adoptar herramientas como el modelo lineal de componentes cruzados (CCLM), los filtros de interpolación de 4 derivaciones, la combinación de intra-predicción dependiente de la posición (PDPC) y la intra-predicción e inter-predicción combinadas (CIIP). Lograr un alto rendimiento de procesamiento en decodificadores de hardware supone un reto. En esta contribución, se proponen dos métodos que mejoran el peor caso de rendimiento en el procesamiento de hardware.

55 El objetivo del esquema de SCIPU es impedir que los intra-CB de crominancia sean más pequeños que 16 muestras de crominancia al limitar la división de los intra-CB de crominancia.

60 En un solo árbol de codificación, una SCIPU se define como un nodo de codificación en árbol cuyo tamaño de bloque de crominancia es mayor o igual que 16 muestras de crominancia y tiene por lo menos un bloque de luminancia secundario más pequeño que 4\*16 muestras de luminancia. Es necesario que en cada SCIPU, todos los CB sean de tipo inter, o que todos los CB no sean de tipo inter, es decir, ya sea intra o copia intra-bloque (IBC). Se podría hacer referencia a los detalles de IBC (documento JVET-N1002 en <http://phenix.int-evry.fr/jvet/>). En el caso de una SCIPU

de tipo no inter, también se requiere que la crominancia de la SCIPU de tipo no inter ya no se divida aún más y que se permita que la luminancia de la SCIPU se divida aún más. De esta forma, el tamaño del intra-CB de crominancia más pequeño es de 16 muestras de crominancia y los CB de crominancia de 2x2, 2x4 y 4x2 se eliminan. Además, la escala de crominancia no se aplica en el caso de una SCIPU de tipo no inter.

5

Dos muestras de SCIPU se muestran en la figura 5. El CB de crominancia 502 de las muestras de crominancia de 8x4 y los CB de luminancia correspondientes (los CB de luminancia correspondientes a los CB de crominancia 502(a), 502(b) y 502(c)) forman una SCIPU porque el árbol ternario (TT) dividido a partir de las muestras de crominancia de 8x4 daría como resultado CB de crominancia más pequeños que 16 muestras de crominancia (por ejemplo, los CB de crominancia 502(a) y 502(c)). El CB de crominancia 504(a) de las muestras de crominancia de 4x4 (lado izquierdo del CB de crominancia 504 de las muestras de crominancia de 8x4) y tres CB de luminancia correspondientes a los CB de crominancia 504(c), 504(d) y 504(e) forman una SCIPU, y el CB de crominancia 504(b) de las muestras de 4x4 (lado derecho de las muestras de crominancia de 8x4) y dos CB de luminancia (por ejemplo, los CB de luminancia correspondientes a los CB de crominancia 504(g) y 504(b)) forman una SCIPU porque el árbol binario (BT) dividido a partir de las muestras de crominancia de 4x4 daría como resultado CB de crominancia más pequeños que 16 muestras de crominancia (por ejemplo, los CB de crominancia 504(g) y 504(f)).

10

15

Se infiere que una SCIPU es de tipo no inter si el segmento actual es un segmento- I o si la SCIPU actual tiene una división de luminancia de 4x4 en la misma después de una división adicional una vez (porque no se permite ninguna división de tipo inter de 4x4 en VVC); de lo contrario, el tipo de la SCIPU (inter o no inter) se indica mediante un indicador señalado antes de analizar sintácticamente las CU en la SCIPU. En la versión 6 del proyecto de especificación de VVC, la derivación de los tipos de una SCIPU se ilustra en la Tabla 1 a continuación. El modeTypeCondition 1 se define como una SCIPU con el tipo "no inter", mientras que el modeTypeCondition 2 se define como una SCIPU con el tipo "inter o no inter" y un indicador "mode\_constraint\_flag" se señala en el flujo de bits para indicar si la SCIPU asociada utiliza el tipo inter o no inter. El modeTypeCondition 0 representa condiciones en las que el nodo de división de la CU actual no es una SCIPU. La definición de todas las variables en la Tabla 2 puede ser referida a la versión 6 del proyecto de especificación de VVC (documento JVET-O2001 en <http://phenix.int-evry.fr/jvet/>). Cabe señalar que el cbWidth y el cbHeight son la anchura y la altura, respectivamente, de un bloque de codificación en las muestras de luminancia.

20

25

30

La variable `modeTypeCondition` se deriva de la siguiente manera:

- Si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 0
  - ~ `slice_type` = 1 y `qfht_dual_tree_intra_flag` es igual a 1
  - ~ `modeTypeCurr` no es igual a `MODE_ALL`.
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1
  - ~ `cbWidth * cbHeight` es igual a 64 y `split_qt_flag` es igual a 1
  - ~ `cbWidth * cbHeight` es igual a 64 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER`
  - ~ `cbWidth * cbHeight` es igual a 32 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER`
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1 + (`slice_type` != 1 ? 1 : 0)
  - ~ `cbWidth * cbHeight` es igual a 64 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER`
  - ~ `cbWidth * cbHeight` es igual a 128 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER`
- De lo contrario, `modeTypeCondition` se establece en un valor igual a 0

`mode_constraint_flag` igual a 0 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual sólo pueden utilizar modos de codificación de intra-predicción. `mode_constraint_flag` igual a 1 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual no pueden utilizar modos de codificación de inter-predicción.

TABLA 1

Al aplicar el esquema de SCIPU en la forma antes descrita, el peor caso de rendimiento en el procesamiento de hardware se produce cuando se procesan bloques de crominancia de 4x4, 2x8 u 8x2, en lugar de bloques de crominancia de 2x2. El peor caso de rendimiento en el procesamiento de hardware es el mismo que en HEVC y es 4X del de VTM5.0.

Las figuras 6A a 6C son diagramas de bloques que ilustran ejemplos de formatos de crominancia (por ejemplo, también conocidos como formatos de submuestreo de crominancia) que son compatibles con VVC de acuerdo con algunas implementaciones de la presente descripción. La Tabla 2 muestra información para diferentes formatos de crominancia. Por ejemplo, las variables `SubWidthC` y `SubHeightC` se utilizan para especificar la relación entre la anchura y la altura de un bloque de luminancia y sus bloques de crominancia asociados.

TABLA 2

<code>chroma_format_idc</code>	<code>separate_colour_plane_flag</code>	Formato de crominancia	SubWidth C	SubHeight C
0	0	Monocromo	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

En el muestreo monocromático sólo hay una matriz de muestras, la cual nominalmente se considera la matriz de luminancia.

## ES 3 019 082 T3

En el muestreo 4:2:0, cada una de las dos matrices de crominancia tiene la mitad de la altura y la mitad de la anchura que la matriz de luminancia.

5 En el muestreo 4:2:2, cada una de las dos matrices de crominancia tiene la misma altura y la mitad de la anchura que la matriz de luminancia.

10 En el muestreo 4:4:4, dependiendo del valor de `separate_color_plane_flag`, se aplica lo siguiente: si `separate_color_plane_flag` es igual a 0, cada una de las dos matrices de crominancia tiene la misma altura y anchura que la matriz de luminancia. De lo contrario (`separate_color_plane_flag` es igual a 1), los tres planos de color se procesan por separado como imágenes monocromáticas muestreadas.

15 Cuando el valor de `chroma_format_idc` es igual a 1, las ubicaciones nominales vertical y horizontal relativas de las muestras de luminancia y crominancia en las imágenes se muestran en la figura 6A. Las ubicaciones relativas alternativas de las muestras de crominancia pueden indicarse en la información de uso del vídeo.

Cuando el valor de `chroma_format_idc` es igual a 2, las muestras de crominancia se co-sitúan con las muestras de luminancia correspondientes y las ubicaciones nominales en una imagen son como se muestra en la figura 6B.

20 Cuando el valor de `chroma_format_idc` es igual a 3, todas las muestras de matriz se co-sitúan para todos los casos de imágenes y las ubicaciones nominales en una imagen son como se muestra en la figura 6C.

La figura 7 es un diagrama de bloques que ilustra ejemplos de las SCIPU en diversos formatos de crominancia de acuerdo con algunas implementaciones de la presente divulgación.

25 Cuando el concepto de SCIPU se adoptó en VVC, en el diseño sólo se consideró el formato de crominancia 4:2:0. Los demás formatos de crominancia no se consideran y, por lo tanto, el diseño impuso reglas excesivamente restrictivas al tamaño de los intra-bloques de crominancia para los formatos de crominancia distintos de 4:2:0 (por ejemplo 4:4:4 y 4:2:2). Específicamente, cuando se propuso la SCIPU, se definió como un nodo de codificación en árbol cuyo tamaño de bloque de crominancia es mayor o igual que 16 muestras de crominancia y tiene por lo menos un bloque de luminancia secundario más pequeño que las muestras de luminancia de 4x16. El uso de 4x de muestras de luminancia (por ejemplo, cuando un bloque se codifica en formato de crominancia 4:2:0, existen cuatro veces más muestras de luminancia que muestras de crominancia) para definir la SCIPU sólo funciona correctamente cuando el formato de crominancia es 4:2:0. En la versión actual del VVC, independientemente del formato de crominancia que se utilice, la restricción de tamaño de crominancia siempre se aplica a través del esquema de SCIPU, suponiendo que el formato de crominancia sea 4:2:0, lo que implica que tanto la anchura como la altura del bloque de crominancia son la mitad de la anchura y la altura del bloque de luminancia asociado.

40 Además, el modo de paleta se incorporó recientemente en la VVC durante la 15ª reunión del JVET (consultar el documento JVET-O0119 en <http://phenix.int-evry.fr/jvet/>), y cuando el modo de la SCIPU se señala como de tipo `inter`, sólo se permiten los modos `inter` dentro de esta SCIPU y los bloques de crominancia pueden entonces dividirse en bloques pequeños, tales como bloques de crominancia de 2x4 o 4x2. Aunque el tipo `no inter` se señala como el modo de la SCIPU o el `modeTypeCondition` de la SCIPU es de tipo `no inter` (por ejemplo, `modeTypeCondition = 0`), sólo se permiten los modos de intra-predicción, IBC y modos de paleta para la SCIPU y el bloque de crominancia no se puede dividir en pequeños bloques de crominancia, tales como 2x2, 2x4 o 4x2. Sin embargo, no requiere información de los píxeles reconstruidos izquierdo y superior contiguos para el modo de paleta y, por lo tanto, el modo de paleta debe considerarse como un modo que permite la división del bloque de crominancia en pequeños bloques de crominancia.

50 En algunas realizaciones, una SCIPU se define como un nodo de codificación en árbol cuyo tamaño de bloque de crominancia es mayor o igual que N muestras de crominancia y tiene por lo menos un bloque de luminancia secundario más pequeño que KxN muestras de luminancia con una división de CU adicional. N podría ser cualquier entero positivo (por ejemplo 16) y K es 4 para el formato de crominancia 4:2:0 (`chroma_format_idc=1`), K es 2 para el formato de crominancia 4:2:2, y K es 1 para el formato de crominancia 4:4:4. Cabe señalar que K=1 para el formato de crominancia 4:4:4 implica que la SCIPU no está habilitada.

