

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 995 045**

51 Int. Cl.:

**H04N 19/52** (2014.01)  
**H04N 19/184** (2014.01)  
**H04N 19/119** (2014.01)  
**H04N 19/44** (2014.01)  
**H04N 19/42** (2014.01)  
**H04N 19/174** (2014.01)  
**H04N 19/436** (2014.01)  
**H04N 19/513** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **16.07.2019** **PCT/US2019/041923**  
 87 Fecha y número de publicación internacional: **23.01.2020** **WO20018486**  
 96 Fecha de presentación y número de la solicitud europea: **16.07.2019** **E 19837247 (6)**  
 97 Fecha y número de publicación de la concesión europea: **04.09.2024** **EP 3808090**

54 Título: **Métodos y aparatos de codificación de video que usan predicción de vector de movimiento basada en historial**

30 Prioridad:

**18.07.2018 US 201862700106 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**05.02.2025**

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:

**WANG, XIANGLIN y  
 CHEN, YI-WEN**

74 Agente/Representante:

**CARVAJAL Y URQUIJO, Isabel**

ES 2 995 045 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

Métodos y aparatos de codificación de video que usan predicción de vector de movimiento basada en historial

5 Campo técnico

La presente solicitud se refiere en general a codificación y decodificación de datos de video y en particular, a método y sistema de codificación de video usando predicción de vector de movimiento basada en historial.

10 Antecedentes

El video digital es compatible con una variedad de dispositivos electrónicos, tal como televisores digitales, computadoras portátiles o de escritorio, computadoras tipo tableta, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, consolas de videojuegos, teléfonos inteligentes, dispositivos de videoconferencia, dispositivos de transmisión de video, etc. Los dispositivos electrónicos transmiten, reciben, codifican, decodifican y/o almacenan datos de video digital al implementar normas de compresión/descompresión de video como se define por MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Video (AVC), Codificación de Video de Alta Eficiencia (HEVC) y norma de Codificación Versátil de Video (VVC). La compresión de video habitualmente incluye realizar predicción espacial (intra-trama) y/o predicción temporal (inter-trama) para reducir o remover redundancia inherente en los datos de video. Para codificación de video basada en bloques, una trama de video se divide en uno o más segmentos, cada segmento que tiene múltiples bloques de video, que también se pueden referir como unidades de árbol de codificación (CTU). Cada CTU puede contener una unidad de codificación (CU) o dividirse recursivamente en CU más pequeñas hasta que se alcance el tamaño mínimo predefinido de CU. Cada CU (también nombrada CU hoja) contiene una o múltiples unidades de transformada (TU) y cada CU también contiene una o múltiples unidades de predicción (PU). Cada CU se puede codificar en modos intra, inter o IBC. Los bloques de video en un segmento intracodificado (I) de una trama de video se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos dentro de la misma trama de video. Los bloques de video en un segmento intercodificado (P o B) de una trama de video pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos dentro de la misma trama de video o predicción temporal con respecto a muestras de referencia en otras tramas de video de referencia anteriores y/o futuras.

La predicción espacial o temporal con base en un bloque de referencia que se ha codificado previamente, por ejemplo, un bloque vecino, da por resultado un bloque predictivo para un bloque de video actual que se va a codificar. El proceso de encontrar el bloque de referencia se puede lograr por el 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 se refieren como un bloque residual o errores de predicción. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de referencia en una trama de referencia que forma el bloque predictivo, y el bloque residual. El proceso para determinar el vector de movimiento se refiere habitualmente como estimación de movimiento. Un bloque intracodificado se codifica de acuerdo con un modo de intrapredicción y el bloque residual. Para compresión adicional, el bloque residual se transforma del dominio de pixel a un dominio de transformada, por ejemplo, dominio de frecuencia, dando por resultado coeficientes de transformada residuales, que entonces se pueden cuantificar. Los coeficientes de transformada cuantificados, inicialmente dispuestos en una matriz bidimensional, se pueden escanear para producir un vector unidimensional de coeficientes de transformada, y entonces se codifican por entropía en un flujo de bits de video para lograr aún más compresión.

El flujo de bits de video codificado entonces se guarda en un medio de almacenamiento leible por computadora (por ejemplo, memoria flash) al que se accede por otro dispositivo electrónico con capacidad de video digital o se transmite directamente al dispositivo electrónico por cable o de forma inalámbrica. El dispositivo electrónico entonces realiza descompresión de video (que es un proceso opuesto a la compresión de video descrita anteriormente), por ejemplo, al analizar el flujo de bits de video codificado para obtener elementos de sintaxis del flujo de bits y reconstruyendo los datos de video digital a su formato original del flujo de bits de video codificado con base al menos en parte en los elementos de sintaxis obtenidos del flujo de bits, y renderiza los datos de video digital reconstruidos en una pantalla del dispositivo electrónico.

Con la calidad del video digital pasando de alta definición, a 4Kx2K o incluso 8Kx4K, la cantidad de datos de video que se van a codificar/decodificar crece exponencialmente. Es un desafío constante en términos de cómo los datos de video se pueden codificar/decodificar de manera más eficiente en tanto que se mantiene la calidad de imagen de los datos de video decodificados.

Documento de técnica anterior:

65 ZHANG (BYTEDANCE) L ET AL: "CE4-related: History-based Motion Vector Prediction", 11. REUNIÓN de JVET; 11-18 de julio de 2018, LIUBLIANA, No. JVETK0104, 14 de julio de 2018, divulga predicción de vector de movimiento basada en historial (HMVP).

Documento de técnica anterior:

5 GARY J SULLIVAN ET AL: "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE  
 TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE, US, vol. 22, no. 12,  
 diciembre de 2012, páginas 1649-1668, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2012.2221191,  
 proporciona una buena visión general de la norma de codificación de video HEVC y divulga implementaciones  
 paralelas de HEVC que prevén predicción de vector de movimiento (en particular por medio de AMVP y/o usando  
 el modo de fusión).

10 Sumario

15 La presente solicitud describe implementaciones relacionadas con codificación y decodificación de datos de video  
 y más en particular, con sistema y método de procesamiento paralelo de datos de video durante codificación y  
 decodificación de video usando predicción de vector de movimiento basada en historial. La invención se refiere a  
 un método de decodificación de datos de video como se define en la reivindicación independiente 1 y a un  
 dispositivo informático correspondiente, un medio de almacenamiento leíble por computadora no transitorio y un  
 producto de programa de computadora como se define, respectivamente, en las reivindicaciones independientes  
 8, 10 y 11. Las reivindicaciones dependientes definen características opcionales adicionales de la invención.

20 Breve descripción de los dibujos

25 Los dibujos anexos, que se incluyen para proporcionar una comprensión adicional de las implementaciones y se  
 incorporan en la presente y constituyen una parte de la especificación, ilustran las implementaciones descritas y  
 junto con la descripción sirven para explicar los principios subyacentes. Los números de referencia similares se  
 refieren a las partes correspondientes.

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

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

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

35 Las figuras 4A-4D son diagramas de bloques que ilustran cómo una trama se divide recursivamente en árboles  
 cuaternarios en múltiples bloques de video de diferentes tamaños de acuerdo con algunas implementaciones de  
 la presente divulgación.

La figura 5A es un diagrama de bloques que ilustra posiciones de bloque vecinas espacialmente y colocadas  
 temporalmente de una CU actual que se va a codificar de acuerdo con algunas implementaciones de la presente  
 divulgación.

40 La figura 5B es un diagrama de bloques que ilustra la codificación de múltiples subprocesos de múltiples filas de  
 las CTU de una imagen usando procesamiento paralelo de frente de onda de acuerdo con algunas  
 implementaciones de la presente divulgación.

45 La figura 6 es un diagrama de flujo que ilustra un proceso de ejemplo por el cual un codificador de video implementa  
 las técnicas de construcción de una lista de candidatos de predictor de vector de movimiento de acuerdo con  
 algunas implementaciones de la presente divulgación.

Descripción detallada

50 La divulgación habilitante de la invención se expone en la figura 6 con las partes correspondientes de la  
 descripción. Las otras partes de la descripción, a menos que proporcionen detalles adicionales con respecto a la  
 divulgación habilitante mencionada anteriormente, se van a entender como destinadas únicamente a propósitos  
 ilustrativos.

55 Ahora se hará referencia con detalle a implementaciones específicas, los ejemplos de la cual se ilustran en los  
 dibujos anexos. En la siguiente descripción detallada, se exponen numerosos detalles específicos no limitantes a  
 fin de ayudar a comprender la materia presentada en la presente. Pero será evidente para un experto en la técnica  
 que se pueden usar diferentes alternativas sin apartarse del alcance de las reivindicaciones y la materia se puede  
 practicar sin estos detalles específicos. Por ejemplo, será evidente para un experto en la técnica que la materia  
 presentada en la presente se puede implementar en muchos tipos de dispositivos electrónicos con capacidades  
 60 de video digital.

65 La figura 1 es un diagrama de bloques que ilustra un sistema de ejemplo 10 para codificar y decodificar bloques  
 de video en paralelo de acuerdo con algunas implementaciones de la presente divulgación. Como se muestra en  
 la figura 1, el sistema 10 incluye un dispositivo fuente 12 que genera y codifica datos de video que se van a  
 decodificar en un momento posterior por un dispositivo de destino 14. El dispositivo fuente 12 y dispositivo de  
 destino 14 pueden comprender cualquiera de una amplia variedad de dispositivos electrónicos, incluidas

computadoras de escritorio o portátiles, computadoras tipo tableta, teléfonos inteligentes, decodificadores, televisiones digitales, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivo de transmisión de video o similares. En algunas implementaciones, el dispositivo fuente 12 y dispositivo de destino 14 se equipan con capacidades de comunicación inalámbrica.

En algunas implementaciones, el dispositivo de destino 14 puede recibir los datos de video codificados que se van a decodificar mediante un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo de comunicación capaz de mover los datos de video codificados desde el dispositivo fuente 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo fuente 12 transmita datos de video codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de video codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o alámbrico tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como 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 fuente 12 al dispositivo de destino 14.

En algunas implementaciones diferentes, los datos de video codificados se pueden transmitir desde la interfaz de salida 22 a un dispositivo de almacenamiento 32. Posteriormente, se puede acceder a los datos de video codificados en el dispositivo de almacenamiento 32 por dispositivo de destino 14 mediante la interfaz de entrada 28. El dispositivo de almacenamiento 32 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o a los que se puede tener acceso localmente tal como un disco duro, discos Blu-ray, DVD, CD-R $\square$ M, memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de video codificados. En un ejemplo adicional, el dispositivo de almacenamiento 32 puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede retener los datos de video codificados generados por el dispositivo fuente 12. El dispositivo de destino 14 puede tener acceso a los datos de video almacenados del dispositivo de almacenamiento 32 mediante transmisión o descarga. El servidor de archivos puede ser cualquier tipo de computadora capaz de almacenar datos de video codificados y transmitir los datos de video codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectado a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede tener acceso a los datos de video codificados a través de cualquier conexión de datos estándar, incluido un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que es adecuada para tener acceso a datos de video codificados almacenados en un servidor de archivos. La transmisión de datos de video codificados del dispositivo de almacenamiento 32 puede ser una transmisión de transmisión en tiempo real, una transmisión de descarga o una combinación de ambas.

Como se muestra en la figura 1, el dispositivo fuente 12 incluye una fuente de video 18, un codificador de video 20 y una interfaz de salida 22. La fuente de video 18 puede incluir una fuente tal como un dispositivo de captura de video, por ejemplo, una cámara de video, un archivo de video que contiene video previamente capturado, una interfaz de alimentación de video para recibir video de un proveedor de contenido de video y/o un sistema de gráficos de computadora para generar datos gráficos de computadora como el video fuente, o una combinación de estas fuentes. Como ejemplo, si la fuente de video 18 es una cámara de video de un sistema de vigilancia de seguridad, el dispositivo fuente 12 y dispositivo de destino 14 pueden formar teléfonos de cámara o teléfonos de video. Sin embargo, las implementaciones descritas en la presente solicitud se pueden aplicar a codificación de video en general y se pueden aplicar a aplicaciones inalámbricas y/o alámbricas.

El video capturado, pre-capturado o generado por computadora se puede codificar por el codificador de video 20. Los datos de video codificados se pueden transmitir directamente al dispositivo de destino 14 mediante la interfaz de salida 22 del dispositivo fuente 12. Los datos de video codificados también se pueden almacenar (o de manera alternativa) en el dispositivo de almacenamiento 32 para acceso posterior por el dispositivo de destino 14 u otros dispositivos, para decodificación y/o reproducción. La interfaz de salida 22 puede incluir además un módem y/o un transmisor.

El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de video 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 video codificados a través del enlace 16. Los datos de video codificados comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 32, pueden incluir una variedad de elementos de sintaxis generados por el codificador de video 20 para usarse por el decodificador de video 30 al decodificar los datos de video. Estos elementos sintácticos se pueden incluir dentro de los datos de video codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

En algunas implementaciones, el dispositivo de destino 14 puede incluir un dispositivo de visualización 34, que puede ser un dispositivo de visualización integrado y un dispositivo de visualización externo que se configura para comunicarse con el dispositivo de destino 14. El dispositivo de visualización 34 muestra los datos de video

decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tal como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

5 El codificador de video 20 y decodificador de video 30 pueden operar de acuerdo con normas patentadas o de la industria, tal como VVC, HEVC, MPEG-4, Parte 10, codificación avanzada de video (AVC) o extensiones de estas normas. Se debe entender que la presente solicitud no se limita a una norma de codificación/decodificación de video específica y puede ser aplicable a otras normas de codificación/decodificación de video. En general, se contempla que el codificador de video 20 del dispositivo fuente 12 se puede configurar para codificar datos de video de acuerdo con cualquiera de estas normas actuales o futuras. De manera similar, también se contempla en general que el decodificador de video 30 del dispositivo de destino 14 se puede configurar para decodificar datos de video de acuerdo con cualquiera de estas normas actuales o futuras.

15 El codificador de video 20 y decodificador de video 30 cada uno se puede implementar como cualquiera de una variedad de circuitería de codificador adecuada, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), conjuntos de compuertas programables en campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando se implementa parcialmente en software, un dispositivo electrónico puede almacenar instrucciones para el software en un medio leible por computadora no transitorio, adecuado y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las operaciones de codificación/decodificación de video divulgadas en la presente divulgación. Cada uno del codificador de video 20 y decodificador de video 30 se pueden incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador combinado (códec) en un respectivo dispositivo.