55 En algunas realizaciones, la SCIPU se define como un nodo de codificación en árbol cuyo tamaño de bloque de crominancia es mayor o igual que N muestras de crominancia y tiene por lo menos un bloque de crominancia secundario más pequeño que N muestras de luminancia con una división de CU adicional. N podría ser cualquier entero positivo (por ejemplo, 16). Por ejemplo, en la figura 7, la SCIPU 702 es una SCIPU válida porque dividir el nodo de codificación en árbol de acuerdo con el método de división predefinido (por ejemplo, división vertical binaria) daría como resultado un componente de crominancia con menos de 16 muestras de crominancia (por ejemplo, un componente de crominancia con muestras de crominancia de 2x4). La SCIPU 704 es una SCIPU válida porque dividir el CB de acuerdo con el método de división predefinido (por ejemplo, división vertical ternaria) daría como resultado un componente de crominancia con menos de 16 muestras de crominancia (por ejemplo, un componente de crominancia con 2x2 muestras de crominancia). La SCIPU 706 es una SCIPU válida porque dividir el CB con el método

de división predefinido (por ejemplo, división vertical ternaria) daría como resultado un componente de crominancia con menos de 16 muestras de crominancia (por ejemplo, un componente de crominancia con muestras de crominancia de 4x2).

- 5 Se infiere que una SCIPU es de tipo no inter si el segmento actual es un segmento- I o si la SCIPU actual tiene una división de luminancia de 4x4 en la misma después de una división adicional una vez (porque no se permite ninguna de tipo inter de 4x4 en VVC); de lo contrario, el tipo de la SCIPU (inter o no inter) se indica mediante un indicador señalado antes de analizar sintácticamente las CU en la SCIPU.
- 10 Cuando el formato de crominancia es 4:4:4 (chroma\_format\_idc=3), el número de las muestras de crominancia y el número de las muestras de luminancia dentro de un bloque son iguales y, por lo tanto, no es necesario imponer restricciones de tamaño en los bloques de crominancia cuando el formato de crominancia es 4:4:4. Por ejemplo, para una SCIPU de 8x8 que utiliza la división de árbol cuaternario, ya que contiene cuatro bloques de luminancia de 4x4 divididos mediante la división de árbol cuaternario, se infiere que el tipo de una SCIPU es no inter de acuerdo con el diseño actual (ver la Tabla 1 anterior). Además, de acuerdo con el diseño actual, una SCIPU de tipo no inter no permite dividir aún más el bloque de crominancia (bloque de crominancia de 8x8 en este ejemplo). Además, es obvio que las restricciones de tamaño en el bloque de crominancia son demasiado restrictivas, ya que un intra-bloque de crominancia de 8x8 puede dividirse en cuatro intra-bloques de crominancia de 4x4 sin ocasionar un problema adicional de rendimiento. De acuerdo con la segunda realización, se propone deshabilitar el esquema de SCIPU cuando la imagen codificada se encuentra en el formato de crominancia 4:4:4. Un ejemplo de especificación se ilustra en la Tabla 3.

La variable modeTypeCondition se deriva de la siguiente manera.

- Si se cumple una de las siguientes condiciones, modeTypeCondition se establece en un valor igual a 0
  - slice\_type = 1 y qfbit\_dual\_tree\_intra\_flag es igual a 1
  - modeTypeCurr no es igual a MODE\_ALL
  - chroma\_format\_idc es igual a 3
- De lo contrario, si se cumple una de las siguientes condiciones, modeTypeCondition se establece en un valor igual a 1
  - cbWidth \* cbHeight es igual a 64 y split\_of\_flag es igual a 1
  - cbWidth \* cbHeight es igual a 64 y MttSplitMode[x0][y0][mttDepth] es igual a SPLIT\_TT\_HOR o SPLIT\_TT\_VER
  - cbWidth \* cbHeight es igual a 32 y MttSplitMode[x0][y0][mttDepth] es igual a SPLIT\_BT\_HOR o SPLIT\_BT\_VER
- De lo contrario, si se cumple una de las siguientes condiciones, modeTypeCondition se establece en un valor igual a 1 + (slice\_type != 1 ? 1 : 0)
  - cbWidth \* cbHeight es igual a 64 y MttSplitMode[x0][y0][mttDepth] es igual a SPLIT\_BT\_HOR o SPLIT\_BT\_VER
  - cbWidth \* cbHeight es igual a 128 y MttSplitMode[x0][y0][mttDepth] es igual a SPLIT\_TT\_HOR o SPLIT\_TT\_VER
- De lo contrario, modeTypeCondition se establece en un valor igual a 0

mode\_constraint\_flag igual a 0 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual sólo pueden utilizar modos de codificación de inter-predicción. mode\_constraint\_flag igual a 1 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual no pueden utilizar modos de codificación de inter-predicción.

TABLA 3: El ejemplo de la especificación del método propuesto para considerar adicionalmente el formato de crominancia 4:4:4

- 25 Cuando el formato de crominancia es 4:2:2 (chroma\_format\_idc=2), la altura de las muestras de crominancia y la altura de las muestras de luminancia dentro de un bloque son iguales, pero la anchura de las muestras de crominancia es la mitad de la anchura de las muestras de luminancia y, por lo tanto, la derivación del modeTypeCondition para la SCIPU debe modificarse en consecuencia cuando el formato de crominancia es 4:2:2. De acuerdo con la tercera realización

en esta divulgación, se propone considerar el formato de crominancia 4:2:2 cuando se aplica el esquema de SCIPU para evitar limitar en exceso el tamaño del intra-bloque de crominancia. Específicamente, la SCIPU se define como un nodo de codificación en árbol cuyo tamaño de bloque de crominancia es mayor o igual que dieciséis muestras de crominancia y tiene por lo menos un bloque de crominancia secundario más pequeño que 16 muestras de crominancia con una división adicional. En la Tabla 4 se ilustra un ejemplo de la restricción propuesta en el formato 4:2:2 con base en la especificación de VVC. En el método de derivación de modeTypeCondition modificado, se agregan otras cinco condiciones para el formato de crominancia 4:2:2 para la SCIPU de tipo no inter. Por ejemplo, un bloque de 8x8 con una división de CU de árbol cuaternario aplicada se considera como una SCIPU porque sus bloques secundarios incluyen un bloque de crominancia más pequeño que 16 muestras de crominancia (contiene cuatro bloques de luminancia de 4x4 y cuatro bloques de crominancia de 2x4 para el formato de crominancia 4:2:2). Además, dado que sus bloques secundarios incluyen bloques de luminancia de 4x4, esta SCIPU se considera como una SCIPU de tipo no inter. Del mismo modo, un bloque de 16x4 con la división horizontal TT (SPLIT\_TT\_VER) también se considera como una SCIPU de tipo no inter, ya que sus bloques secundarios incluyen bloques de luminancia de 4x4 y bloques de crominancia menores a 16 muestras de crominancia (contiene dos bloques de luminancia de 4x4 y un bloque de luminancia de 8x4, dos bloques de crominancia de 2x4 y un bloque de crominancia de 4x4). Se pueden consultar más ejemplos en la Tabla 4 a continuación.

La variable `modeTypeCondition` se deriva de la siguiente manera:

- Si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 0
  - `slice_type = 1` y `qtotl_dual_tree_intra_flag` es igual a 1
  - `modeTypeCurr` no es igual a `MIDDE_ALL`
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1
  - `cbWidth * cbHeight` es igual a 64 y `split_qt_flag` es igual a 1 y `chroma_format_idc` es igual a 1
  - `cbWidth` es igual a 8 y `cbHeight` es igual a 8 y `split_qt_flag` es igual a 1 y `chroma_format_idc` es igual a 2
  - `cbWidth * cbHeight` es igual a 64 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER` y `chroma_format_idc` es igual a 1
  - `cbWidth` es igual a 16 y `cbHeight` es igual a 4 y `MttSplitMode[x0][y0][mttDepth]` `SPLIT_TT_VER` y `chroma_format_idc` es igual a 2
  - `cbWidth` es igual a 4 y `cbHeight` es igual a 16 y `MttSplitMode[x0][y0][mttDepth]` `SPLIT_TT_HOR` y `chroma_format_idc` es igual a 2
  - `cbWidth * cbHeight` es igual a 32 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1
  - `cbWidth` es igual a 8 y `cbHeight` es igual a 4 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_VER` y `chroma_format_idc` es igual a 2
  - `cbWidth` es igual a 4 y `cbHeight` es igual a 8 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` y `chroma_format_idc` es igual a 2
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1 + (`slice_type` - 1 ? 1 : 0)
  - `cbWidth * cbHeight` es igual a 64 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1
  - `cbWidth * cbHeight` es igual a 128 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER` y `chroma_format_idc` es igual a 1
  - `cbWidth * cbHeight` es igual a 64 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1
- De lo contrario, `modeTypeCondition` se establece en un valor igual a 0

`mode_constraint_flag` igual a 0 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual sólo pueden utilizar modos de codificación de inter-predicción. `mode_constraint_flag` igual a 1 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual no pueden utilizar modos de codificación de inter-predicción.

TABLA 4: El ejemplo de la especificación del método propuesto para considerar adicionalmente el formato de crominancia 4:2:2.

- Cabe señalar que las CU con una anchura \* altura igual a 64 podrían ser de 8x8, 16x4 o 4x16 en la versión actual del VVC. La CU de 8x8 es el único tamaño que podría aplicar una división de árbol cuaternario (`split_qt_flag` es igual a 1).
- 5 No se permite la división de árbol terciario (TT) para una CU de 8x8. Además, sólo se permite `SPLIT_TT_VER` para una CU de 16x4 en la versión actual del VVC y sólo se permite `SPLIT_TT_HOR` para una CU de 4x16. De igual modo, las CU con una anchura \* altura igual a 32 podrían ser de 8x4 o 4x8 en la versión actual del VVC. Sólo se permite `SPLIT_BT_VER` para una CU de 8x4 y sólo se permite `SPLIT_BT_HOR` para una CU de 4x8. Por lo tanto, la derivación de `modeTypeCondition` podría simplificarse de la Tabla 4 a la Tabla 5 que se muestra más adelante y se observa que
- 10 los métodos de derivación que se indican en la Tabla 4 y la Tabla 5 darán como resultado el mismo `modeTypeCondition` en la versión actual del VVC.