25 La figura 2 es un diagrama de bloques que ilustra un codificador de video 20 de ejemplo de acuerdo con algunas implementaciones descritas en la presente solicitud. El codificador de video 20 puede realizar codificación intra e interpredictiva de bloques de video dentro de tramas de video. La codificación intrapredictiva se basa en predicción espacial para reducir o remover redundancia espacial en datos de video dentro de una imagen o trama de video dada. La codificación interpredictiva se basa en predicción temporal para reducir o remover redundancia temporal en datos de video dentro de tramas o imágenes de video adyacentes de una secuencia de video.

35 Como se muestra en la figura 2, el codificador de video 20 incluye memoria de datos de video 40, unidad de procesamiento de predicción 41, memoria intermedia de imagen decodificada (DPB) 64, sumador 50, unidad de procesamiento de transformada 52, unidad de cuantificación 54 y unidad de codificación entrópica 56. La unidad de procesamiento de predicción 41 incluye además la unidad de estimación de movimiento 42, unidad de compensación de movimiento 44, unidad de división 45, unidad de procesamiento de intrapredicción 46 y unidad de copia de intrabloque (BC) 48. En algunas implementaciones, el codificador de video 20 también incluye unidad de cuantificación inversa 58, unidad de procesamiento de transformada inversa 60 y sumador 62 para reconstrucción de bloques de video. Se puede colocar un filtro de desbloqueo (no mostrado) entre el sumador 62 y DPB 64 para filtrar límites de bloque para remover artefactos de bloqueo del video reconstruido. También se puede usar un filtro en bucle (no mostrado) además del filtro de desbloqueo para filtrar la salida del sumador 62. El codificador de video 20 puede tomar la forma de una unidad de hardware fija o programable o se puede dividir entre una o más de las unidades de hardware fijas o programables ilustradas.

45 La memoria de datos de video 40 puede almacenar datos de video que van a codificar por los componentes del codificador de video 20. Los datos de video en la memoria de datos de video 40 se pueden obtener, por ejemplo, de la fuente de video 18. La DPB 64 es una memoria intermedia que almacena datos de video de referencia para uso en la codificación de datos de video por el codificador de video 20 (por ejemplo, en modos de codificación intra o interpredictiva). La memoria de datos de video 40 y DPB 64 se pueden formar por cualquiera de una variedad de dispositivos de memoria. En diferentes ejemplos, la memoria de datos de video 40 puede estar en el chip con otros componentes del codificador de video 20, o fuera del chip con respecto a aquellos componentes.

55 Como se muestra en la figura 2, después de recibir datos de video, la unidad de división 45 dentro de la unidad de procesamiento de predicción 41 divide los datos de video en bloques de video. Esta división también puede incluir dividir una trama de video en segmentos, mosaicos u otras unidades de codificación (CU) más grandes de acuerdo con una estructura de división predefinida, tal como una estructura de árbol cuaternario asociada con los datos de video. La trama de video se puede dividir en múltiples bloques de video (o conjuntos de bloques de video referidos como mosaicos). La unidad de procesamiento de predicción 41 puede seleccionar uno de una pluralidad de posibles modos de codificación predictiva, tal como uno de una pluralidad de modos de codificación intrapredictiva o uno de una pluralidad de modos de codificación interpredictiva, para el bloque de video actual con base en los resultados de error (por ejemplo, velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque codificado por intra o interpredicción resultante al sumador 50 para generar un bloque residual y al sumador 62 para reconstruir el bloque codificado para uso como parte de una trama de referencia posteriormente. La unidad de procesamiento de predicción 41 también proporciona elementos de sintaxis, tal como vectores de movimiento, indicadores intramodo, información de partición y otra información de sintaxis, a la unidad de codificación entrópica 56.

A fin de seleccionar un modo de codificación intrapredictiva apropiado para el bloque de video actual, la unidad de procesamiento de intrapredicción 46 dentro de la unidad de procesamiento de predicción 41 puede realizar codificación intrapredictiva del bloque de video actual con respecto a uno o más bloques vecinos en la misma trama que el bloque actual que se va a codificar para proporcionar predicción espacial. La unidad de estimación de movimiento 42 y unidad de compensación de movimiento 44 dentro de unidad de procesamiento de predicción 41 realizan codificación interpredictiva del bloque de video actual con respecto a uno o más bloques predictivos en una o más tramas de referencia para proporcionar predicción temporal. El codificador de video 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación adecuado para cada bloque de datos de video.

En algunas implementaciones, la unidad de estimación de movimiento 42 determina el modo de interpredicción para una trama de video actual al generar un vector de movimiento, que indica el desplazamiento de una unidad de predicción (PU) de un bloque de video dentro de la trama de video actual con respecto a un bloque predictivo dentro de una trama de video de referencia, de acuerdo con un patrón predeterminado dentro de una secuencia de tramas de video. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso para generar vectores de movimiento, que estiman el movimiento para bloques de video. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de video dentro de una trama o imagen de video actual con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada) con respecto al bloque actual que se codifica dentro de la trama actual (u otra unidad codificada). El patrón predeterminado puede designar tramas de video en la secuencia como tramas P o tramas B. La unidad intra BC 48 puede determinar vectores, por ejemplo, vectores de bloque, para codificación intra BC de una manera similar a la determinación de vectores de movimiento por la unidad de estimación de movimiento 42 para interpredicción, o puede utilizar la unidad de estimación de movimiento 42 para determinar el vector de bloque.

Un bloque predictivo es un bloque de una trama de referencia que se considera que coincide estrechamente con la PU del bloque de video que se va a codificar en términos de diferencia de píxeles, que se puede determinar por suma de diferencia absoluta (SAD), suma de diferencia cuadrada (SSD) u otras métricas de diferencia. En algunas implementaciones, el codificador de video 20 puede calcular valores para posiciones de píxeles sub-enteros de tramas de referencia almacenadas en la DPB 64. Por ejemplo, el codificador de video 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo píxel u otras posiciones de píxeles fraccionarios de la trama de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxeles completos y posiciones de píxeles fraccionarios y generar un vector de movimiento con precisión de píxeles fraccionarios.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de video en una trama codificada por interpredicción al comparar la posición de la PU con la posición de un bloque predictivo de una trama de referencia seleccionada de una primera lista de tramas de referencia (lista 0) o una segunda lista de tramas de referencia (lista 1), cada una de las cuales identifica una o más tramas de referencia almacenadas en DPB 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de compensación de movimiento 44 y entonces a la unidad de codificación entrópica 56.

La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar recuperar o generar el bloque predictivo con base en el vector de movimiento determinado por la unidad de estimación de movimiento 42. Después de recibir el vector de movimiento para la PU del bloque de video actual, la unidad de compensación de movimiento 44 puede ubicar un bloque predictivo al cual apunta el vector de movimiento en una de las listas de tramas de referencia, recuperar el bloque predictivo de la DPB 64 y reenviar el bloque predictivo al sumador 50. Entonces, el sumador 50 forma un bloque de video residual de valores de diferencia de píxeles al restar valores de píxeles del bloque predictivo proporcionado por la unidad de compensación de movimiento 44 de los valores de píxeles del bloque de video actual que se codifica. Los valores de diferencia de píxeles que forman el bloque de video residual pueden incluir componentes de diferencia de luma o croma o ambos. La unidad de compensación de movimiento 44 también puede generar elementos de sintaxis asociados con los bloques de video de una trama de video para uso por el decodificador de video 30 al decodificar los bloques de video de la trama de video. Los elementos de sintaxis pueden incluir, por ejemplo, elementos de sintaxis que definen el vector de movimiento usado para identificar el bloque predictivo, cualquier bandera que indique el modo de predicción, o cualquier otra información de sintaxis descrita en la presente. Se señalar que la unidad de estimación de movimiento 42 y unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado para propósitos conceptuales.

En algunas implementaciones, la unidad intra BC 48 puede generar vectores y recuperar bloques predictivos de una manera similar a la descrita anteriormente en relación con la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, pero con los bloques predictivos que están en la misma trama que el bloque actual que se está codificando y con los vectores a los que se hace referencia como vectores de bloque en lugar de vectores de movimiento. En particular, la unidad intra BC 48 puede determinar un modo de intrapredicción para uso para codificar un bloque actual. En algunos ejemplos, la unidad intra BC 48 puede determinar un modo de intrapredicción para uso para codificar un bloque actual usando diferentes modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas, y probar su rendimiento a través del análisis de velocidad-distorsión. Después, la unidad intra BC 48 puede

seleccionar, entre los diferentes modos de intrapredicción probados, un modo de intrapredicción apropiado para usar y generar un indicador de intramodo en consecuencia. Por ejemplo, la unidad intra BC 48 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diferentes modos de intrapredicción probados y seleccionar el modo de intrapredicción que tiene las mejores características de velocidad-distorsión entre los modos probados como el modo de intrapredicción apropiado para uso. El análisis de velocidad-distorsión en general determina una cantidad de distorsión (o error) entre un bloque codificado y un bloque no codificado original que se codificó para producir el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad intra BC 48 puede calcular relaciones a partir de las distorsiones y velocidades para los diferentes bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor de velocidad-distorsión para el bloque.

En otros ejemplos, la unidad intra BC 48 puede usar la unidad de estimación de movimiento 42 y unidad de compensación de movimiento 44, en su totalidad o en parte, para realizar estas funciones para predicción intra BC de acuerdo con las implementaciones descritas en la presente. En cualquier caso, para la copia de intrabloque, un bloque predictivo puede ser un bloque que se considera que coincide estrechamente con el bloque que se va a codificar, en términos de diferencia de píxeles, que se puede determinar por suma de diferencia absoluta (SAD), suma de diferencia cuadrada (SSD) u otras métricas de diferencia, y la identificación del bloque predictivo puede incluir el cálculo de valores para posiciones de píxeles sub-enteros.

Si el bloque predictivo es de la misma trama de acuerdo con la intrapredicción, o una trama diferente de acuerdo con la interpredicción, el codificador de video 20 puede formar un bloque de video residual al restar valores de píxeles del bloque predictivo de los valores de píxeles del bloque de video actual que se codifica, formando valores de diferencia de píxeles. Los valores de diferencia de píxeles que forman el bloque de video residual pueden incluir diferencias tanto de componentes de luma como de croma.

La unidad de procesamiento de intrapredicción 46 puede intraprededir un bloque de video actual como una alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y unidad de compensación de movimiento 44, o la predicción de copia intrabloque realizada por la unidad intra BC 48, como se describió anteriormente. En particular, la unidad de procesamiento de intrapredicción 46 puede determinar un modo de intrapredicción para uso para codificar un bloque actual. Para hacerlo, la unidad de procesamiento de intrapredicción 46 puede codificar un bloque actual usando diferentes modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas, y la unidad de procesamiento de intrapredicción 46 (o una unidad de selección de modo, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado para uso de los modos de intrapredicción probados. La unidad de procesamiento de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad de codificación entrópica 56. La unidad de codificación entrópica 56 puede codificar la información que indica el modo de intrapredicción seleccionado en el flujo de bits.

Después de que la unidad de procesamiento de predicción 41 determina el bloque predictivo para el bloque de video actual mediante interpredicción o intrapredicción, el sumador 50 forma un bloque de video residual al restar el bloque predictivo del bloque de video actual. Los datos de video residuales en el bloque residual se pueden incluir 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 video residuales en coeficientes de transformada residuales usando una transformada, tal como una transformada de coseno discreta (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 adicionalmente 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 se puede modificar al ajustar un parámetro de cuantificación. En algunos ejemplos, unidad de cuantificación 54 puede realizar entonces un escaneo de una matriz que incluye los coeficientes de transformada cuantificados. De manera alternativa, la unidad de codificación entrópica 56 puede realizar el escaneo.

Después de la cuantificación, la unidad de codificación entrópica 56 codifica por entropía los coeficientes de transformada cuantificados en un flujo de bits de video usando, por ejemplo, codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación entrópica de división de intervalo de probabilidad (PIPE) u otra metodología o técnica de codificación entrópica. El flujo de bits codificado entonces se puede transmitir al decodificador de video 30 o archivar en el dispositivo de almacenamiento 32 para transmisión posterior o recuperación por el decodificador de video 30. La unidad de codificación entrópica 56 también puede codificar por entropía los vectores de movimiento y los otros elementos de sintaxis para la trama de video actual que se codifica.

La unidad de cuantificación inversa 58 y unidad de procesamiento de transformada inversa 60 aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque de video residual en

el dominio de píxeles para generar un bloque de referencia para predicción de otros bloques de video. 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 las tramas almacenadas 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 valores de píxeles sub-enteros para uso en la estimación de movimiento.

El sumador 62 suma el bloque residual reconstruido con el bloque predictivo compensado en movimiento, producido por la unidad de compensación de movimiento 44 para producir un bloque de referencia para almacenamiento en la DPB 64. El bloque de referencia se puede usar entonces por la unidad intra BC 48, unidad de estimación de movimiento 42 y unidad de compensación de movimiento 44 como un bloque predictivo para interpredecir otro bloque de video en una trama de video posterior.

La figura 3 es un diagrama de bloques que ilustra un decodificador de video 30 de ejemplo de acuerdo con algunas implementaciones de la presente solicitud. El decodificador de video 30 incluye memoria de datos de video 79, unidad de decodificación entrópica 80, unidad de procesamiento de predicción 81, unidad de cuantificación inversa 86, unidad de procesamiento de transformada inversa 88, sumador 90 y DPB 92. La unidad de procesamiento de predicción 81 incluye además la unidad de compensación de movimiento 82, unidad de procesamiento de intrapredicción 84 y unidad de intra BC 85. El decodificador de video 30 puede realizar un proceso de decodificación en general recíproco al proceso de codificación descrito anteriormente con respecto al codificador de video 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 entrópica 80, en tanto que la unidad de intrapredicción 84 puede generar datos de predicción con base en indicadores de modo de intrapredicción recibidos de la unidad de decodificación entrópica 80.

En algunos ejemplos, se puede asignar a una unidad de decodificador de video 30 la tarea de realizar las implementaciones de la presente solicitud. También, en algunos ejemplos, las implementaciones de la presente divulgación se pueden dividir entre una o más de las unidades del decodificador de video 30. Por ejemplo, la unidad intra BC 85 puede realizar las implementaciones de la presente solicitud, sola o en combinación con otras unidades del decodificador de video 30, tal como la unidad de compensación de movimiento 82, unidad de procesamiento de intrapredicción 84 y unidad de decodificación entrópica 80. En algunos ejemplos, el decodificador de video 30 puede no incluir la unidad intra BC 85 y la funcionalidad de la unidad intra BC 85 se puede realizar por otros componentes de la unidad de procesamiento de predicción 81, tal como la unidad de compensación de movimiento 82.

La memoria de datos de video 79 puede almacenar datos de video, tal como un flujo de bits de video codificado, que se va a decodificar por los otros componentes del decodificador de video 30. Los datos de video almacenados en la memoria de datos de video 79 se pueden obtener, por ejemplo, del dispositivo de almacenamiento 32, de una fuente de video local, tal como una cámara, mediante comunicación de datos de video en red alámbrica o inalámbrica, o por el acceso a medios de almacenamiento de datos físicos (por ejemplo, una unidad flash o disco duro). La memoria de datos de video 79 puede incluir una memoria intermedia de imagen codificada (CPB) que almacena datos de video codificados de un flujo de bits de video codificado. La memoria intermedia de imagen decodificada (DPB) 92 del decodificador de video 30 almacena datos de video de referencia para uso en la decodificación de datos de video por el decodificador de video 30 (por ejemplo, en modos de codificación intra o interpredictiva). La memoria de datos de video 79 y DPB 92 se pueden formar por cualquiera de una variedad de dispositivos de memoria, tal como memoria de acceso aleatorio dinámica (DRAM), incluida DRAM síncrona (SDRAM), RAM magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. Para propósitos ilustrativos, la memoria de datos de video 79 y la DPB 92 se representan como dos componentes distintos del decodificador de video 30 en la figura 3. Pero será evidente para un experto en la técnica que la memoria de datos de video 79 y DPB 92 se pueden proporcionar por el mismo dispositivo de memoria o dispositivos de memoria separados. En algunos ejemplos, la memoria de datos de video 79 puede estar en el chip con otros componentes del decodificador de video 30, o fuera del chip con respecto a aquellos componentes.

Durante el proceso de decodificación, el decodificador de video 30 recibe un flujo de bits de video codificado que representa bloques de video de una trama de video codificada y elementos de sintaxis asociados. El decodificador de video 30 puede recibir los elementos de sintaxis al nivel de trama de video y/o al nivel de bloque de video. La unidad de decodificación entrópica 80 del decodificador de video 30 decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción y otros elementos de sintaxis. La unidad de decodificación entrópica 80 entonces reenvía los vectores de movimiento y otros elementos de sintaxis a la unidad de procesamiento de predicción 81.

Cuando la trama de video se codifica como una trama codificada intrapredictiva (I) o para bloques predictivos intracodificados en otros tipos de tramas, la unidad de procesamiento de intrapredicción 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de video de la trama de video actual con base en un modo de intrapredicción señalado y datos de referencia de bloques previamente decodificados de la trama actual.

5 Cuando la trama de video se codifica como una trama codificada interpredictiva (es decir, B o P), la unidad de compensación de movimiento 82 de la unidad de procesamiento de predicción 81 produce uno o más bloques predictivos para un bloque de video de la trama de video actual con base en los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad de decodificación entrópica 80. Cada uno de los bloques predictivos se puede producir a partir de una trama de referencia dentro de una de las listas de tramas de referencia. El decodificador video 30 puede construir las listas de tramas de referencia, lista 0 y lista 1, usando técnicas de construcción predeterminadas con base en tramas de referencia almacenadas en la DPB 92.

10 En algunos ejemplos, cuando el bloque de video se codifica de acuerdo con el modo intra BC descrito en la presente, la unidad intra BC 85 de la unidad de procesamiento de predicción 81 produce bloques predictivos para el bloque de video actual con base en vectores de bloque y otros elementos de sintaxis recibidos de la unidad de decodificación entrópica 80. Los bloques predictivos pueden estar dentro de una región reconstruida de la misma imagen que el bloque de video actual definido por el codificador de video 20.

15 La unidad de compensación de movimiento 82 y/o unidad intra BC 85 determina la información de predicción para un bloque de video de la trama de video actual por análisis de los vectores de movimiento y otros elementos de sintaxis y entonces usa la información de predicción para producir los bloques predictivos para el bloque de video actual que se decodifica. Por ejemplo, la unidad de compensación de movimiento 82 usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra o interpredicción) usado para  
20 codificar bloques de video de la trama de video, un tipo de trama de interpredicción (por ejemplo, B o P), información de construcción para una o más de las listas de tramas de referencia para la trama, vectores de movimiento para cada bloque de video codificado interpredictivo de la trama, estado de interpredicción para cada bloque de video codificado interpredictivo de la trama y otra información para decodificar los bloques de video en la trama de video actual.

25 De manera similar, la unidad intra BC 85 puede usar algunos de los elementos de sintaxis recibidos, por ejemplo, una bandera, para determinar que el bloque de video actual se predijo usando el modo intra BC, información de construcción de qué bloques de video de la trama están dentro de la región reconstruida y se deben almacenar en la DPB 92, vectores de bloque para cada bloque de video predicho intra BC de la trama, estado de predicción intra  
30 BC para cada bloque de video predicho intra BC de la trama y otra información para decodificar los bloques de video en la trama de video actual.

35 La unidad de compensación movimiento 82 puede realizar también interpolación usando los filtros de interpolación como se usa por el codificador de video 20 durante la codificación de los bloques de video para calcular valores interpolados para píxeles sub-enteros de bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación usados por el codificador de video 20 de los elementos de sintaxis recibidos y usar los filtros de interpolación para producir bloques predictivos.

40 La unidad de cuantificación inversa 86 cuantifica inversamente los coeficientes de transformada cuantificados proporcionados en el flujo de bits y decodificados por entropía por la unidad de decodificación entrópica 80 usando el mismo parámetro de cuantificación calculado por el codificador de video 20 para cada bloque de video en la trama de video 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 entera inversa o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada a fin de reconstruir los  
45 bloques residuales en el dominio de píxeles.

Después de que la unidad de compensación de movimiento 82 o la unidad intra BC 85 genera el bloque predictivo para el bloque de video actual con base en los vectores y otros elementos de sintaxis, el sumador 90 reconstruye el bloque de video decodificado para el bloque de video actual al sumar el bloque residual de la unidad de  
50 procesamiento de transformada inversa 88 y un bloque predictivo correspondiente generado por la unidad de compensación de movimiento 82 y la unidad intra BC 85. Se puede colocar un filtro en bucle (no ilustrado) entre el sumador 90 y DPB 92 para procesar adicionalmente el bloque de video decodificado. Los bloques de video decodificados en una trama dada entonces se almacenan en la DPB 92, que almacena tramas de referencia usadas para compensación de movimiento posterior de los siguientes bloques de video. La DPB 92, o un dispositivo de memoria separado de la DPB 92, también puede almacenar video decodificado para posterior  
55 presentación en un dispositivo de visualización, tal como el dispositivo de visualización 34 de la figura 1.

60 En un proceso habitual de codificación de video, una secuencia de video habitualmente incluye un conjunto ordenado de tramas o imágenes. Cada trama puede incluir tres matrices de muestra, denotadas SL, SCb y SCr. SL es una matriz bidimensional de muestras de luma. SCb es una matriz bidimensional de muestras de croma Cb. SCr es una matriz bidimensional de muestras de croma Cr. En otros casos, una trama puede ser monocroma y por lo tanto, incluye solo una matriz bidimensional de muestras de luma.

65 Como se muestra en la figura 4A, el codificador de video 20 (o más específicamente la unidad de división 45) genera una representación codificada de una trama al dividir primero la trama en un conjunto de unidades de árbol de codificación (CTU). Una trama de video puede incluir un número entero de CTU ordenadas consecutivamente

- en un orden de escaneo de trama de izquierda a derecha y de arriba a abajo. Cada CTU es una unidad de codificación lógica más grande y el ancho y altura de la CTU se señalan por el codificador de video 20 en un conjunto de parámetros de secuencia, de modo que todas las CTU en una secuencia de video tienen el mismo tamaño que es uno de 1283128, 64364, 32332 y 16316. Pero se debe señalar que la presente solicitud no se limita necesariamente a un tamaño particular. Como se muestra en la figura 4B, cada CTU puede comprender un bloque de árbol de codificación (CTB) de muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma y elementos de sintaxis usados para codificar las muestras de los bloques de árbol de codificación. Los elementos de sintaxis describen propiedades de diferentes tipos de unidades de un bloque codificado de píxeles y cómo se puede reconstruir la secuencia de video en el decodificador de video 30, que incluye interpredicción o intrapredicción, modo de intrapredicción, vectores de movimiento y otros parámetros. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CTU puede comprender un bloque de árbol de codificación individual y elementos de sintaxis usados para codificar las muestras del bloque de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque NxN de muestras.
- Para lograr un mejor rendimiento, el codificador de video 20 puede realizar de forma recursiva la división de árbol tal como división de árbol binario, división de árbol cuaternario o una combinación de ambos en los bloques de árbol de codificación 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, CU 410 y CU 420 se dividen cada una en cuatro CU de 16x16 por tamaño de bloque. Las dos CU 430 y 440 de 16x16 se dividen cada una además en cuatro CU de 8x8 por tamaño de bloque. La figura 4D representa una estructura de datos de árbol cuaternario que ilustra el resultado final del proceso de división de la CTU 400 como se representa en la figura 4C, cada nodo hoja del árbol cuaternario que corresponde a una CU de un tamaño respectivo que varía de 32x32 a 8x8. Al igual que la CTU representada en la figura 4B, cada CU puede comprender un bloque de codificación (CB) de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una trama del mismo tamaño, y elementos de sintaxis usados para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CU puede comprender un bloque de codificación individual y estructuras de sintaxis usadas para codificar las muestras del bloque de codificación.
- En algunas implementaciones, el codificador de video 20 puede dividir además un bloque de codificación de una CU en uno o más bloques de predicción MxN (PB). Un bloque de predicción es un bloque rectangular (cuadrado o no cuadrado) de muestras en el que se aplica la misma predicción, inter o intra. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y elementos de sintaxis usados para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una PU puede comprender un bloque de predicción individual y estructuras de sintaxis usadas para predecir el bloque de predicción. El codificador de video 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.
- El codificador de video 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de video 20 usa intrapredicción para generar los bloques predictivos de una PU, el codificador de video 20 puede generar los bloques predictivos de la PU con base en muestras decodificadas de la trama asociada con la PU. Si el codificador de video 20 usa interpredicción para generar los bloques predictivos de una PU, el codificador de video 20 puede generar los bloques predictivos de la PU con base en muestras decodificadas de una o más tramas diferentes de la trama asociada con la PU.
- Después de que el codificador de video 20 genera bloques de luma predictivos, Cb y Cr para una o más PU de una CU, el codificador de video 20 puede generar un bloque residual de luma para la CU al restar los bloques de luma predictivos de la CU de su bloque de codificación de luma original de modo que cada muestra en el bloque residual de luma de la CU indique una diferencia entre una muestra de luma en uno de los bloques de luma predictivos de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. De manera similar, el codificador de video 20 puede generar un bloque residual Cb y un bloque residual Cr para la CU, respectivamente, de modo que cada muestra en el bloque residual Cb de la CU indique una diferencia entre una muestra Cb en uno de los bloques Cb predictivos de la CU y una muestra correspondiente en el bloque de codificación Cb original de la CU y cada muestra en el bloque residual Cr de la CU puede indicar una diferencia entre una muestra Cr en uno de los bloques Cr predictivos de la CU y una muestra correspondiente en el bloque de codificación Cr original de la CU.
- Además, como se ilustra en la figura 4C, el codificador de video 20 puede usar división de árbol cuaternario para descomponer los bloques residuales de luma, Cb y Cr de una CU en uno o más bloques de transformada de luma, 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 luma, dos bloques de transformada correspondientes de muestras de croma y elementos de sintaxis usados para transformar las muestras de bloque de transformada. Por lo tanto, cada TU de una CU se puede asociar con un bloque de transformada de luma, un bloque de transformada Cb y un bloque de transformada Cr. En algunos ejemplos, el bloque de transformada de luma asociado con la TU puede ser un

subbloque del bloque residual de luma de la CU. El bloque de transformada Cb puede ser un sub-bloque del bloque residual Cb de la CU. El bloque de transformada Cr puede ser un sub-bloque del bloque residual Cr de la CU. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una TU puede comprender un bloque de transformada individual y estructuras de sintaxis usadas para transformar las muestras del bloque de transformada.

El codificador de video 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficiente de luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. El codificador de video 20 puede aplicar una o más transformadas a un bloque de transformada Cb de una TU para generar un bloque de coeficiente Cb para la TU. El codificador de video 20 puede aplicar una o más transformadas a un bloque de transformada Cb de una TU para generar un bloque de coeficiente Cb para la TU.

Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes Cb o un bloque de coeficientes Cr), el codificador de video 20 puede cuantificar el bloque de coeficientes. La cuantificación en general se refiere a un proceso en el cual los coeficientes de transformada se cuantifican para posiblemente reducir la cantidad de datos usados para representar los coeficientes de transformada, proporcionando compresión adicional. Después de que el codificador de video 20 cuantifica un bloque de coeficientes, el codificador de video 20 puede codificar por entropía elementos de sintaxis que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de video 20 puede realizar codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos de sintaxis que indican los coeficientes de transformada cuantificados. Finalmente, el codificador de video 20 puede emitir un flujo de bits que incluye una secuencia de bits que forma una representación de tramas codificadas y datos asociados, que se guarda en el dispositivo de almacenamiento 32 o se transmite al dispositivo de destino 14.

Después de recibir un flujo de bits generado por el codificador de video 20, el decodificador de video 30 puede analizar el flujo de bits para obtener elementos de sintaxis del flujo de bits. El decodificador de video 30 puede reconstruir las tramas de los datos de video con base al menos en parte en los elementos de sintaxis obtenidos del flujo de bits. El proceso de reconstrucción de los datos de video es en general recíproco al proceso de codificación realizado por el codificador de video 20. Por ejemplo, el decodificador de video 30 puede realizar transformadas inversas en los bloques de coeficientes asociados con las TU de una CU actual para reconstruir los bloques residuales asociados con las TU de la CU actual. El decodificador de video 30 también reconstruye los bloques de codificación de la CU actual al agregar 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 una trama, el decodificador de video 30 puede reconstruir la trama.