La variable `modeTypeCondition` se deriva de la siguiente manera:

- Si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 0
  - `slice_type = 1` y `qtbft_dual_free_intra_flag` es igual a 1
  - `modeTypeCurr` no es igual a `MODE_ALL`.
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1
  - `cbWidth * cbHeight` es igual a 64 y `split_qt_flag` es igual a 1 y `chroma_format_idc` es igual a 1 o 2
  - `cbWidth * cbHeight` es igual a 64 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER` y `chroma_format_idc` es igual a 1 o 2
  - `cbWidth * cbHeight` es igual a 32 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1 o 2
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1 + (`slice_type != 1 ? 1 : 0`)
  - `cbWidth * cbHeight` es igual a 64 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1
  - `cbWidth * cbHeight` es igual a 128 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER` y `chroma_format_idc` es igual a 1
  - `cbWidth * cbHeight` es igual a 64 y `MtSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1
- De lo contrario, `modeTypeCondition` se establece en un valor igual a 0

`mode_constraint_flag` igual a 0 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual sólo pueden utilizar modos de codificación de inter-predicción. `mode_constraint_flag` igual a 1 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual no pueden utilizar modos de codificación de inter-predicción.

TABLA 5: El ejemplo de la especificación del método propuesto para considerar adicionalmente el formato de crominancia 4.2.2

5 Cuando el formato de crominancia es monocromático (`chroma_format_idc=0`), no existen muestras de crominancia dentro de un bloque y, por lo tanto, no es necesario imponer restricciones de tamaño en los bloques de crominancia cuando el formato de crominancia es monocromático. De acuerdo con la cuarta realización en esta divulgación, se propone considerar el formato de crominancia monocromático cuando el esquema de SCIPU se aplica para evitar limitar en exceso el tamaño del intra-bloque. Un ejemplo de la especificación se ilustra en la Tabla 6 a continuación.

<p>-- Si se cumple una de las siguientes condiciones, modeTypeCondition se establece en un valor igual a 0</p> <ul style="list-style-type: none"> <li>-- slice_type != 1 y qfbit_dual_tree_intra_flag es igual a 1</li> <li>-- modeTypeCurr no es igual a MODE_ALL</li> <li>-- chroma_format_idc es igual a 0</li> </ul> <p>-- De lo contrario, si se cumple una de las siguientes condiciones, modeTypeCondition se establece en un valor igual a 1</p> <ul style="list-style-type: none"> <li>-- cbWidth * cbHeight es igual a 64 y split_qt_flag es igual a 1</li> <li>-- cbWidth * cbHeight es igual a 64 y MtISplitMode[ x0 ][ y0 ][ mtDepth ] es igual a SPLIT_TT_HOR o SPLIT_TT_VER</li> <li>-- cbWidth * cbHeight es igual a 32 y MtISplitMode[ x0 ][ y0 ][ mtDepth ] es igual a SPLIT_BT_HOR o SPLIT_BT_VER</li> </ul> <p>-- De lo contrario, si se cumple una de las siguientes condiciones, modeTypeCondition se establece en un valor igual a 1 + (slice_type != 1 ? 1 : 0)</p> <ul style="list-style-type: none"> <li>-- cbWidth * cbHeight es igual a 64 y MtISplitMode[ x0 ][ y0 ][ mtDepth ] es igual a SPLIT_BT_HOR o SPLIT_BT_VER</li> <li>-- cbWidth * cbHeight es igual a 128 y MtISplitMode[ x0 ][ y0 ][ mtDepth ] es igual a SPLIT_TT_HOR o SPLIT_TT_VER</li> </ul> <p>-- De lo contrario, modeTypeCondition se establece en un valor igual a 0</p> <p>mode_constraint_flag igual a 0 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual sólo pueden utilizar modos de codificación de inter-predicción. mode_constraint_flag igual a 1 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual no pueden utilizar modos de codificación de inter-predicción.</p>
--

TABLA 8: El ejemplo de la especificación del método propuesto para considerar adicionalmente el formato de crominancia monocromático

5 Para considerar el formato de crominancia monocromo, 4:2:0, 4:4:4 y 4:2:2 todos juntos para la derivación del modeTypeCondition, se proporciona una especificación de ejemplo de acuerdo con la presente invención en la Tabla 7 a continuación.

La variable `modeTypeCondition` se deriva de la siguiente manera:

- Si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 0
  - `slice_type == 1` y `qtbt_dual_tree_intra_flag` es igual a 1
  - `modeTypeCur` no es igual a `MODE_ALL`
  - `chroma_format_idc` es igual a 0
  - `chroma_format_idc` es igual a 3
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1
  - `cbWidth * cbHeight` es igual a 64 y `split_ct_flag` es igual a 1 y `chroma_format_idc` es igual a 1 o 2
  - `cbWidth * cbHeight` es igual a 64 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER` y `chroma_format_idc` es igual a 1 o 2
  - `cbWidth * cbHeight` es igual a 32 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1 o 2
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1 + (`slice_type != 1 ? 1 : 0`)
  - `cbWidth * cbHeight` es igual a 64 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1
  - `cbWidth * cbHeight` es igual a 128 y `MttSplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER` y `chroma_format_idc` es igual a 1
- De lo contrario, `modeTypeCondition` se establece en un valor igual a 0

`mode_constraint_flag` igual a 0 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual sólo pueden utilizar modos de codificación de inter-predicción. `mode_constraint_flag` igual a 1 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual no pueden utilizar modos de codificación de inter-predicción.

TABLA 7: El ejemplo de la especificación del método propuesto para considerar el formato de crominancia monocromático, 4.2.0, 4.4.4 y 4.2.2

Se observa que la Tabla 7 podría simplificarse aún más en la Tabla 8 que se muestra a continuación sin cambiar la función de la derivación de `modeTypeCondition`.

La variable `modeTypeCondition` se deriva de la siguiente manera:

- Si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 0
  - ... `slice_type == 1` y `qtbtt_dual_intra_flag` es igual a 1
  - ... `modeTypeCur` no es igual a `MODE_ALL`
  - ... `chroma_format_idc` es igual a 0
  - ... `chroma_format_idc` es igual a 3
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1
  - ... `cbWidth * cbHeight` es igual a 64 y `split_qt_flag` es igual a 1
  - ... `cbWidth * cbHeight` es igual a 64 y `MtISplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER`
  - ... `cbWidth * cbHeight` es igual a 32 y `MtISplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER`
- De lo contrario, si se cumple una de las siguientes condiciones, `modeTypeCondition` se establece en un valor igual a 1 + (`slice_type != 1 ? 1 : 0`)
  - ... `cbWidth * cbHeight` es igual a 64 y `MtISplitMode[x0][y0][mttDepth]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER` y `chroma_format_idc` es igual a 1
  - ... `cbWidth * cbHeight` es igual a 128 y `MtISplitMode[x0][y0][mttDepth]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER` y `chroma_format_idc` es igual a 1
- De lo contrario, `modeTypeCondition` se establece en un valor igual a 0

`mode_constraint_flag` igual a 0 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual sólo pueden utilizar modos de codificación de inter-predicción, `mode_constraint_flag` igual a 1 especifica que las unidades de codificación dentro del nodo de árbol de codificación actual no pueden utilizar modos de codificación de inter-predicción.

0 TABLA 8: El ejemplo de la especificación del método propuesto para considerar el formato de crominancia monocromático, 4:2:0, 4:4:4 and 4:2:2

5 En algunas realizaciones, se propone que cuando el tipo no inter se señale como el modo de la SCIPU o el `modeTypeCondition` de la SCIPU sea de tipo no inter (por ejemplo, `modeTypeCondition = 0`), sólo los modos de intra-predicción, IBC se permitan para las CU dentro de la SCIPU y el bloque de crominancia no pueda dividirse en pequeños bloques de crominancia, tales como de 2x2, 2x4 o 4x2. De lo contrario, cuando se señale el tipo inter como el modo de la SCIPU, los modos tipo inter y los modos de paleta se permitan para las CU dentro de la SCIPU y el bloque de crominancia pueda dividirse en pequeños bloques de crominancia, tales como de 2x2, 2x4 o 4x2. Se observa que la denominación de los modos podría renombrarse de “no inter” a “intra+ibc” y de “inter” a “inter+plt” en consecuencia para evitar confusiones.

15 La figura 8 es un diagrama de flujo que ilustra un ejemplo de proceso 800 mediante el cual un decodificador de vídeo 30 implementa las técnicas de decodificar una unidad de intra-predicción de crominancia más pequeña (SCIPU) de acuerdo con algunas implementaciones del presente divulgación.

20 Como primera etapa, el decodificador de vídeo 30 recibe, de un flujo de bits, datos que codifican un nodo de codificación en árbol, en donde el nodo de codificación en árbol se asocia con un método de división predefinido (por ejemplo, división binaria horizontal/vertical, división ternaria horizontal/vertical, etc.) (810). Consultar la figura 4E y la descripción relacionada para ver los ejemplos de métodos de división predefinidos.

25 Después, el decodificador de vídeo 30 determina un tipo de modo del nodo de codificación en árbol (por ejemplo, establecer el indicador `modeTypeCondition` asociado con el nodo de codificación en árbol) (820) de la siguiente manera: de acuerdo con una determinación de que el nodo de codificación en árbol está codificado en un formato de submuestreo de crominancia 4:4:4 o 4:0:0, el decodificador de vídeo 30 establece el nodo de codificación en árbol en un primer tipo de modo (por ejemplo, `modeTypeCondition == 0`, lo que indica que el nodo de codificación en árbol no es una SCIPU) (830); además, de acuerdo con una determinación de que el nodo de codificación en árbol está codificado en un formato de submuestreo de crominancia 4:2:2 o 4:2:0 y el nodo de codificación en árbol es una unidad de intra-predicción de crominancia más pequeña (SCIPU) (840): el decodificador de vídeo 30 establece el nodo de

codificación en árbol en un segundo tipo de modo (por ejemplo, `modeTypeCondition == 1`) si un bloque de luminancia del nodo de codificación en árbol y el método de división cumplen con un primer criterio (por ejemplo, “`modeTypeCondition == 1`” significa que dividir un bloque de luminancia del nodo de codificación en árbol con el método de división daría como resultado un bloque de luminancia secundario con un tamaño de bloque igual a 16 muestras de luminancia; por lo tanto, el modo de predicción se infiere como “intra” y no se señala ningún elemento de sintaxis del modo de predicción) (840-1); y el decodificador de vídeo 30 establece el nodo de codificación en árbol en un tercer tipo de modo (por ejemplo, `modeTypeCondition == 2`) si el bloque de luminancia del nodo de codificación en árbol y el método de división cumplen con un segundo criterio (por ejemplo, “`modeTypeCondition == 2`” significa que dividir el bloque de luminancia del nodo de codificación en árbol con el método de división no daría como resultado el bloque de luminancia secundario con el tamaño de bloque igual a 16 muestras de luminancia y se señala el elemento de sintaxis del modo de predicción) (840-2).

Después de configurar el tipo de modo, el decodificador de vídeo 30 decodifica la unidad de codificación en árbol con base en el tipo de modo (850) de la siguiente manera: de acuerdo con una determinación de que el nodo de codificación en árbol es el segundo modo (por ejemplo, modo de tipo no inter para la SCIPU, `modeTypeCondition == 1`): el decodificador de vídeo 30 decodifica cada bloque de codificación en el nodo de codificación en árbol mediante el modo de intra-predicción (860); además, de acuerdo con una determinación de que el nodo de codificación en árbol es el primer modo (por ejemplo, no es una SCIPU) o el tercer modo (por ejemplo, de tipo no inter para la SCIPU) (870): el decodificador de vídeo 30 recibe, del flujo de bits, un elemento de sintaxis (por ejemplo, `mode_constraint_flag`) que indica un modo de predicción asociado con el nodo de codificación en árbol (880).

Después de recibir el elemento de sintaxis, de acuerdo con una determinación de que el elemento de sintaxis tiene un primer valor (por ejemplo, `mode_constraint_flag == 0`): el decodificador de vídeo 30 decodifica cada bloque de codificación en el nodo de codificación en árbol mediante el modo de inter-predicción (880-1); además, de acuerdo con una determinación de que el elemento de sintaxis tiene un segundo valor (por ejemplo, `mode_constraint_flag == 1`): el decodificador de vídeo 30 decodifica cada bloque de codificación en el nodo de codificación en árbol mediante el modo de intra-predicción (880-2).

Los bloques de crominancia del nodo de codificación en árbol pueden dividirse aún más cuando el nodo de codificación en árbol es el primer tipo de modo.

Los bloques de crominancia del nodo de codificación en árbol no se pueden dividir aún más cuando el nodo de codificación en árbol es el segundo tipo de modo.

Los bloques de crominancia del nodo de codificación en árbol no se pueden dividir aún más cuando el nodo de codificación en árbol es el tercer tipo de modo y un modo de predicción indica que el nodo de codificación en árbol actual es el modo intra.

Los bloques de crominancia del nodo de codificación en árbol pueden dividirse aún más cuando el nodo de codificación en árbol es el tercer tipo de modo y un modo de predicción indica que el nodo de codificación en árbol actual NO es un modo intra.

El nodo de codificación en árbol es la SCIPU si un bloque de crominancia del nodo de codificación en árbol tiene un bloque de crominancia y dividir el bloque de crominancia con el método de división daría como resultado que un bloque de crominancia secundario infrinja una limitación de tamaño mínimo del intra-bloque de crominancia (por ejemplo, con el tamaño del bloque de crominancia secundario más pequeño que 16 muestras de crominancia).

El bloque de luminancia del nodo de codificación en árbol y el método de división cumplen con el primer criterio si dividir el bloque de luminancia del nodo de codificación en árbol con el método de división daría como resultado un bloque de luminancia secundario con un tamaño de bloque igual a 16 muestras de luminancia.

El bloque de luminancia del nodo de codificación en árbol y el método de división cumplen con el primer criterio si: el tamaño del bloque de luminancia es de 64 muestras de luminancia, y el método de división es una división cuaternaria (por ejemplo, como se muestra en la Tabla 9:  $cbWidth * cbHeight$  es igual a 64 y `split_qt_flag` es igual a 1 y `chroma_format_idc` es igual a 1 o 2); el tamaño del bloque de luminancia es de 64 muestras de luminancia y el método de división es una división ternaria horizontal o vertical (por ejemplo, como se muestra en la Tabla 9:  $cbWidth * cbHeight$  es igual a 64 y `MttSplitMode[ x0 ][ y0 ][ mttDepth ]` es igual a `SPLIT_TT_HOR` o `SPLIT_TT_VER`); o el tamaño del bloque de luminancia es de 32 muestras de luminancia y el método de división es una división binaria horizontal o vertical (por ejemplo, como se muestra en la Tabla 9:  $cbWidth * cbHeight$  es igual a 32 y `MttSplitMode[ x0 ][ y0 ][ mttDepth ]` es igual a `SPLIT_BT_HOR` o `SPLIT_BT_VER`).

El bloque de luminancia del nodo de codificación en árbol y el método de división cumplen con el segundo criterio si el nodo de codificación en árbol no está en un Segmento-I y dividir el bloque de luminancia del nodo de codificación en árbol con el método de división no daría como resultado un bloque de luminancia secundario con un tamaño de

bloque igual a 16 muestras de luminancia (por ejemplo, el bloque de luminancia secundario resultante será más grande que 16 muestras de luminancia).

5 El bloque de luminancia del nodo de codificación en árbol y el método de división cumplen con el segundo criterio si el nodo de codificación en árbol no es de un Segmento-I (por ejemplo, en la tabla 9: slice\_type != i) y: el tamaño del bloque de luminancia es de 64 muestras de luminancia y el método de división es una división binaria horizontal o vertical (por ejemplo, en la tabla 9: cbWidth \* cbHeight es igual a 64 y MttSplitMode[ x0 ][ y0 ][ mttDepth ] es igual a SPLIT\_BT\_HOR o SPLIT\_BT\_VER y chroma\_format\_idc es igual a 1); o el tamaño del bloque de luminancia es de 128 muestras de luminancia y el método de división es una división ternaria horizontal o vertical (por ejemplo, en la 10 tabla 9: cbWidth \* cbHeight es igual a 128 y MttSplitMode[ x0 ][ y0 ][ mttDepth ] es igual a SPLIT\_TT\_HOR o SPLIT\_TT\_VER y chroma\_format\_idc es igual a 1).

15 De acuerdo con una determinación de que el nodo de codificación en árbol está codificado en un formato de submuestreo de crominancia 4:2:2 o 4:2:0 y no es la SCIPU (por ejemplo, dividir un bloque de crominancia no daría como resultado una infracción de la limitación de tamaño mínimo del intra-bloque de crominancia), el decodificador de vídeo 30 establece el nodo de codificación en árbol en el primer modo (por ejemplo, la división no daría como resultado un bloque de crominancia secundario más pequeño que 16 muestras de crominancia).

20 En uno o más ejemplos, las funciones descritas pueden implementarse en el hardware, software, firmware o en cualquier combinación de los mismos. Si se implementan en el software, las funciones pueden almacenarse o transmitirse en un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse a través de una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, los cuales corresponden a un medio tangible, como los medios de 25 almacenamiento de datos, o medios de comunicación, incluido cualquier medio que facilite la transferencia de un programa de cómputo de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador por lo general pueden corresponder a (1) medios de almacenamiento tangibles y legibles por ordenador que no sean 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 pueda acceder uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para 30 poner en práctica las implementaciones que se describen en la presente solicitud. Un producto de programa informático puede incluir un medio legible por ordenador.

35 La terminología utilizada en la descripción de las implementaciones en el presente documento tiene el propósito de describir implementaciones particulares únicamente y no pretende limitar el alcance de las reivindicaciones. Tal como se utiliza en la descripción de las implementaciones y en las reivindicaciones anexas, las formas en singular "un", "uno", "una," y "el/la" tienen por objeto incluir también las formas en plural, a menos que el contexto indique claramente lo contrario. También se entenderá que el término "y/o", en la forma utilizada en el presente documento, se refiere y abarca todas las posibles combinaciones de uno o más de los elementos enumerados asociados. Se entenderá, además, que los términos "comprende" y/o "que comprende", cuando se utilizan en esta especificación, especifican la 40 presencia de atributos, elementos y/o componentes señalados, pero no excluyen la presencia o la adición de uno o más atributos, elementos, componentes y/o grupos de los mismos.

También se entenderá que, aunque los términos primero, segundo, etc., pueden utilizarse en el presente documento para describir distintos elementos, esos elementos no deben limitarse por tales términos. Estos términos sólo se 45 utilizan para hacer la distinción entre un elemento y otro. Por ejemplo, un primer electrodo podría denominarse un segundo electrodo y, de manera similar, un segundo electrodo podría denominarse un primer electrodo, sin apartarse del alcance de las implementaciones. Tanto el primer electrodo como el segundo electrodo son electrodos, pero no son el mismo electrodo. La descripción de la presente solicitud se presenta para fines ilustrativos y descriptivos, y no tiene por objeto ser exhaustiva ni limitarse a la invención en la forma en que se describe. Diversas modificaciones, 50 variaciones e implementaciones alternativas serán aparentes para los expertos en la técnica que tienen el beneficio de las enseñanzas presentadas en las descripciones anteriores y en los dibujos correspondientes. La realización se eligió y describió con el fin de explicar de la menor manera los principios de la invención y su aplicación práctica, así como para permitir que otros expertos en la técnica entiendan la invención para distintas implementaciones y para que utilicen de la mejor manera los principios subyacentes y las diversas modificaciones en la forma que sea adecuada para el uso específico contemplado. Por lo tanto, debe entenderse que el alcance de las reivindicaciones no debe 55 limitarse a los ejemplos específicos de las implementaciones descritas y que se tiene por objeto incluir las modificaciones y otras implementaciones que se encuentren dentro del alcance de la reivindicaciones anexas.

**REIVINDICACIONES**

1. Un método para la decodificación de vídeo, que comprende:
  - 5 recibir, desde un flujo de bits, una unidad de codificación asociada con un método de división predefinido;
    - determinar un tipo de modo de la unidad de codificación, que incluye:
      - 10 de acuerdo con una determinación de que la unidad de codificación está codificada en formato de crominancia 4:2:2 o 4:2:0 y la unidad de codificación es una unidad de intrapredicción de crominancia más pequeña (SCIPU), en donde la unidad de codificación es una SCIPU si la unidad de codificación tiene un bloque de crominancia con un tamaño de bloque de crominancia que es mayor o igual que N muestras de crominancia y dividir el bloque de crominancia con el método de división da como resultado un bloque de crominancia hijo menor que N muestras de crominancia, en donde N es un número entero positivo:
        - 15 establecer la condición de tipo de modo de la unidad de codificación a una condición de segundo tipo de modo si un bloque de luminancia de la unidad de codificación y el método de división cumplen con un primer criterio; y
        - 20 establecer el tipo de modo de la unidad de codificación en un tercer tipo de modo si el bloque de luminancia de la unidad de codificación y el método de división satisfacen un segundo criterio;
          - de acuerdo con una determinación de que la unidad de codificación está codificada en formato de crominancia 4:2:2 o 4:2:0 y el bloque de luminancia de la unidad de codificación y el método de división no satisfacen el primer criterio y el segundo criterio, establecer el tipo de modo de la unidad de codificación en un primer tipo de modo;
            - 30 de acuerdo con una determinación de que la unidad de codificación está codificada en formato de crominancia 4:4:4 o 4:0:0, establecer el tipo de modo de la unidad de codificación al primer tipo de modo, en donde una restricción de tamaño de crominancia SCIPU no se aplica a la unidad de codificación cuando el tipo de modo de la unidad de codificación es el primer tipo de modo, en donde la restricción de tamaño de crominancia SCIPU es no permitir uno o más bloques de codificación de croma (CB) de la unidad de codificación más pequeños que 16 muestras de croma al restringir la división adicional de un CB de croma; y
      - 35 decodificar la unidad de codificación en función del tipo de modo de la unidad de codificación, que incluye:
        - de acuerdo con una determinación de que el tipo de modo de la unidad de codificación es el segundo tipo de modo:
          - 40 decodificar cada bloque de codificación en la unidad de codificación usando el modo intra,
            - de acuerdo con una determinación de que el tipo de modo de la unidad de codificación es el tercer tipo de modo:
              - 45 recibir, desde el flujo de bits, un elemento sintáctico que indica un modo de predicción asociado con la unidad de codificación:
                - de acuerdo con una determinación de que el elemento sintáctico tiene un primer valor: decodificar cada bloque de codificación en la unidad de codificación usando un inter modo; y
                - 50 de acuerdo con una determinación de que el elemento de sintaxis tiene un segundo valor: decodificar cada bloque de codificación en la unidad de codificación usando el modo intra,
        - 55 en donde el bloque de luminancia de la unidad de codificación y el método de división satisfacen el segundo criterio, en respuesta a que la unidad de codificación no está en una sección I y uno de los siguientes:
          - el tamaño del bloque de luminancia es 64 y el método de división es una división binaria horizontal o vertical, y la unidad de codificación está codificada en formato de crominancia 4:2:0; o
          - 60 el tamaño del bloque de luminancia es 128 y el método de división es una división ternaria horizontal o vertical, y la unidad de codificación se codifica en formato de crominancia 4:2:0,
        - en donde el bloque de luminancia de la unidad de codificación y el método de división satisfacen el primer criterio si:

## ES 3 019 082 T3

- el tamaño del bloque de luminancia es de 64 muestras de luminancia y el método de división es cuádruple;
- 5 el tamaño del bloque de luminancia es de 64 muestras de luminancia y el método de división es una división ternaria horizontal o vertical; o
- el tamaño del bloque de luminancia es de 32 muestras de luminancia y el método de división es una división binaria horizontal o vertical.
- 10 2. El método de la reivindicación 1, en donde no se permite que los bloques de crominancia de la unidad de codificación se dividan adicionalmente cuando el tipo de modo de la unidad de codificación es el segundo tipo de modo.
- 15 3. El método de la reivindicación 1, en donde no se permite que los bloques de crominancia de la unidad de codificación se dividan adicionalmente cuando el tipo de modo de la unidad de codificación es el tercer tipo de modo y el modo de predicción indica que la unidad de codificación actual está en el modo intra.
4. El método de la reivindicación 1, en donde los bloques de crominancia de la unidad de codificación pueden dividirse adicionalmente cuando la unidad de codificación es el tercer tipo de modo y el modo de predicción indica que la unidad de codificación actual no está en el modo intra.
- 20 5. El método de la reivindicación 1, en donde el bloque de luminancia de la unidad de codificación y el método de división satisfacen el primer criterio si la división del bloque de luminancia de la unidad de codificación con el método de división debe dar como resultado un bloque de luminancia secundario con un tamaño de bloque igual a 16 muestras de luminancia.
- 25 6. El método de la reivindicación 1, en donde el bloque de luminancia de la unidad de codificación y el método de división satisfacen el segundo criterio si la unidad de codificación no está en una sección I y dividir el bloque de luminancia de la unidad de codificación con el método de división no debe dar como resultado un bloque de luminancia hijo con un tamaño de bloque igual a 16 muestras de luminancia.
- 30 7. El método de la reivindicación 1, que comprende además:
- 35 de acuerdo con una determinación de que la unidad de codificación está codificada en formato de crominancia 4:2:2 o 4:2:0 y no es la SCIPU, establecer el tipo de modo de la unidad de codificación en el primer tipo de modo.
8. Un aparato electrónico que comprende:
- 40 una o más unidades de procesamiento;
- una memoria acoplada a la una o más unidades de procesamiento; y
- 45 una pluralidad de programas almacenados en la memoria que, cuando son ejecutados por la una o más unidades de procesamiento, hacen que el aparato electrónico realice el método de una cualquiera de las reivindicaciones 1-7.
9. Un medio de almacenamiento legible por ordenador no transitorio que almacena un flujo de bits para su procesamiento por un aparato electrónico que tiene una o más unidades de procesamiento, donde el flujo de bits es decodificable por el aparato electrónico con el método de una cualquiera de las reivindicaciones 1-7.
- 50 10. Un producto de programa informático que comprende un programa informático, en donde el programa informático, cuando es ejecutado por un procesador, implementa el método de una cualquiera de las reivindicaciones 1-7.
- 55