Como se señaló anteriormente, la codificación de video logra la compresión de video usando principalmente dos modos, es decir, predicción intra-trama (o intrapredicción) y predicción inter-trama (o interpredicción). Se señala que la IBC se puede considerar como una predicción intra-trama o un tercer modo. Entre los dos modos, la predicción inter-trama contribuye más a la eficiencia de codificación que la predicción intra-trama debido al uso de vectores de movimiento para predecir un bloque de video actual a partir de un bloque de video de referencia.

Pero con la tecnología de captura de datos de video cada vez mejor y el tamaño de bloque de video más refinado para conservar detalles en los datos de video, la cantidad de datos requeridos para representar vectores de movimiento para una trama actual también incrementa sustancialmente. Una forma de superar este desafío es beneficiarse del hecho de que no solo un grupo de CU vecinas en los dominios espacial y temporal tienen datos de video similares para propósito de predecir, sino que los vectores de movimiento entre estas CU vecinas también son similares. Por lo tanto, es posible usar la información de movimiento de CU espacialmente vecinas y/o CU colocadas temporalmente como una aproximación de la información de movimiento (por ejemplo, vector de movimiento) de una CU actual al explorar su correlación espacial y temporal, que también se refiere como "predicador de vector de movimiento" (MVP) de la CU actual.

En lugar de codificar, en el flujo de bits de video, un vector de movimiento real de la CU actual determinado por la unidad de estimación de movimiento 42 como se describió anteriormente en relación con la figura 2, el predicador de vector de movimiento de la CU actual se resta del vector de movimiento real de la CU actual para producir una diferencia de vector de movimiento (MVD) para la CU actual. Al hacerlo, no hay necesidad de codificar el vector de movimiento determinado por la unidad de estimación de movimiento 42 para cada CU de una trama en el flujo de bits de video y la cantidad de datos usados para representar información de movimiento en el flujo de bits de video se puede disminuir significativamente.

Al igual que el proceso de elegir un bloque predictivo en una trama de referencia durante la predicción inter-trama de un bloque de código, se necesitan adoptar un conjunto de reglas tanto por el codificador de video 20 como el decodificador de video 30 para construir una lista de candidatos de vector de movimiento para una CU actual usando aquellos vectores de movimiento candidatos potenciales asociados con CU espacialmente vecinas y/o CU colocadas temporalmente de la CU actual y entonces seleccionar un miembro de la lista de candidatos de vector

de movimiento como un predictor de vector de movimiento para la CU actual. Al hacerlo, no hay necesidad de transmitir la propia lista de candidatos a vector de movimiento entre el codificador de video 20 y el decodificador de video 30 y un índice del predictor de vector de movimiento seleccionado dentro de la lista de candidatos de vector de movimiento es suficiente para que el codificador de video 20 y decodificador de video 30 usen el mismo predictor de vector de movimiento dentro de la lista de candidatos de vector de movimiento para codificar y decodificar la CU actual.

En algunas implementaciones, cada CU de interpredicción tiene tres modos de predicción de vector de movimiento que incluyen inter (que también se refiere como "predicción avanzada de vector de movimiento" (AMVP)), salto y fusión para construir la lista de candidatos de vector de movimiento. En cada modo, se pueden agregar uno o más candidatos de vector de movimiento a la lista de candidatos de vector de movimiento de acuerdo con los algoritmos descritos más adelante. En última instancia, uno de ellos en la lista de candidatos se usa como el mejor predictor de vector de movimiento de la CU de interpredicción que se va a codificar en el flujo de bits de video por el codificador de video 20 o se va a decodificar del flujo de bits de video por el decodificador de video 30. Para encontrar el mejor predictor de vector de movimiento de la lista de candidatos, se introduce un esquema de competencia de vector de movimiento (MVC) para seleccionar un vector de movimiento de un conjunto de candidatos dado de vectores de movimiento, es decir, la lista de candidatos de vector de movimiento, que incluye candidatos de vector de movimiento espacial y temporal.

Además de derivar candidatos de predictor de vector de movimiento de CU espacialmente vecinas o colocadas temporalmente, los candidatos de predictor de vector de movimiento también se pueden derivar de la llamada tabla de "predicción de vector de movimiento basada en historial" (HMVP). La tabla de HMVP aloja un número predefinido de predictores de vector de movimiento, cada uno que se ha usado para codificar/decodificar una CU particular de la misma fila de las CTU (o, algunas veces, la misma CTU). Debido a la proximidad espacial/temporal de estas CU, existe una alta probabilidad de que uno de los predictores de vector de movimiento en la tabla de HMVP se pueda reutilizar para codificar/decodificar diferentes CU dentro de la misma fila de las CTU. Por lo tanto, es posible lograr una mayor eficiencia de código al incluir la tabla de HMVP en el proceso de construcción de la lista de candidatos de vector de movimiento.

En algunas implementaciones, la tabla de HMVP tiene una longitud fija (por ejemplo, 5) y se gestiona de una manera casi de primero en entrar, primero en salir (FIFO). Por ejemplo, se reconstruye un vector de movimiento para una CU cuando se decodifica un bloque intercodificado de la CU. La tabla de HMVP se actualiza sobre la marcha con el vector de movimiento reconstruido debido a que el vector de movimiento puede ser el predictor de vector de movimiento de una CU posterior. Al actualizar la tabla de HMVP, hay dos escenarios: (i) el vector de movimiento reconstruido es diferente de otros vectores de movimiento existentes en la tabla de HMVP o (ii) el vector de movimiento reconstruido es el mismo que uno de los vectores de movimiento existentes en la tabla de HMVP. Para el primer escenario, el vector de movimiento reconstruido se agrega a la tabla de HMVP como el más nuevo si la tabla de HMVP no está llena. Si la tabla de HMVP ya está llena, el vector de movimiento más antiguo en la tabla de HMVP se necesita remover de la tabla de HMVP primero antes de que el vector de movimiento reconstruido se agregue como el más nuevo. En otras palabras, la tabla de HMVP en este caso es similar a una memoria intermedia FIFO y asociada con otro bloque previamente intercodificado se desplaza fuera de la memoria intermedia FIFO y el vector de movimiento reconstruido se agrega a la cola de la memoria intermedia FIFO como el miembro más nuevo en la tabla de HMVP. Para el segundo escenario, el vector de movimiento existente en la tabla de HMVP que es sustancialmente idéntico al vector de movimiento reconstruido se remueve de la tabla de HMVP antes de que el vector de movimiento reconstruido se agregue a la tabla de HMVP como el más nuevo. Si la tabla de HMVP también se mantiene en forma de una memoria intermedia FIFO, los predictores de vector de movimiento después del vector de movimiento idéntico en la tabla de HMVP se desplazan hacia adelante por un elemento para ocupar el espacio dejado por el vector de movimiento removido y el vector de movimiento reconstruido entonces se agrega a la cola de la memoria intermedia FIFO como el miembro más nuevo en la tabla de HMVP.

Los vectores de movimiento en la tabla de HMVP se pueden agregar a las listas de candidatos de vector de movimiento bajo diferentes modos de predicción tal como AMVP, fusión, salto, etc. Se ha encontrado que la información de movimiento de bloques previamente intercodificados almacenados en la tabla de HMVP incluso no adyacentes al bloque actual se puede utilizar para una predicción de vector de movimiento más eficiente.

Después de seleccionar un candidato de MVP dentro del conjunto de candidatos dado de vectores de movimiento para una CU actual, el codificador de video 20 puede generar uno o más elementos de sintaxis para el candidato de MVP correspondiente y codificarlos en el flujo de bits de video de modo que el decodificador de video 30 pueda recuperar el candidato de MVP del flujo de bits de video usando los elementos de sintaxis. Dependiendo del modo específico usado para construir el conjunto de candidatos de vector de movimiento, diferentes modos (por ejemplo, AMVP, fusión, salto, etc.) tienen diferentes conjuntos de elementos de sintaxis. Para el modo AMVP, los elementos de sintaxis incluyen indicadores de interpredicción (lista 0, lista 1 o predicción bidireccional), índices de referencia, índices de candidatos de vector de movimiento, señal residual de predicción de vector de movimiento, etc. Para el modo de salto y el modo de fusión, solo los índices de fusión se codifican en el flujo de bits debido a que la CU

actual hereda los otros elementos de sintaxis, incluidos los indicadores de interpredicción, índices de referencia y vectores de movimiento de una CU vecina referida por el índice de fusión codificado. En el caso de una CU codificada por salto, también se omite la señal residual de predicción de vector de movimiento.

5 La figura 5A es un diagrama de bloques que ilustra posiciones de bloque vecinas espacialmente y colocadas temporalmente de una CU actual que se va a codificar/decodificar de acuerdo con algunas implementaciones de la presente divulgación. Para un modo dado, se construye una lista de candidatos de predicción de vector de movimiento (MVP) al verificar primero la disponibilidad de vectores de movimiento asociados con las posiciones de bloque vecinas espacialmente izquierda y superior, y la disponibilidad de vectores de movimiento asociados con posiciones de bloque colocadas temporalmente y entonces los vectores de movimiento en la tabla de HMVP. Durante el proceso de construcción de la lista de candidatos de MVP, algunos candidatos de MVP redundantes se remueven de la lista de candidatos y, si es necesario, se agrega un vector de movimiento de valor cero para hacer que la lista de candidatos tenga una longitud fija (se señala que los diferentes modos pueden tener diferentes longitudes fijas). Después de la construcción de la lista de candidatos de MVP, el codificador de video 20 puede seleccionar el mejor predictor de vector de movimiento de la lista de candidatos y codificar el índice correspondiente que indica el candidato elegido en el flujo de bits de video.

Usando la figura 5A como ejemplo y suponiendo que la lista de candidatos tiene una longitud fija de dos, la lista de candidatos de predictor de vector de movimiento (MVP) para la CU actual se puede construir al realizar los siguientes pasos en orden bajo el modo AMVP:

- 1) Selección de candidatos de MVP de las CU espacialmente vecinas
  - a) Derivar hasta un candidato de MVP no escalado de una de las dos CU vecinas espaciales izquierdas que comienzan con A0 y terminan con A1;
  - 25 b) Si no hay un candidato de MVP no escalado desde la izquierda disponible en el paso anterior, derivar hasta un candidato de MVP escalado de una de las dos CU vecinas espaciales izquierdas que comienzan con A0 y terminan con A1;
  - c) Derivar hasta un candidato a MVP no escalado de una de las tres CU vecinas espaciales anteriores que comienzan con B0, entonces B1 y terminan con B2;
  - 30 d) Si ni A0 ni A1 están disponibles o si se codifican en modos intra, derivar hasta un candidato de MVP escalado de una de las tres CU vecinas espaciales anteriores que comienzan con B0, entonces B1 y terminan con B2;
- 2) Si se encuentran dos candidatos de MVP en los pasos anteriores y son idénticos, remover uno de los dos candidatos de la lista de candidatos de MVP;
- 35 3) Selección de candidatos de MVP de las CU colocadas temporalmente
  - a) Si la lista de candidatos de MVP después del paso anterior no incluye dos candidatos de MVP, derivar hasta un candidato de MVP de las CU colocadas temporalmente (por ejemplo, T0)
- 40 4) Selección de candidatos de MVP de la tabla de HMVP
  - a) Si la lista de candidatos de MVP después del paso anterior no incluye dos candidatos de MVP, derivar hasta dos MVP basados en historial de la tabla de HMVP; y
- 45 5) Si la lista de candidatos de MVP después del paso anterior no incluye dos candidatos de MVP, agregar hasta dos MVP de valor cero a la lista de candidatos de MVP.

Puesto que solo hay dos candidatos en la lista de candidatos de MVP en modo AMVP construida anteriormente, un elemento de sintaxis asociado como una bandera binaria se codifica en el flujo de bits para indicar cuál de los dos candidatos de MVP dentro de la lista de candidatos se usa para decodificar la CU actual.

En algunas implementaciones, la lista de candidatos de MVP para la CU actual bajo el modo de salto o fusión se puede construir al realizar un conjunto similar de pasos en orden como los anteriores. Se señala que un tipo especial de candidato de fusión llamado "candidato de fusión por pares" también se incluye en la lista de candidatos de MVP para el modo de salto o fusión. El candidato de fusión por pares se genera al promediar los MV de los dos candidatos de vector de movimiento de modo de fusión previamente derivados. El tamaño de la lista de candidatos de MVP de fusión (por ejemplo, de 1 a 6) se señala en un encabezado de segmento de la CU actual. Para cada CU en el modo de fusión, se codifica un índice del mejor candidato de fusión usando binarización unaria truncada (TU). La primera bandeja del índice de fusión se codifica con contexto y la codificación de omisión se usa para otras bandejas.

Como se mencionó anteriormente, los MVP basados en historial se pueden agregar a la lista de candidatos de MVP en modo AMVP o a la lista de candidatos de MVP de fusión después del MVP espacial y MVP temporal. La información de movimiento de una CU previamente intercodificada se almacena en la tabla de HMVP y se usa como un candidato de MVP para la CU actual. La tabla de HMVP se mantiene durante el proceso de codificación/decodificación. Siempre que haya una CU no de sub-bloque inter-codificada, la información de vector

de movimiento asociada se agrega a la última entrada de la tabla de HMVP como un nuevo candidato en tanto que la información de vector de movimiento almacenada en la primera entrada de la tabla de HMVP se remueve de la misma (si la tabla de HMVP ya está llena y no hay un duplicado idéntico de la información de vector de movimiento asociada en la tabla). De manera alternativa, el duplicado idéntico de la información de vector de movimiento asociada se remueve de la tabla antes de que la información de vector de movimiento asociada se agregue a la última entrada de la tabla de HMVP.

Como se señaló anteriormente, la copia intrabloque (IBC) puede mejorar significativamente la eficiencia de codificación de los materiales de contenido de pantalla. Puesto que el modo IBC se implementa como un modo de codificación de nivel de bloque, la coincidencia de bloques (BM) se realiza en el codificador de video 20 para encontrar un vector de bloque óptimo para cada CU. Aquí, se usa un vector de bloque para indicar el desplazamiento del bloque actual a un bloque de referencia, que ya se ha reconstruido dentro de la imagen actual. Una CU codificada por IBC se trata como el tercer modo de predicción diferente de los modos de intra o interpredicción.

A nivel de CU, el modo IBC se puede señalar como modo IBC AMVP o modo de salto/fusión IBC como sigue:

Modo IBC AMVP: una diferencia de vector de bloque (BVD) entre el vector de bloque real de una CU y un predictor de vector de bloque de la CU seleccionado de candidatos de vector de bloque de la CU se codifica de la misma manera que una diferencia de vector de movimiento se codifica bajo el modo AMVP descrito anteriormente. El método de predicción de vector de bloque usa dos candidatos de vector de bloque como predictores, uno del vecino izquierdo y el otro del vecino superior (si se codifica IBC). Cuando cualquiera de los vecinos no está disponible, se usará un vector de bloque predeterminado como predictor de vector de bloque. Se señala una bandera binaria para indicar el índice de predictor de vector de bloque. La lista de candidatos de IBC AMVP consiste en candidatos espaciales y de HMVP.

Modo de salto/fusión de IBC: se usa un índice de candidato de fusión para indicar cuál de los candidatos de vector de bloque en la lista de candidatos de fusión de bloques codificados de IBC vecinos se usa para predecir el vector de bloque para el bloque actual. La lista de candidatos de fusión de IBC consiste en candidatos espaciales, HMVP y por pares.

□tro enfoque para mejorar la eficiencia de codificación adoptado por la norma de codificación del estado de la técnica es introducir el procesamiento paralelo en el proceso de codificación/decodificación de video usando, por ejemplo, un procesador de múltiples núcleos. Por ejemplo, el procesamiento paralelo de frente de onda (WPP) ya se ha introducido en HEVC como una característica de codificación o decodificación de múltiples filas CTU en paralelo usando múltiples subprocesos.

La figura 5B es un diagrama de bloques que ilustra la codificación de múltiples subprocesos de múltiples filas de las CTU de una imagen usando procesamiento paralelo de frente de onda (WPP) de acuerdo con algunas implementaciones de la presente divulgación. Cuando WPP se habilita, es posible procesar múltiples filas de CTU en paralelo de una manera de frente de onda, donde puede haber un retraso de dos CTU entre el inicio de dos frentes de onda vecinos. Por ejemplo, para codificar la imagen 500 usando WPP, un codificador de video, tal como el codificador de video 20 y decodificador de video 30, puede dividir las unidades de árbol de codificación (CTU) de la imagen 500 en una pluralidad de frentes de onda, cada frente de onda corresponde a una fila respectiva de las CTU en la imagen. El codificador de video puede comenzar a codificar un frente de onda superior, por ejemplo, usando un primer núcleo o subproceso de codificador. Después de que el codificador de video ha codificado dos o más CTU del frente de onda superior, el codificador de video puede comenzar a codificar un segundo-a-superior frente de onda en paralelo con la codificación del frente de onda superior, por ejemplo, usando un segundo núcleo o subproceso de codificador paralelo. Después de que el codificador de video ha codificado dos o más CTU del segundo a superior frente de onda, el codificador de video puede comenzar a codificar un tercer a superior frente de onda en paralelo con la codificación de los frentes de onda más altos, por ejemplo, usando un tercer núcleo o subproceso de codificador paralelo. Este patrón puede continuar por los frentes de onda en la imagen 500. En la presente divulgación, un conjunto de las CTU que un codificador de video está codificando simultáneamente, usando WPP, se refiere como un "grupo de CTU". Por lo tanto, cuando el codificador de video usa WPP para codificar una imagen, cada CTU del grupo de CTU puede pertenecer a un frente de onda único de la imagen y la CTU se puede desplazar de una CTU en un frente de onda anterior respectivo por al menos dos columnas de las CTU de la imagen.

El codificador de video puede inicializar un contexto para un frente de onda actual para realizar codificación aritmética binaria adaptativa al contexto (CABAC) del frente de onda actual con base en datos de los primeros dos bloques del frente de onda anterior, así como uno o más elementos de un encabezado de segmento para un segmento que incluye el primer bloque de código del frente de onda actual. El codificador de video puede realizar la inicialización CABAC de un frente de onda posterior (o fila de CTU) usando los estados de contexto después de codificar dos CTU de una fila de CTU por arriba de la fila de CTU posterior. En otras palabras, antes de comenzar la codificación de un frente de onda actual, un codificador de video (o más específicamente, un subproceso del codificador de video) puede codificar al menos dos bloques de un frente de onda por arriba del frente de onda actual, suponiendo que el frente de onda actual no es la fila superior de las CTU de una imagen. El codificador de

video entonces puede inicializar un contexto CABAC para el frente de onda actual después de codificar al menos dos bloques de un frente de onda por arriba del frente de onda actual. En este ejemplo, cada fila de CTU de la imagen 500 es una división separada y tiene un subproceso asociado (subproceso 1 de WPP, subproceso 2 de WPP, ...) de modo que el número de filas de CTU en la imagen 500 se puede codificar en paralelo.

Debido a que la implementación actual de la tabla de HMVP usa una memoria intermedia de vector de movimiento global (MV) para almacenar vectores de movimiento previamente reconstruidos, esta tabla de HMVP no se puede implementar en el esquema de codificación en paralelo habilitado para WPP descrito anteriormente en relación con la figura 5B. En particular, el hecho de que la memoria intermedia de MV global se comparta por todos los subprocesos del proceso de codificación/decodificación de un codificador de video evita que los subprocesos de WPP después de que se inicie el primer subproceso de WPP (es decir, el subproceso 1 de WPP), puesto que estos subprocesos de WPP tienen que esperar a que se complete la actualización de la tabla de HMVP de la última CTU (es decir, la CTU más a la derecha) del primer subproceso de WPP (es decir, la primera fila de CTU).

Para superar el problema, se propone que la memoria intermedia de MV global compartida por los subprocesos de WPP se reemplace con múltiples memorias intermedias dedicadas a filas de CTU de modo que cada frente de onda de la fila de CTU tenga su propia memoria intermedia para almacenar una tabla de HMVP que corresponde a la fila de CTU que se procesa por un subproceso de WPP correspondiente cuando WPP se habilita en el codificador de video. Se señala que cada fila de CTU que tiene su propia tabla de HMVP es equivalente a restablecer la tabla de HMVP antes de codificar una primera CU de la fila de CTU. El restablecimiento de tabla de HMVP es para eliminar todos los vectores de movimiento en la tabla de HMVP que resultan de la codificación de otra fila de CTU. En una implementación, la operación de restablecimiento es para establecer el tamaño de los predictores de vector de movimiento disponibles en la tabla de HMVP para que sea cero. En aun otra implementación, las operaciones de restablecimiento pueden ser para establecer el índice de referencia de todas las entradas en la tabla de HMVP para que sea un valor no válido tal como -1. Al hacerlo, la construcción de la lista de candidatos de MVP para una CTU actual dentro de un frente de onda particular, independientemente de cuál de los tres modos, AMVP, fusión y salto, depende de una tabla de HMVP asociada con un subproceso de WPP que procesa el frente de onda particular. No hay interdependencia entre diferentes frentes de onda que no sea el retardo de dos CTU descrito anteriormente y la construcción de listas de candidatos de vector de movimiento asociadas con diferentes frentes de onda puede proceder en paralelo como el proceso de WPP representado en la figura 5B. En otras palabras, al comienzo del procesamiento de un frente de onda particular, la tabla de HMVP se restablece para que esté vacía sin afectar la codificación de otro frente de onda de CTU por otro subproceso de WPP. En algunos casos, la tabla de HMVP se puede restablecer para que esté vacía antes de la codificación de cada CTU individual. En este caso, los vectores de movimiento en la tabla de HMVP se limitan a una CTU particular y probablemente hay una mayor probabilidad de que un vector de movimiento dentro de la tabla de HMVP se seleccione como un vector de movimiento de una CU actual dentro de la CTU particular.

La figura 6 es un diagrama de flujo que ilustra un proceso de ejemplo por el cual un codificador de video tal como el codificador de video 20 o decodificador de video 30 implementa las técnicas de construcción de una lista de candidatos de predictor de vector de movimiento usando al menos la tabla HMVP de acuerdo con algunas implementaciones de la presente divulgación. Para propósitos ilustrativos, el diagrama de flujo representa un proceso de decodificación de video. En primer lugar, el decodificador de video 30 adquiere (610) un flujo de bits de video codificado que incluye datos asociados con múltiples imágenes codificadas. Como se ilustra en las figuras 4A y 4C, cada imagen incluye múltiples filas de unidades de árbol de codificación (CTU) y cada CTU incluye una o más unidades de codificación (CU). El decodificador de video 30 extrae diferentes piezas de información del flujo de bits de video, tal como elementos de sintaxis y valores de píxel, para reconstruir la imagen fila por fila.

Antes de decodificar una fila actual de las CTU, el decodificador de video 30 primero restablece (620) una tabla de predictor de vector de movimiento basado en historial (HMVP) para la fila actual de la CTU. Como se señaló anteriormente, el restablecimiento de la tabla de HMVP asegura que el decodificador de video 30 sea capaz de decodificar múltiples filas de CTU de la imagen actual en paralelo usando, por ejemplo, un proceso de múltiples subprocesos, un subproceso que tiene su propia tabla de HMVP por fila de CTU o un procesador de múltiples núcleos, un núcleo que tiene su propia tabla de HMVP por fila de CTU o ambos. En incluso algunas otras realizaciones, antes de decodificar una CTU actual, el decodificador de video 30 primero restablece (620) una tabla de predictor de vector de movimiento basado en historial (HMVP) para la CTU actual. Como se señaló anteriormente, el restablecimiento de la tabla de HMVP asegura que el decodificador de video 30 sea capaz de decodificar múltiples CTU de la imagen actual en paralelo usando, por ejemplo, un proceso de múltiples subprocesos, un subproceso que tiene su propia tabla de HMVP por CTU o un procesador de múltiples núcleos, un núcleo que tiene su propia tabla de HMVP por CTU o ambos.

En tanto que se decodifica la fila actual de las CTU (630), el decodificador de video (30) mantiene (630-1) una pluralidad de predictores de vector de movimiento en la tabla de HMVP. Como se señaló anteriormente, cada predictor de vector de movimiento almacenado en la tabla de HMVP se ha usado para decodificar al menos otra CU dentro de la fila actual de las CTU. El hecho de que el predictor de vector de movimiento esté presente en la tabla de HMVP se debe a que se puede usar nuevamente para predecir otra CU dentro de la fila actual de CTU cuando la tabla de HMVP participa en el proceso de construcción de la lista de candidatos de vector de movimiento

como se describió anteriormente.

Para una CU actual de la fila actual de las CTU, el decodificador de video 30 primero extrae (630-3) un modo de predicción del flujo de bits de video. Como se señaló anteriormente, una CU puede tener múltiples tipos de modos de predicción, incluido el modo de predicción de vector de movimiento avanzado (AMVP), un modo de fusión, un modo de salto, un modo IBC AMVP y un modo de fusión de IBC. Una vez que el codificador de video 20 elige un modo de predicción apropiado para la CU, el modo de predicción elegido se señala en el flujo de bits. Como se señaló anteriormente, hay diferentes conjuntos de pasos ejecutados en diferentes órdenes para construir una lista de candidatos de vector de movimiento. Aquí, el decodificador de video 30 construye (630-5) una lista de candidatos de vector de movimiento de acuerdo con el modo de predicción y con base, al menos en parte, en la pluralidad de predictores de vector de movimiento en la tabla de HMVP. Después de las fuentes a partir de las cuales la lista de candidatos de vector de movimiento incluye predictores de vector de movimiento de CU espacialmente vecinas y/o CU colocadas temporalmente de la CU actual (cuando el modo de predicción es uno del modo AMVP, el modo IBC AMVP y el modo de fusión de IBC) y opcionalmente predictores de vector de movimiento por pares (cuando el modo de predicción es uno del modo de fusión y el modo de salto). Opcionalmente, cuando la lista de candidatos a vector de movimiento no alcanza una longitud predefinida, se pueden agregar uno o más predictores de vector de movimiento de valor cero a la lista de candidatos a vector de movimiento.

Después, el decodificador de video 30 selecciona (630-7), de la lista de candidatos de vector de movimiento, un predictor de vector de movimiento para la CU actual y determina (630-9) un vector de movimiento con base, al menos en parte, en el modo de predicción y el predictor de vector de movimiento seleccionado. Como se señaló anteriormente, dependiendo de si el modo de predicción es modo AMVP o no, el predictor de vector de movimiento seleccionado puede o no ser el vector de movimiento estimado para la CU actual. Por ejemplo, si el modo de predicción es el modo AMVP, el vector de movimiento estimado se determina al agregar una diferencia del vector de movimiento recuperada del flujo de bits al predictor del vector de movimiento seleccionado y la CU actual se decodifica entonces usando, al menos parcialmente, el vector de movimiento estimado y una CU correspondiente dentro de una imagen de referencia. Pero si el modo de predicción es el modo de fusión o salto, el predictor de vector de movimiento seleccionado ya es el vector de movimiento estimado, que se puede usar para decodificar la CU actual junto con la CU correspondiente dentro de la imagen de referencia. Finalmente, el decodificador de video 30 actualiza (630-11) la tabla de HMVP con base en el vector de movimiento determinado. Como se señaló anteriormente, cada miembro en la tabla de HMVP se ha usado para decodificar al menos otra CU antes y se mantiene en la tabla de HMVP para construir la lista de candidatos de vector de movimiento hasta que se remueve de la tabla de HMVP ya sea por un restablecimiento de tabla o por la inserción de un vector de movimiento usado para decodificar otra CU posterior dentro de la fila actual de las CTU.

En algunas implementaciones, la inserción de un vector de movimiento en la tabla de HMVP tiene dos escenarios posibles con base en un resultado de comparación entre el vector de movimiento determinado para la CU actual y la pluralidad de predictores de vector de movimiento en la tabla de HMVP. Si ninguno de la pluralidad de predictores de vector de movimiento en la tabla de HMVP es idéntico al vector de movimiento determinado, un predictor de vector de movimiento más antiguo o más antiguo se remueve de la tabla de HMVP cuando la tabla de HMVP está llena y el vector de movimiento se agrega a la tabla como el más nuevo. Si uno de la pluralidad de predictores de vector de movimiento en la tabla de HMVP es idéntico al vector de movimiento, el predictor de vector de movimiento idéntico entonces se remueve de la tabla de HMVP y todos los demás predictores de vector de movimiento después de que el predictor de vector de movimiento removido se mueve hacia adelante en la tabla de HMVP de modo que el vector de movimiento se agrega al final de la tabla de HMVP como el más nuevo.