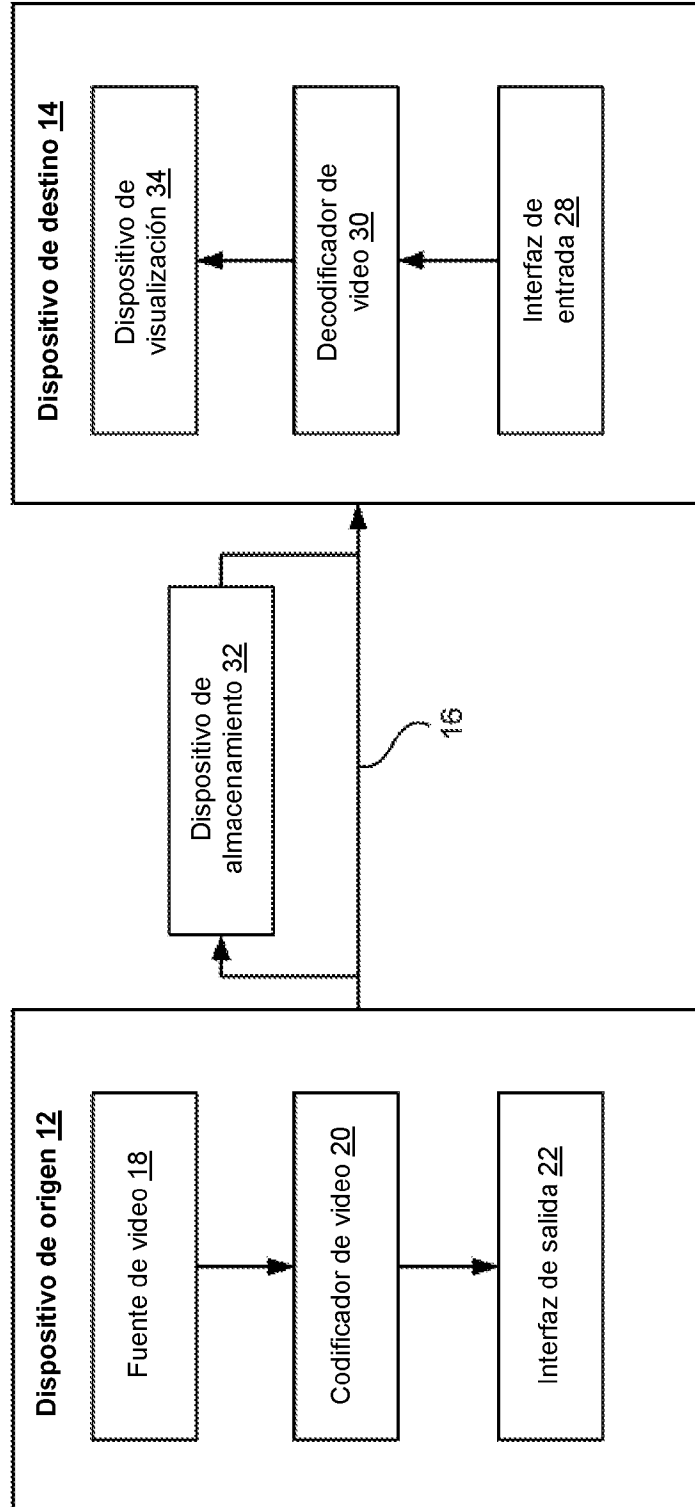


FIG. 1

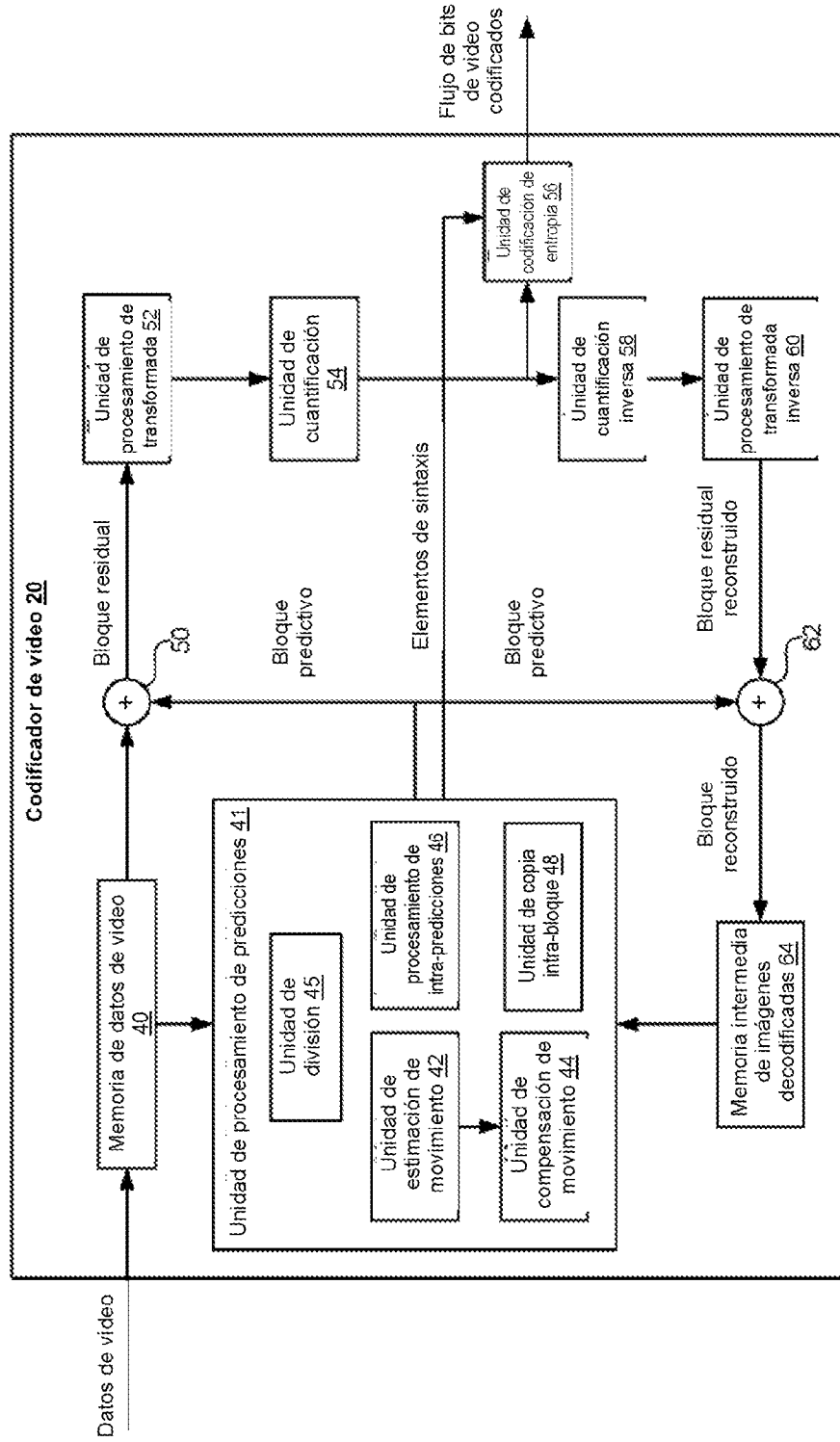


FIG. 2

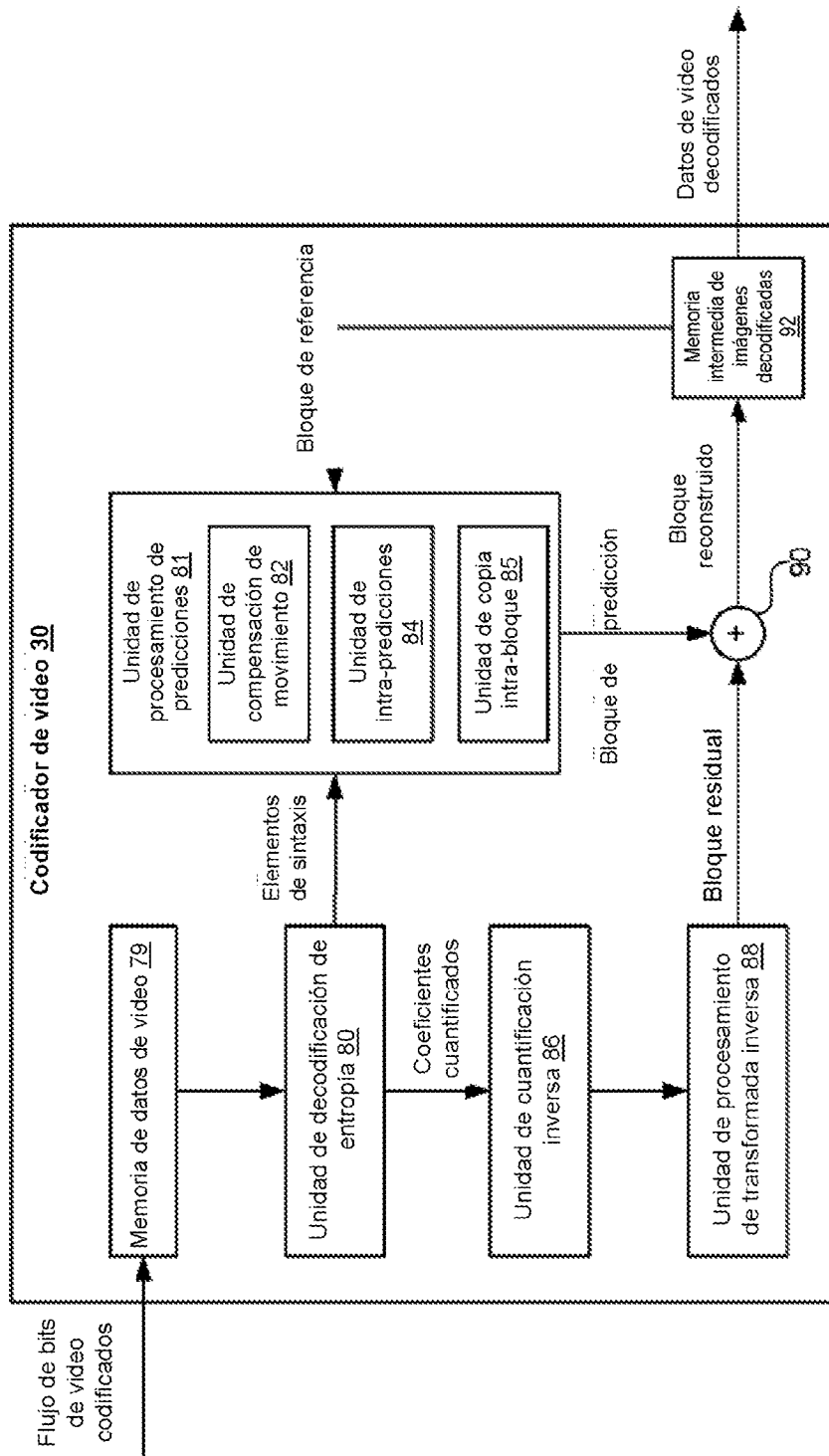


FIG. 3

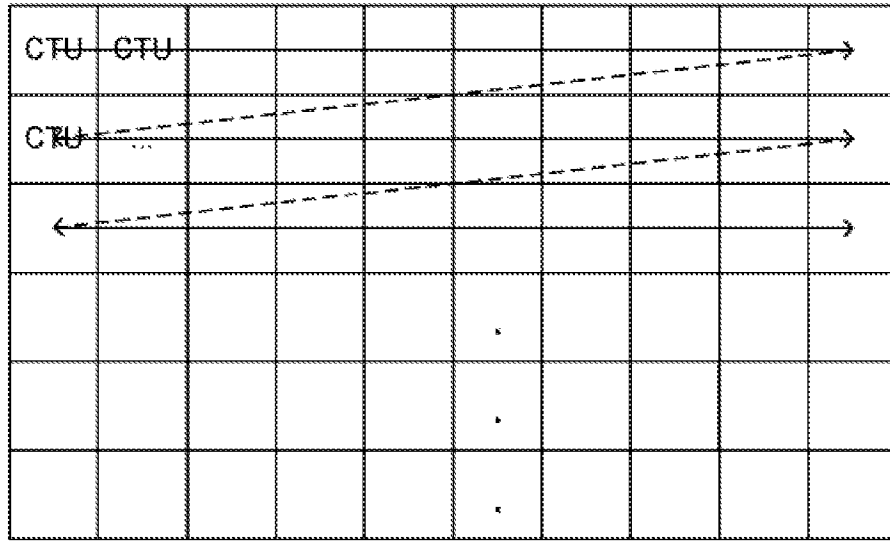


FIG. 4A