Como se señaló anteriormente, dos o más de las múltiples filas de las CTU se pueden codificar/decodificar en paralelo, por ejemplo, usando WPP, cada fila de las CTU tiene una tabla de HMVP asociada para almacenar una pluralidad de predictores de vector de movimiento basados en historial usados para codificar/decodificar la fila correspondiente de las CTU. Por ejemplo, se asigna un subproceso a la decodificación de una fila particular de las CTU de la imagen actual que se está decodificando, de modo que las diferentes filas de las CTU tengan diferentes subprocesos asociados y se puedan decodificar en paralelo, como se describió anteriormente en relación con la figura 5B. En algunos ejemplos, el decodificador de video 30 identifica uno o más predictores de vector de movimiento dentro de la lista de candidatos de vector de movimiento como redundantes y los remueve de la lista de candidatos de vector de movimiento para mejorar aún más la eficiencia de codificación.

En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar o transmitir como una o más instrucciones o código, un medio leíble por computadora y ejecutarse por una unidad de procesamiento basada en hardware. Los medios leíbles por computadora pueden incluir medios de almacenamiento leíbles por computadora, que corresponden a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa de computadora de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios leíbles por computadora en general pueden corresponder a (1) medios de almacenamiento leíbles por computadora tangibles que no son transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible

al que se puede tener acceso por una o más computadoras o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementación de las implementaciones descritas en la presente solicitud. Un producto de programa de computadora puede incluir un medio leíble por computadora.

- 5 La terminología usada en la descripción de las implementaciones en la presente es para el propósito de describir implementaciones particulares solamente y no se propone que limite el alcance de las reivindicaciones. Como se usa en la descripción de las implementaciones y las reivindicaciones adjuntas, se propone que las formas singulares "un", "una", "el" y "la" incluyan también las formas plurales, a menos que el contexto indique claramente lo contrario. También se entenderá que el término "y/o" como se usa en la presente se refiere y abarca todas y cada una de las combinaciones posibles de uno o más de los elementos listados asociados. Se entenderá además que los términos "comprende" y/o "que comprende", cuando se usan en la esta especificación, especifican la presencia de características, elementos y/o componentes señalados, pero no excluyen la presencia o adición de una o más características, elementos, componentes y/o grupos de los mismos.
- 10
- 15 Se entenderá que aunque los términos primero, segundo, etc., se pueden usar en la presente para describir diferentes elementos, estos elementos no se deben limitar por estos términos. Estos términos sólo se usan para distinguir un elemento de otro. Por ejemplo, un primer electrodo se puede denominar un segundo electrodo, y, de manera similar, un segundo electrodo se puede denominar un primer electrodo, sin apartarse del alcance de las implementaciones. El primer electrodo y el segundo electrodo son ambos electrodos, pero no son el mismo electrodo.
- 20

La descripción de la presente solicitud se ha presentado para propósitos de ilustración y descripción, y no se propone que sea exhaustiva o limitada a la invención en la forma divulgada. Muchas modificaciones, variaciones e implementaciones alternativas serán evidentes para aquellos expertos en la técnica que tienen el beneficio de las enseñanzas presentadas en las descripciones anteriores y los dibujos asociados. La realización se eligió y describió a fin de explicar mejor los principios de la invención, la aplicación práctica y permitir que otros expertos en la técnica entiendan la invención para diferentes implementaciones y utilicen mejor los principios subyacentes y diferentes implementaciones con diferentes modificaciones que sean adecuadas para el uso particular contemplado. El alcance de la invención se define por las reivindicaciones anexas.

25

REIVINDICACIONES

1. Un método de decodificación de datos de video, el método que comprende:

5 adquirir (610) un flujo de bits de video que incluye datos asociados con múltiples imágenes codificadas, cada imagen incluye múltiples filas de unidades de árbol de codificación, CTU, y cada CTU que incluye una o más unidades de codificación, CU;  
 10 restablecer (620) una tabla de predictores de vector de movimiento basados en historial, HMVP, antes de decodificar una primera CTU de cada fila de las CTU de una imagen actual que se está decodificando para admitir la decodificación en paralelo de múltiples filas de las CTU de la imagen actual, donde el restablecimiento de la tabla de HMVP comprende: establecer un tamaño de predictores de vector de movimiento disponibles en la tabla de HMVP para que sea cero;  
 15 en tanto que se decodifica (630) una fila actual de las CTU de la imagen actual:  
 mantener (630-1) una pluralidad de predictores de vector de movimiento en la tabla de HMVP, cada predictor de vector de movimiento en la tabla de HMVP que se ha usado para decodificar al menos una CU de la fila actual de las CTU;  
 20 para una CU actual de la fila actual de las CTU que se van a decodificar:  
 extraer (630-3) un modo de predicción del flujo de bits de video;  
 construir (630-5) una lista de candidatos de vector de movimiento de acuerdo con el modo de predicción y con base, al menos en parte, en la pluralidad de predictores de vector de movimiento en la tabla de HMVP;  
 25 seleccionar (630-7), de la lista de candidatos a vector de movimiento, un predictor de vector de movimiento;  
 determinar (630-9) un vector de movimiento con base, al menos en parte, en el modo de predicción y el predictor del vector de movimiento seleccionado para decodificar la CU actual; y  
 actualizar (630-11) la tabla de HMVP con base en el vector de movimiento determinado, donde la actualización de la tabla de HMVP con base en el vector de movimiento determinado comprende:  
 30 comparar la pluralidad de predictores de vector de movimiento en la tabla de HMVP con el vector de movimiento determinado;  
 de acuerdo con una determinación de que ninguno de la pluralidad de predictores de vector de movimiento en la tabla de HMVP es idéntico al vector de movimiento determinado:  
 35 remover un predictor de vector de movimiento más temprano de la primera entrada de la tabla de HMVP cuando la tabla de HMVP está llena; y  
 agregar el vector de movimiento determinado como el más nuevo a la última entrada de la tabla de HMVP;  
 de acuerdo con una determinación de que uno de la pluralidad de predictores de vector de movimiento en la tabla de HMVP es idéntico al vector de movimiento determinado:  
 40 remover el único predictor de vector de movimiento idéntico de la tabla de HMVP;  
 mover cada uno de los predictores de vector de movimiento después del predictor de vector de movimiento removido hacia adelante en la tabla de HMVP; y  
 45 agregar el vector de movimiento determinado como el más nuevo a la última entrada de la tabla de HMVP.

40 2. El método de la reivindicación 1, donde el modo de predicción es uno seleccionado de un grupo que consiste en un modo de predicción de vector de movimiento avanzado, AMVP, y un modo de fusión.

45 3. El método de la reivindicación 2, donde el modo de predicción es el modo AMVP, y la decodificación de la CU actual comprende además:

50 recuperar, del flujo de bits de video, una diferencia del vector de movimiento para la CU actual;  
 agregar la diferencia de vector de movimiento y el predictor de vector de movimiento seleccionado conjuntamente como el vector de movimiento determinado; y  
 decodificar la CU actual usando el vector de movimiento determinado y una CU correspondiente dentro de una imagen de referencia.

55 4. El método de la reivindicación 2, donde el modo de predicción es el modo AMVP, y la construcción de la lista de candidatos de vector de movimiento comprende además:

60 agregar cero o más predictores de vector de movimiento de las CU espacialmente vecinas y/o CU colocadas temporalmente de la CU actual a la lista de candidatos de vector de movimiento;  
 agregar cero o más predictores de vector de movimiento basados en historial de la tabla de HMVP a la lista de candidatos de vector de movimiento hasta que una longitud actual de la lista de candidatos de vector de movimiento alcance un primer umbral predefinido; y  
 65 agregar cero o más predictores de vector de movimiento de valor cero a la lista de candidatos de vector de movimiento hasta que la longitud actual de la lista de candidatos de vector de movimiento sea igual al primer umbral predefinido.

5. El método de la reivindicación 2, donde el modo de predicción es el modo de fusión, y la decodificación de la CU actual comprende además:

usar el predictor de vector de movimiento seleccionado como el vector de movimiento determinado; y decodificar la CU actual usando el vector de movimiento determinado y una CU correspondiente dentro de una imagen de referencia.

5 6. El método de la reivindicación 1, donde, cuando dos o más filas de las CTU de la imagen actual que se está decodificando se decodifican en paralelo, cada fila de las CTU tiene una tabla de HMVP asociada para almacenar una pluralidad de predictores de vector de movimiento basados en historial usados para decodificar la fila correspondiente de las CTU.

10 7. El método de la reivindicación 1, que comprende además:

asignar un subproceso a la decodificación de la fila actual de las CTU de la imagen actual que se está decodificando, de modo que las diferentes filas de las CTU de la imagen actual que se está decodificando tengan diferentes subprocesos asociados.

15 8. Un dispositivo informático que comprende:

uno o más procesadores; memoria acoplada al uno o más procesadores; y una pluralidad de programas almacenados en la memoria que, cuando se ejecutan por uno o más procesadores, provocan que el dispositivo informático realice operaciones que incluyen:

20 adquirir (610) un flujo de bits de video que incluye datos asociados con múltiples imágenes codificadas, cada imagen incluye múltiples filas de unidades de árbol de codificación, CTU, y cada CTU que incluye una o más unidades de codificación, CU;

25 restablecer (620) una tabla de predictores de vector de movimiento basados en historial, HMVP, antes de decodificar una primera CTU de cada fila de las CTU de una imagen actual que se está decodificando para admitir la decodificación en paralelo de múltiples filas de las CTU de la imagen actual, donde el restablecimiento de la tabla de HMVP comprende: establecer un tamaño de predictores de vector de movimiento disponibles en la tabla de HMVP para que sea cero;

en tanto que se decodifica (630) una fila actual de las CTU de la imagen actual:

30 mantener (630-1) una pluralidad de predictores de vector de movimiento en la tabla de HMVP, cada predictor de vector de movimiento en la tabla de HMVP que se ha usado para decodificar al menos una CU de la fila actual de las CTU;

para una CU actual de la fila actual de las CTU que se van a decodificar;

35 extraer (630-3) un modo de predicción del flujo de bits de video;

construir (630-5) una lista de candidatos de vector de movimiento de acuerdo con el modo de predicción y con base, al menos en parte, en la pluralidad de predictores de vector de movimiento en la tabla de HMVP;

seleccionar (630-7), de la lista de candidatos a vector de movimiento, un predictor de vector de movimiento;

40 determinar (630-9) un vector de movimiento con base, al menos en parte, en el modo de predicción y el predictor del vector de movimiento seleccionado para decodificar la CU actual; y

45 actualizar (630-11) la tabla de HMVP con base en el vector de movimiento determinado, donde la actualización de la tabla de HMVP con base en el vector de movimiento determinado comprende:

comparar la pluralidad de predictores de vector de movimiento en la tabla de HMVP con el vector de movimiento determinado;

de acuerdo con una determinación de que ninguno de la pluralidad de predictores de vector de movimiento en la tabla de HMVP es idéntico al vector de movimiento determinado:

50 remover un predictor de vector de movimiento más temprano de la primera entrada de la tabla de HMVP cuando la tabla de HMVP está llena; y

agregar el vector de movimiento determinado como el más nuevo a la última entrada de la tabla de HMVP;

de acuerdo con una determinación de que uno de la pluralidad de predictores de vector de movimiento en la tabla de HMVP es idéntico al vector de movimiento determinado:

55 remover el único predictor de vector de movimiento idéntico de la tabla de HMVP;

mover cada uno de los predictores de vector de movimiento después del predictor de vector de movimiento removido hacia adelante en la tabla de HMVP; y

agregar el vector de movimiento determinado como el más nuevo a la última entrada de la tabla de HMVP.

9. El dispositivo informático de la reivindicación 8, donde el modo de predicción es uno seleccionado de un grupo que consiste en un modo de predicción de vector de movimiento avanzado, AMVP, y un modo de fusión.

60 10. Un medio de almacenamiento leible por computadora no transitorio que almacena una pluralidad de programas para ejecución por un dispositivo informático que tiene uno o más procesadores, donde la pluralidad de programas, cuando se ejecutan por uno o más procesadores, provocan que el dispositivo informático realice operaciones que incluyen:

65 adquirir (610) un flujo de bits de video que incluye datos asociados con múltiples imágenes codificadas, cada imagen incluye múltiples filas de unidades de árbol de codificación, CTU, y cada CTU que incluye una o más unidades de codificación, CU;

- restablecer (620) una tabla de predictores de vector de movimiento basados en historial, HMVP, antes de decodificar una primera CTU de cada fila de las CTU de una imagen actual que se está decodificando para admitir la decodificación en paralelo de múltiples filas de las CTU de la imagen actual, donde el restablecimiento de la tabla de HMVP comprende: establecer un tamaño de predictores de vector de movimiento disponibles en la tabla de HMVP para que sea cero;
- 5 en tanto que se decodifica (630) una fila actual de las CTU de la imagen actual:
- mantener (630-1) una pluralidad de predictores de vector de movimiento en la tabla de HMVP, cada predictor de vector de movimiento en la tabla de HMVP que se ha usado para decodificar al menos una CU de la fila actual de las CTU;
- 10 para una CU actual de la fila actual de las CTU que se van a decodificar:
- extraer (630-3) un modo de predicción del flujo de bits de video;
- construir (630-5) una lista de candidatos de vector de movimiento de acuerdo con el modo de predicción y con base, al menos en parte, en la pluralidad de predictores de vector de movimiento en la tabla de HMVP;
- 15 seleccionar (630-7), de la lista de candidatos a vector de movimiento, un predictor de vector de movimiento;
- determinar (630-9) un vector de movimiento con base, al menos en parte, en el modo de predicción y el predictor del vector de movimiento seleccionado para decodificar la CU actual; y
- actualizar (630-11) la tabla de HMVP con base en el vector de movimiento determinado, donde la actualización de la tabla de HMVP con base en el vector de movimiento determinado comprende:
- 20 comparar la pluralidad de predictores de vector de movimiento en la tabla de HMVP con el vector de movimiento determinado;
- de acuerdo con una determinación de que ninguno de la pluralidad de predictores de vector de movimiento en la tabla de HMVP es idéntico al vector de movimiento determinado:
- remover un predictor de vector de movimiento más temprano de la primera entrada de la tabla de HMVP cuando la tabla de HMVP está llena; y
- 25 agregar el vector de movimiento determinado como el más nuevo a la última entrada de la tabla de HMVP;
- de acuerdo con una determinación de que uno de la pluralidad de predictores de vector de movimiento en la tabla de HMVP es idéntico al vector de movimiento determinado:
- remover el único predictor de vector de movimiento idéntico de la tabla de HMVP;
- 30 mover cada uno de los predictores de vector de movimiento después del predictor de vector de movimiento removido hacia adelante en la tabla de HMVP; y
- agregar el vector de movimiento determinado como el más nuevo a la última entrada de la tabla de HMVP.
11. Un producto de programa de computadora que comprende una pluralidad de programas para ejecución por un dispositivo informático que tiene uno o más procesadores, donde la pluralidad de programas, cuando se ejecutan por uno o más procesadores, provocan que el dispositivo informático realice los pasos del método de cualquiera de las reivindicaciones 1 a 7.
- 35

DIBUJOS

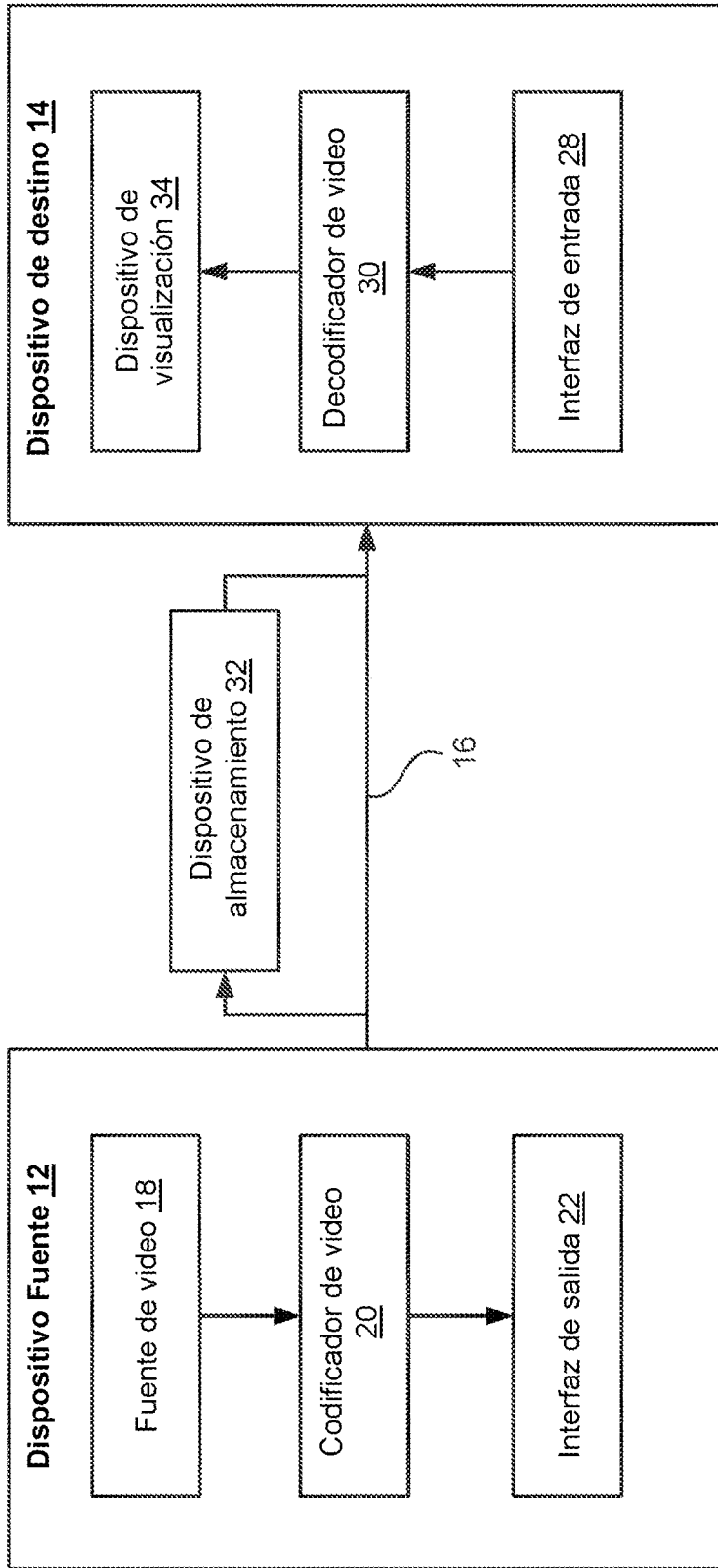


Figura 1

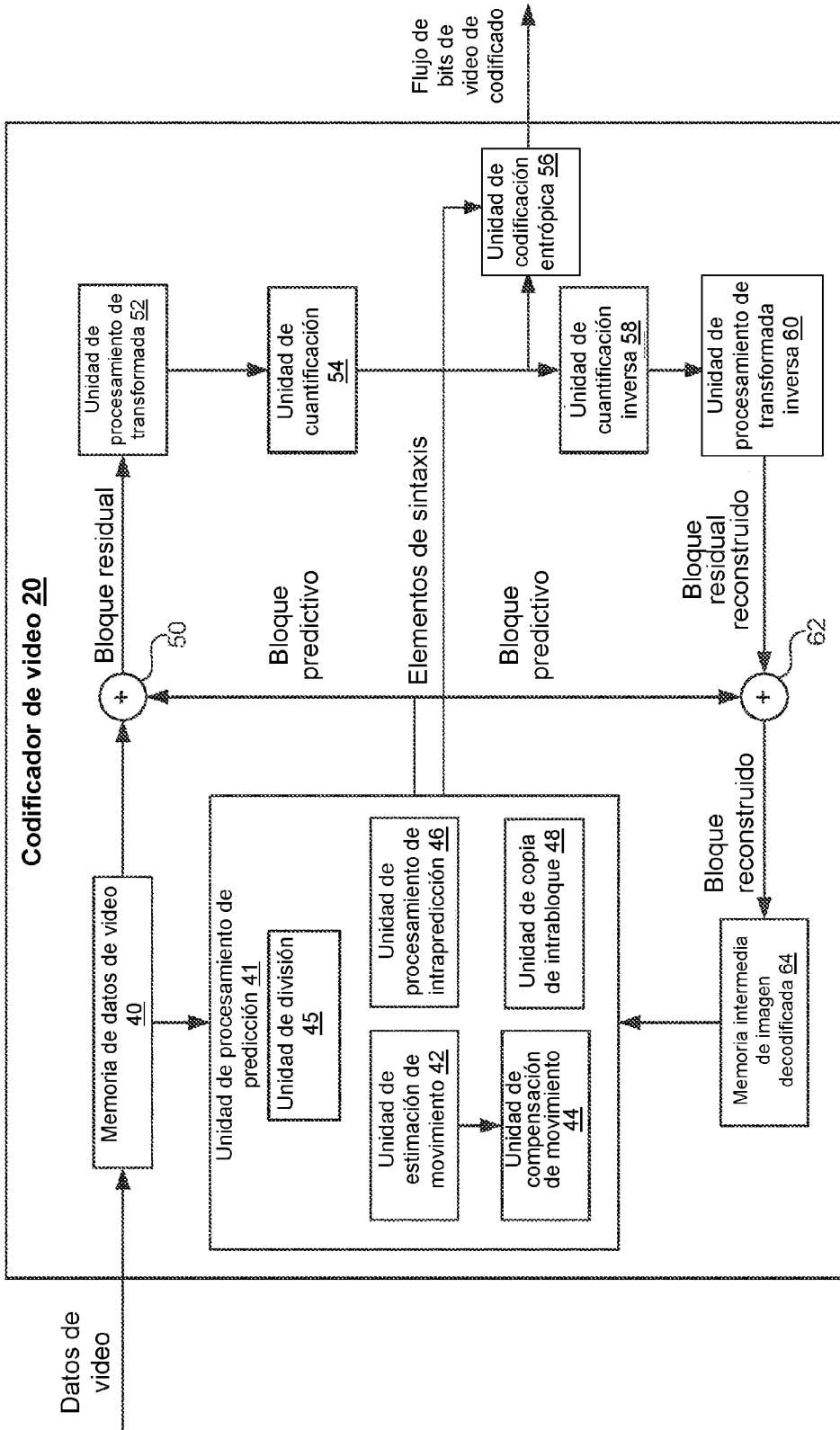


Figura 2

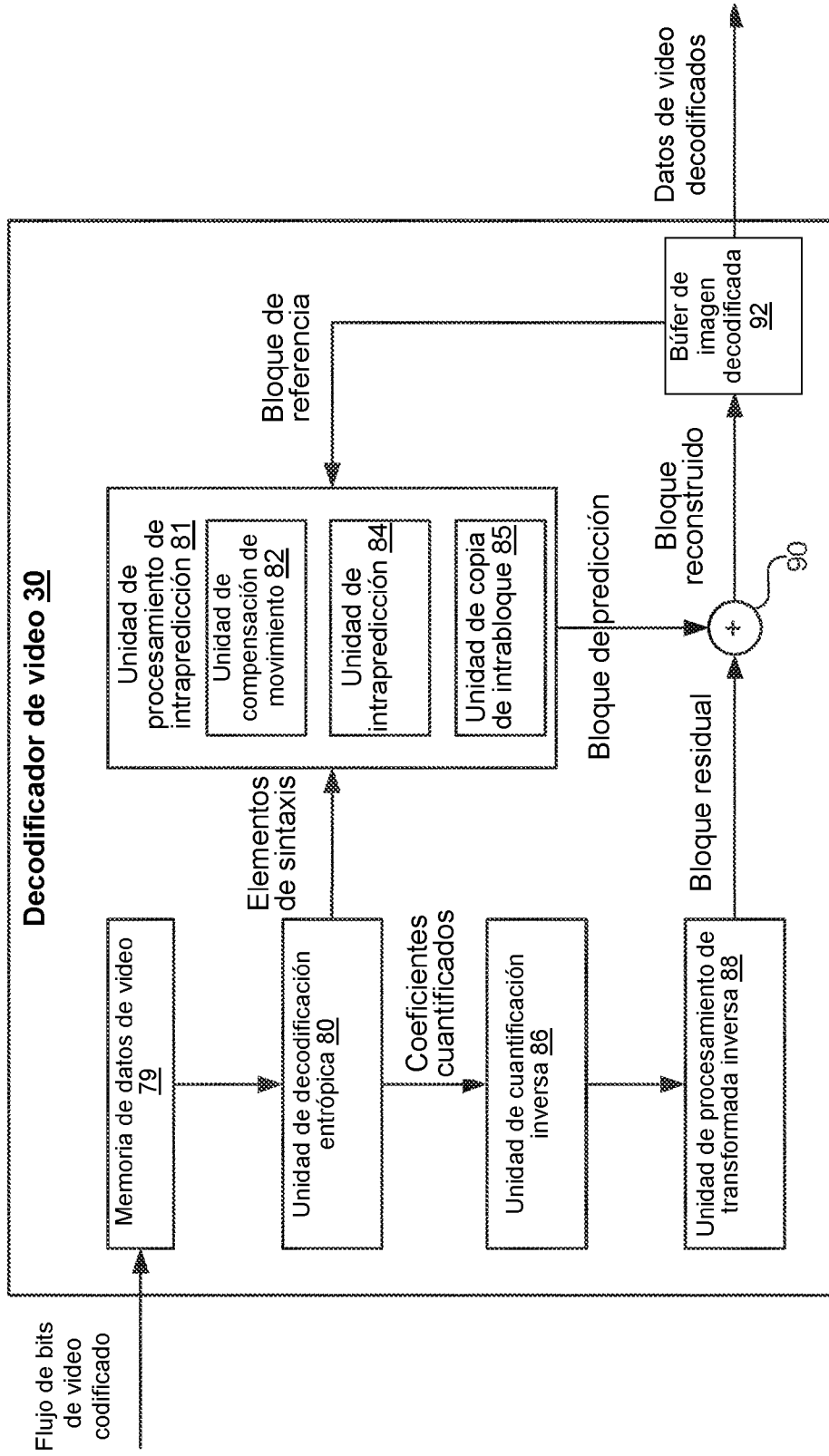


Figura 3

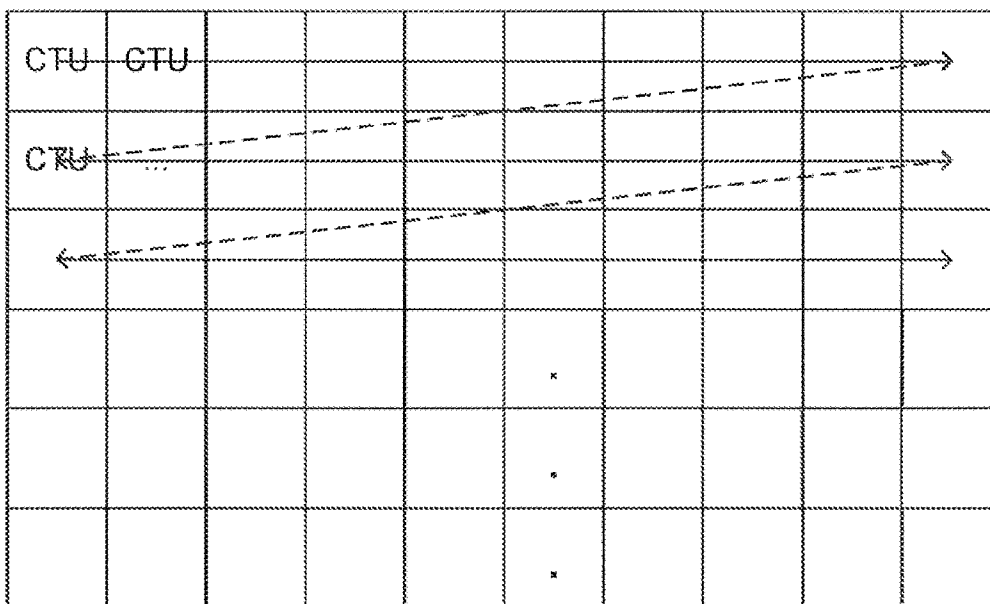


Figura 4A

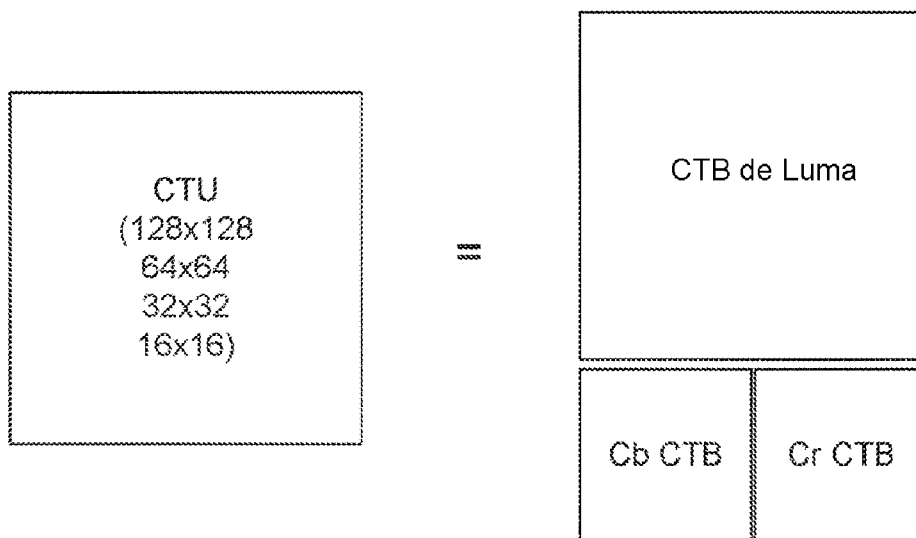


Figura 4B