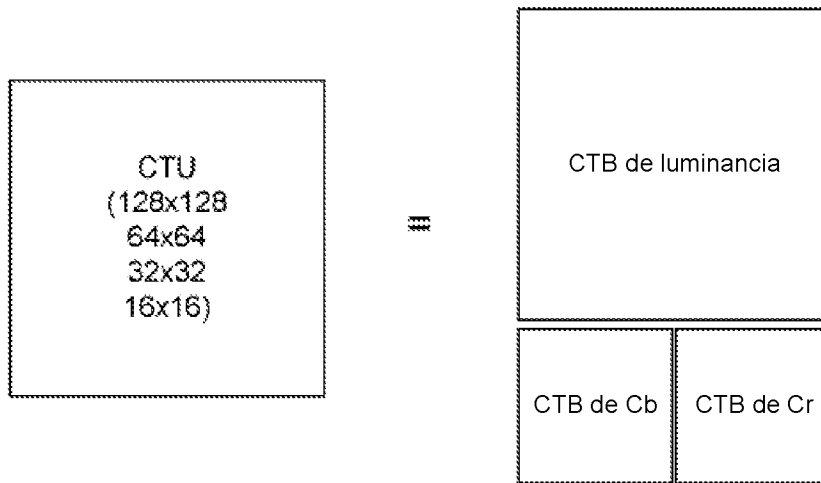


FIG. 4B

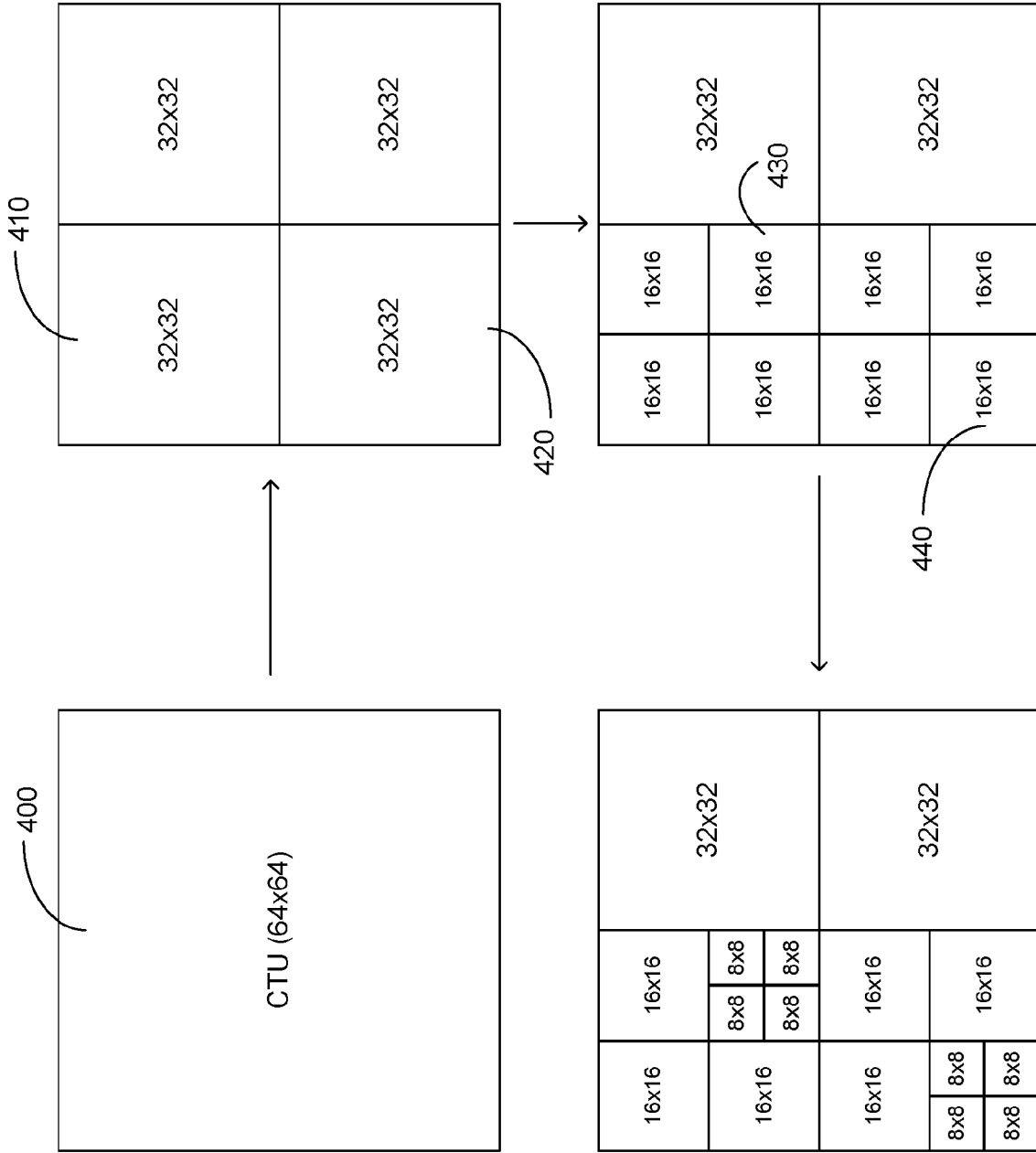


FIG. 4C

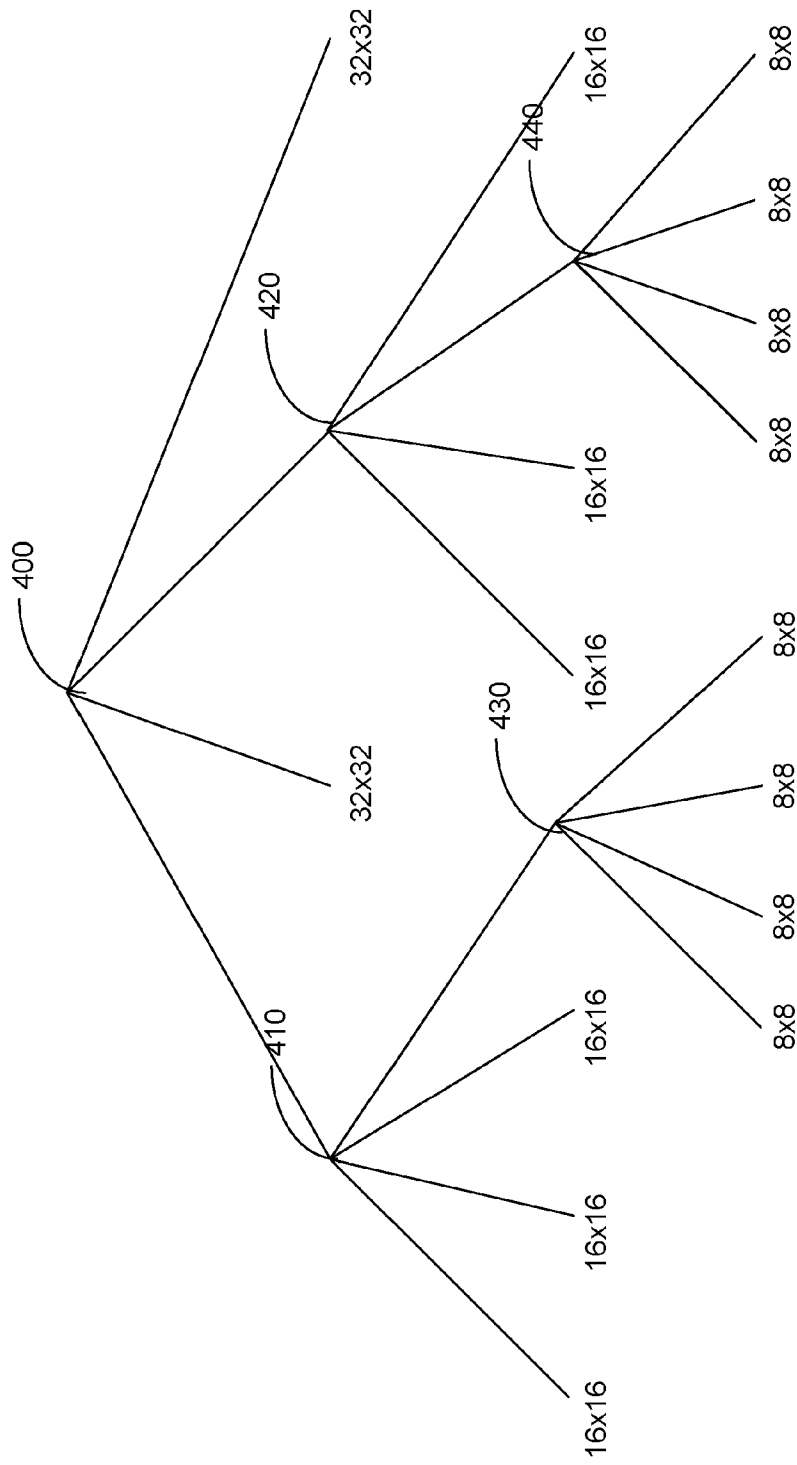


FIG. 4D

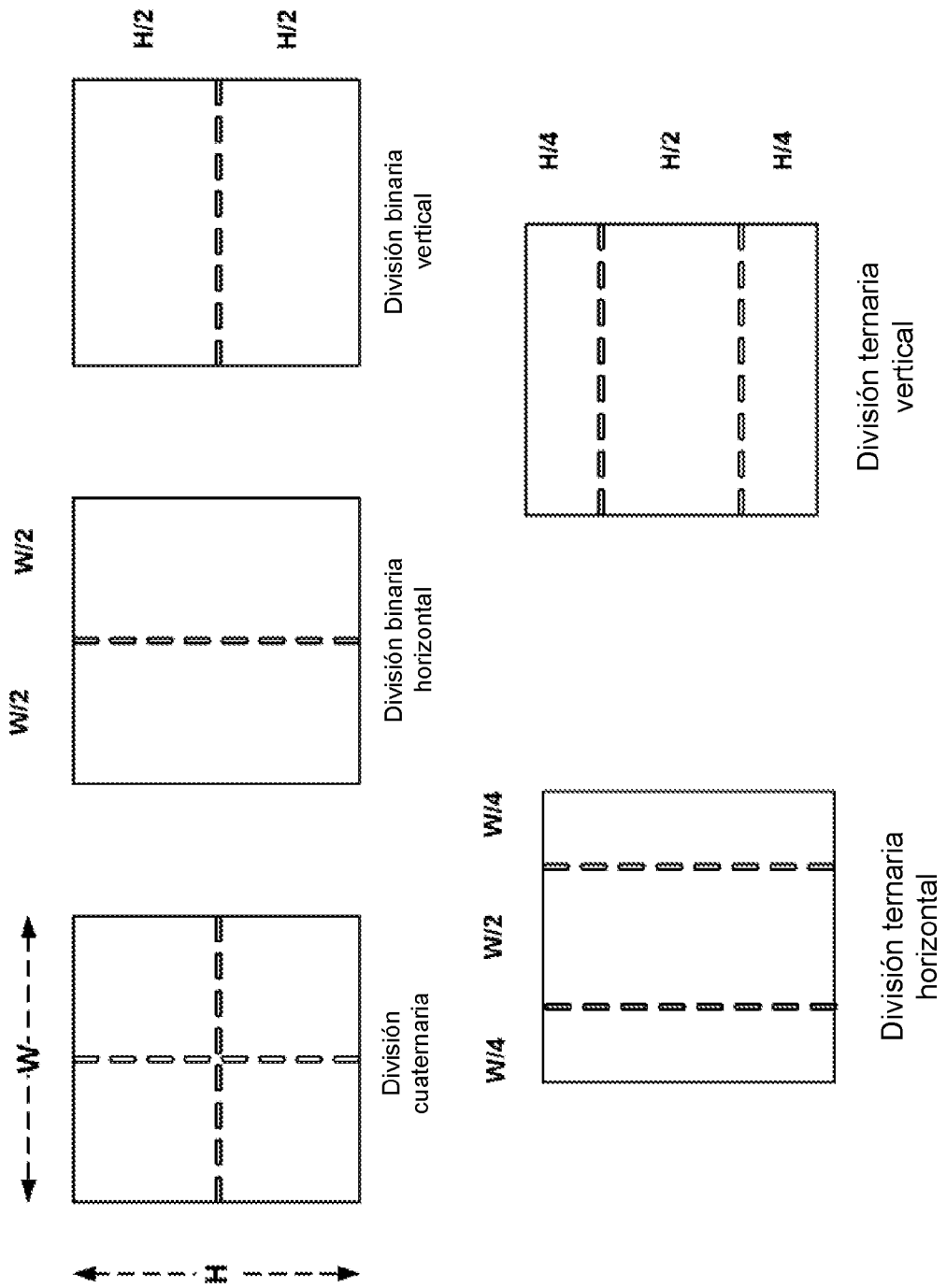
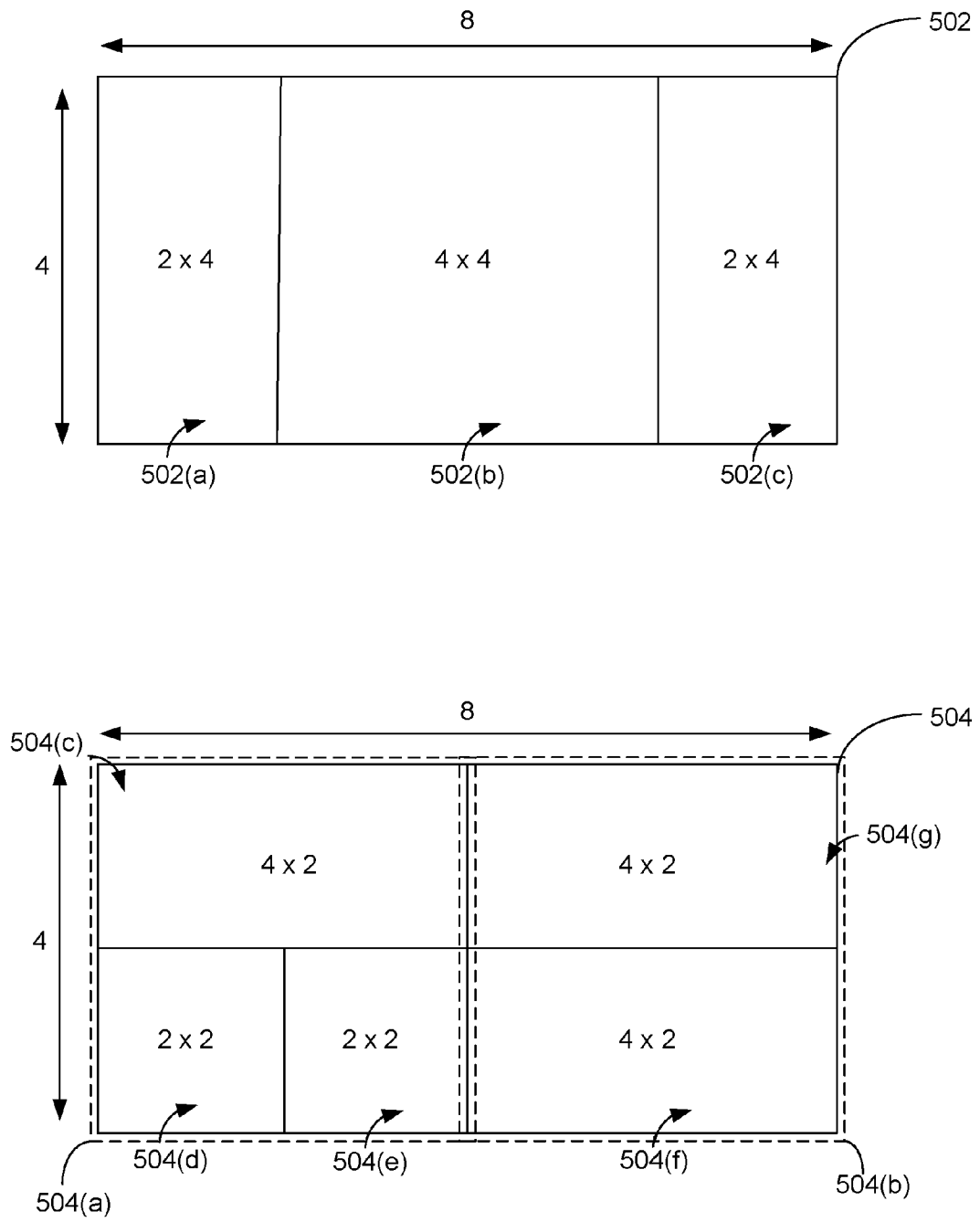