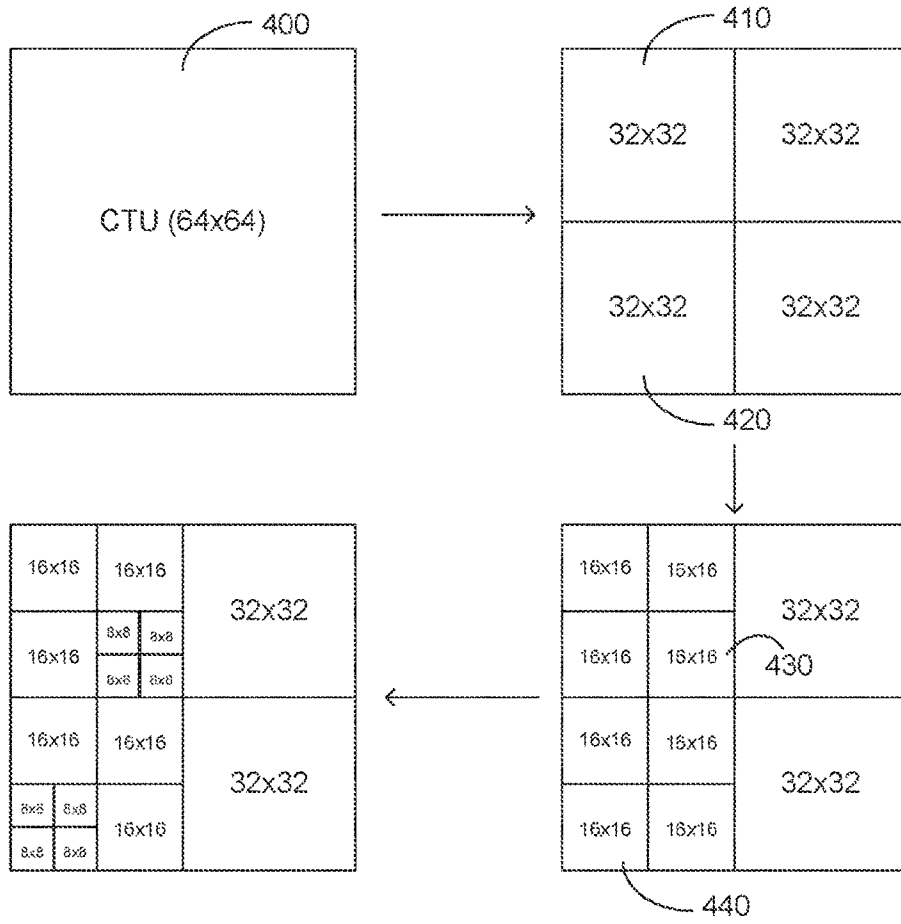


Figura 4C

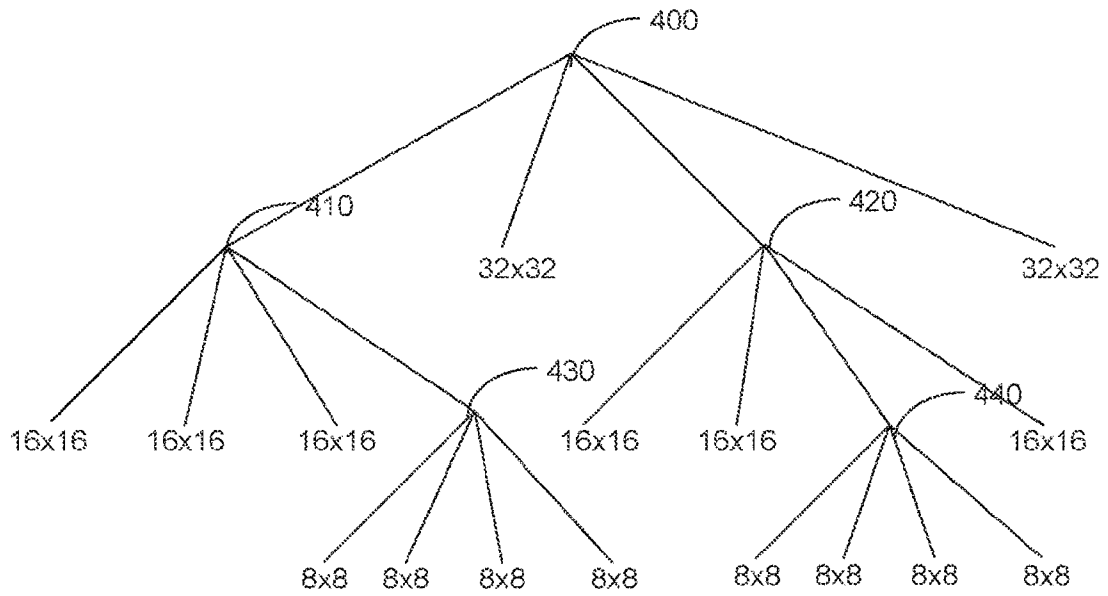


Figura 4D

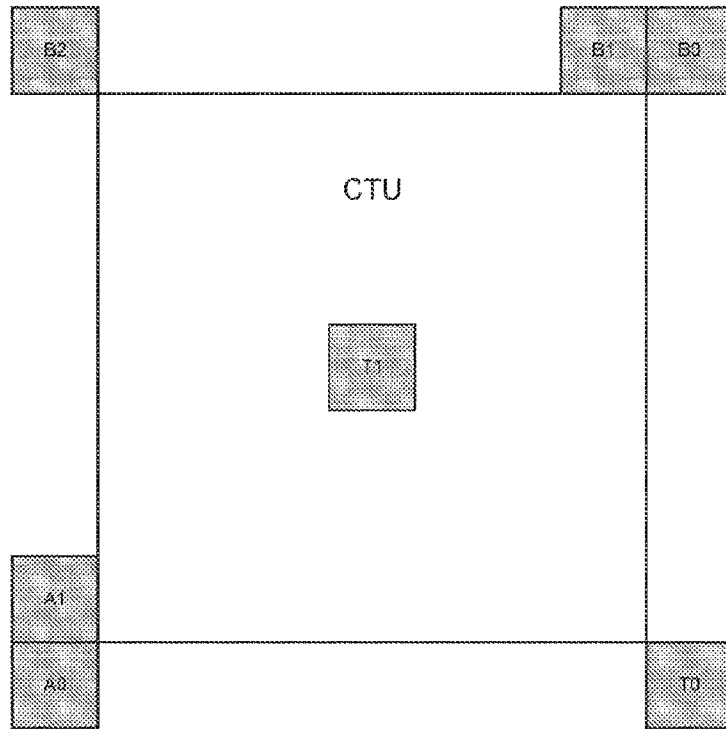


Figura 5A

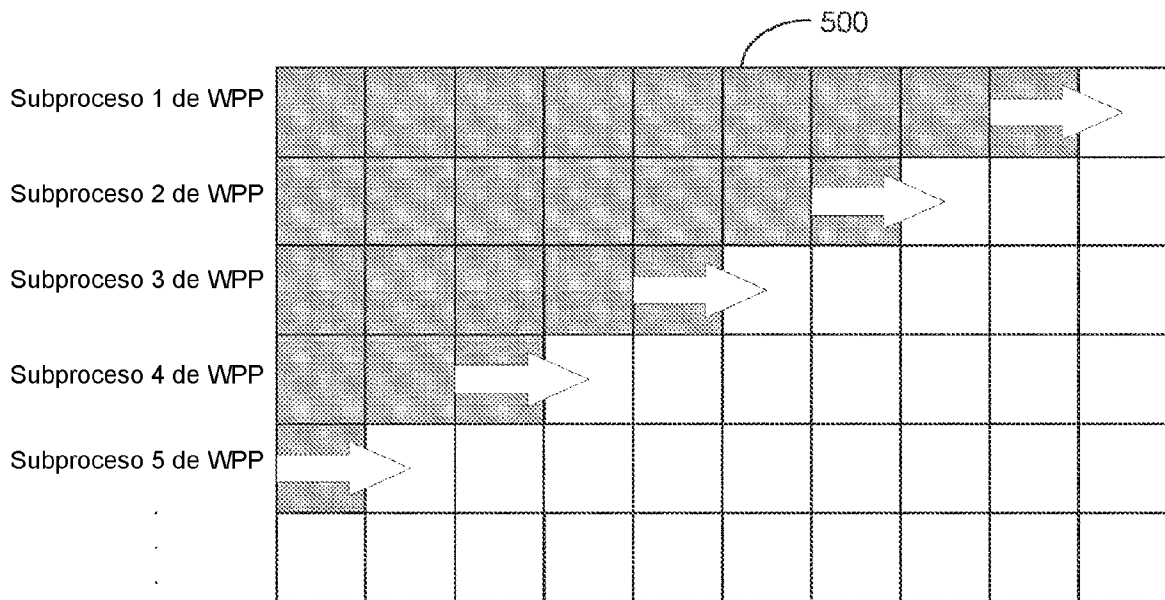


Figura 5B

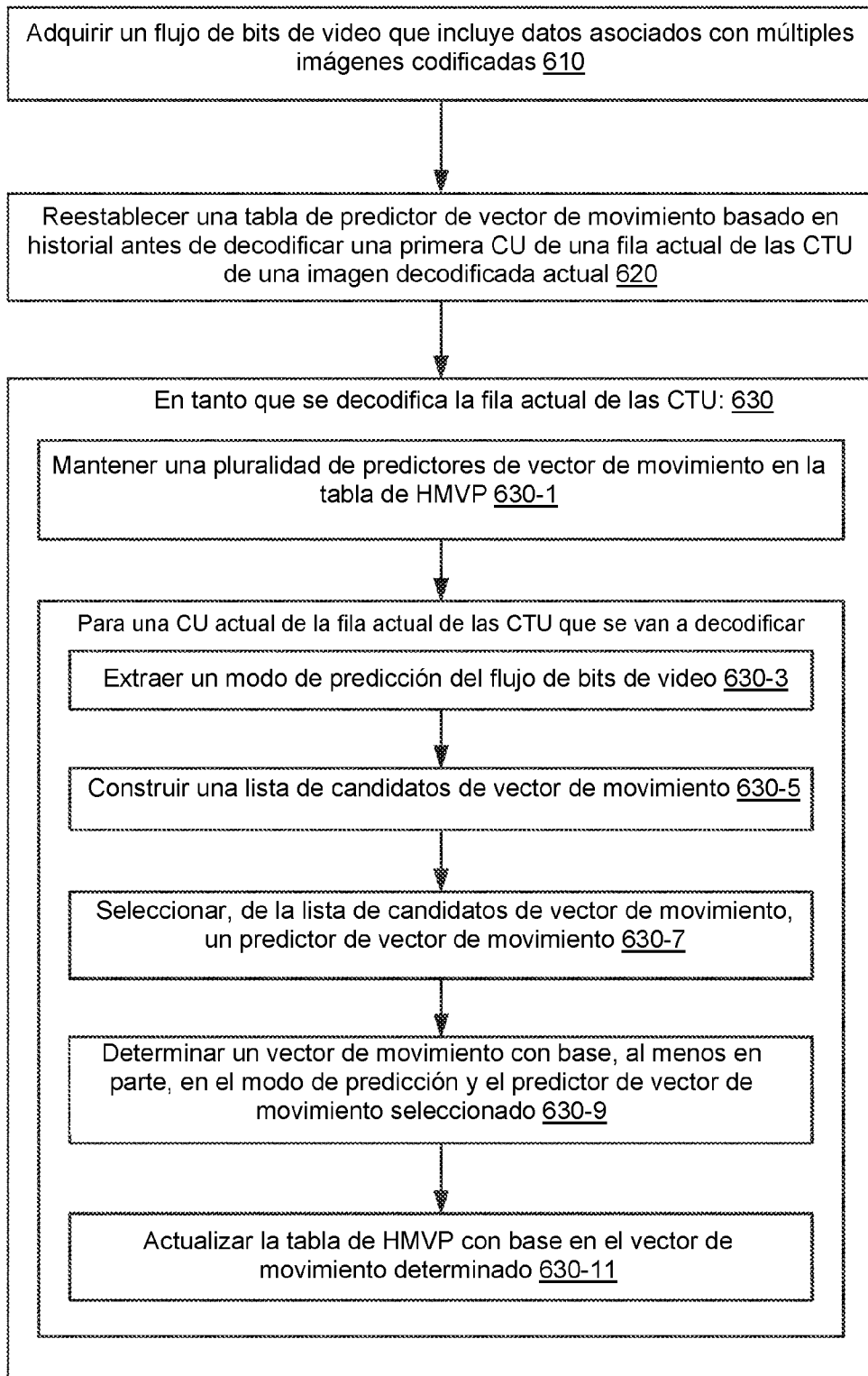


Figura 6