FIG. 4E



**FIG. 5**

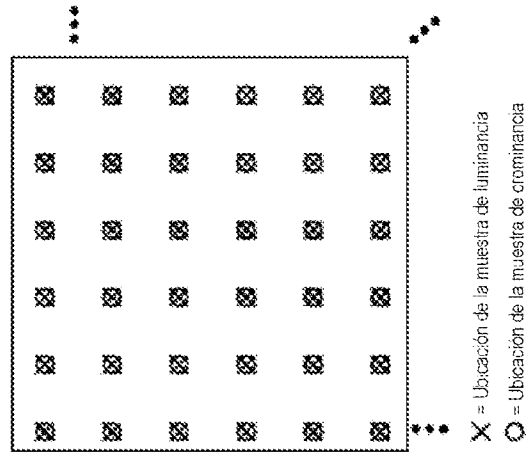


FIG. 6A

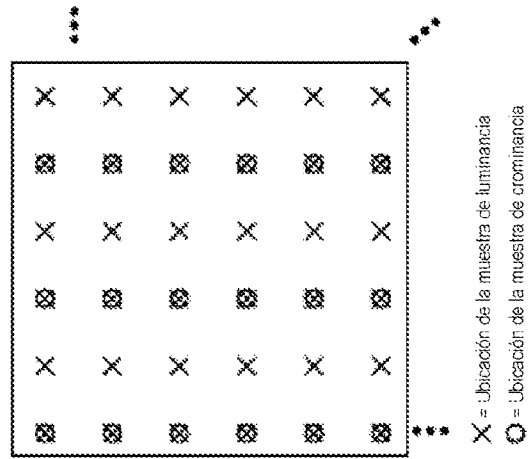


FIG. 6B

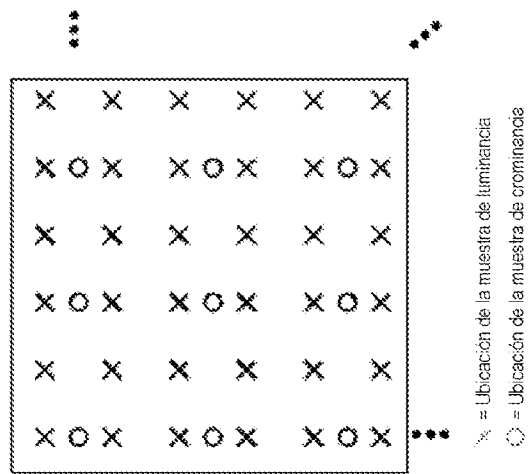


FIG. 6C

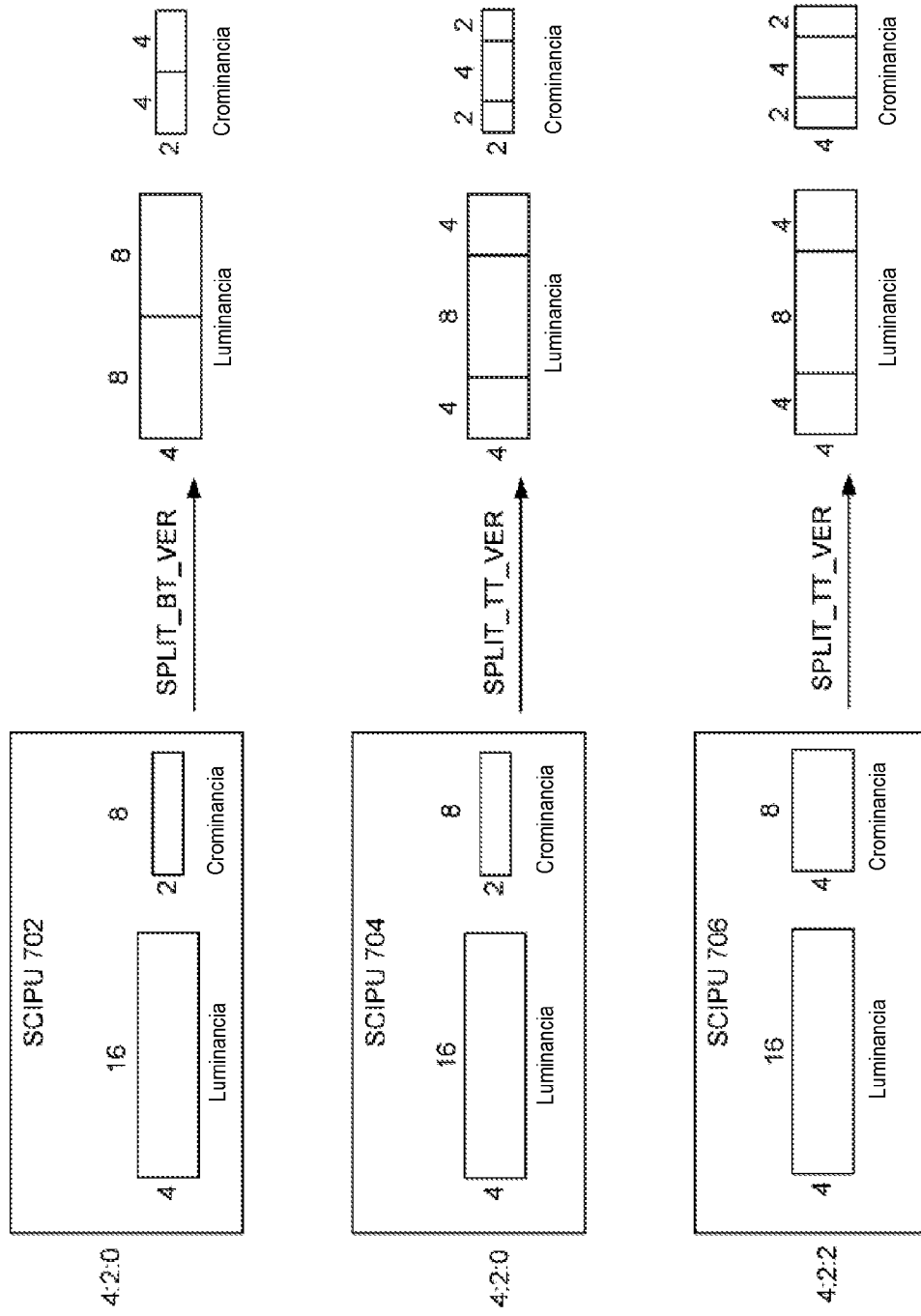


FIG. 7

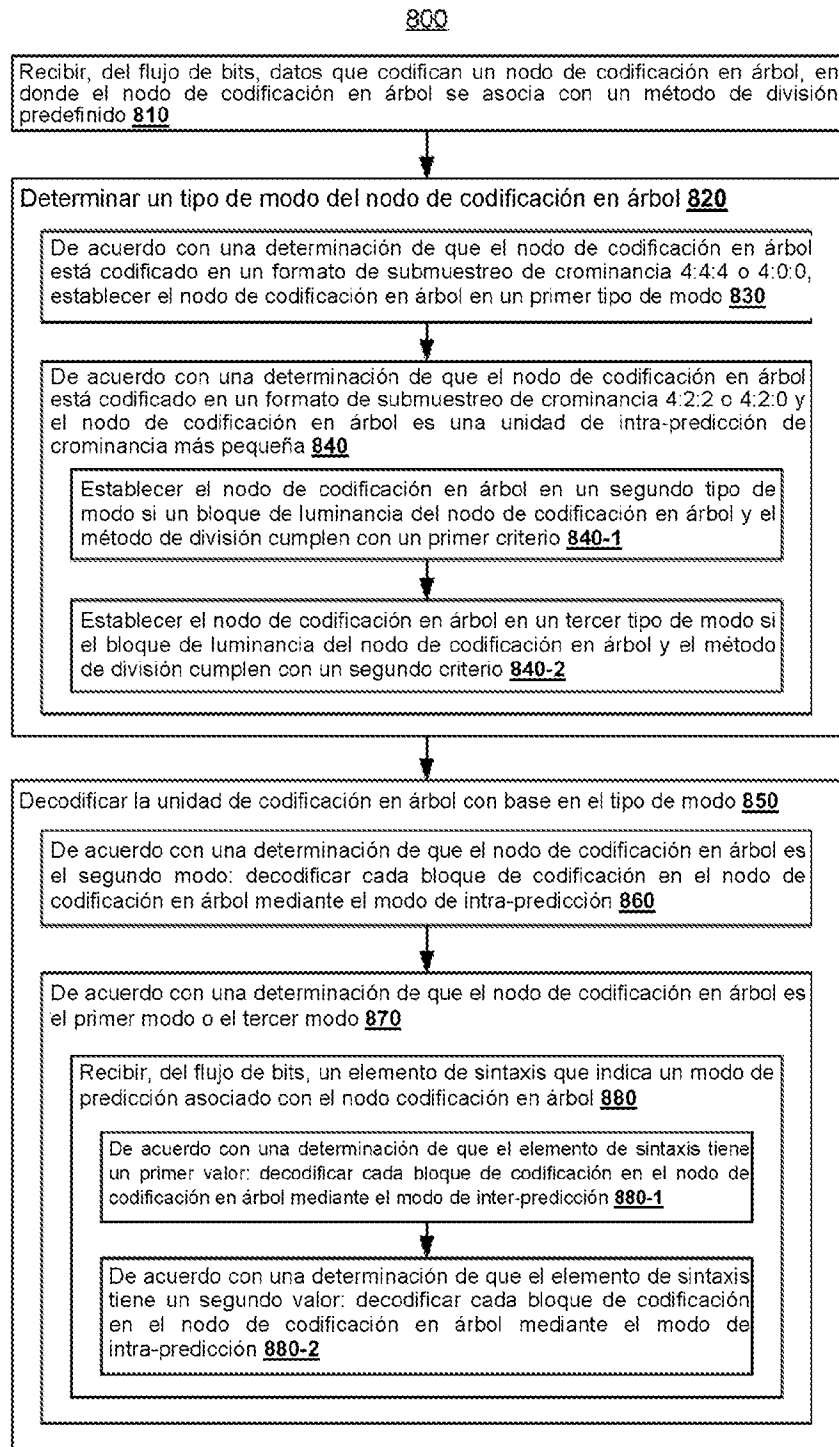


FIG. 8