

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 588 314**

51 Int. Cl.:

H04N 19/33 (2014.01)

H04N 19/423 (2014.01)

H04N 19/436 (2014.01)

H04N 19/597 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **03.01.2014 PCT/US2014/010190**

87 Fecha y número de publicación internacional: **10.07.2014 WO14107583**

96 Fecha de presentación y número de la solicitud europea: **03.01.2014 E 14702663 (7)**

97 Fecha y número de publicación de la concesión europea: **25.05.2016 EP 2941876**

54 Título: **Gestión de memoria intermedia de imágenes decodificadas de múltiple resolución para codificación de vídeo multi-capa**

30 Prioridad:

04.01.2013 US 201361749105 P
02.01.2014 US 201414146507

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

02.11.2016

73 Titular/es:

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

72 Inventor/es:

CHEN, YING y
WANG, YE-KUI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 588 314 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

Gestión de memoria intermedia de imágenes decodificadas de múltiple resolución para codificación de vídeo multi-capa

5

CAMPO TÉCNICO

Esta divulgación se refiere a técnicas para codificación de vídeo y, más específicamente, a técnicas para codificar vídeo multi-capa.

10

ANTECEDENTES

Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas digitales de difusión directa, sistemas de difusión inalámbricos, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digitales, reproductores de multimedios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, dispositivos de videoconferencia, etc. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), el estándar de codificación de vídeo de alta eficacia (HEVC) en desarrollo actualmente, y extensiones de dichos estándares, para transmitir, recibir y almacenar información de vídeo digital más eficazmente.

15

20

El documento WO 2008/084443 describe diversos modos de decodificar datos de vídeo para producir componentes de capa de decodificación.

25

Las extensiones de algunos de los estándares que se han mencionado anteriormente, incluyendo la H.264/AVC, proporcionan técnicas para codificación de vídeo multi-capa. Las técnicas de codificación de vídeo multi-capa pueden incluir técnicas de codificación de vídeo ajustables a escala, técnicas de codificación de vídeo multi-vista, y técnicas de codificación de vídeo multi-vista más profundidad. Para producir vídeo en estéreo o tridimensional ("3D"), por ejemplo, se han propuesto técnicas para codificación multi-vista para su uso en la AVC, con el estándar de codificación de vídeo ajustable a escala (SVC) (que es la extensión ajustable a escala para la H.264/AVC), y el estándar de codificación de vídeo multi-vista (MVC) (que se ha convertido en la extensión multi-vista para la H.264/AVC).

30

35

Habitualmente, se consigue vídeo en estéreo usando dos vistas, por ejemplo, una vista izquierda y una vista derecha. Una imagen de la vista izquierda puede mostrarse esencialmente de forma simultánea con una imagen de la vista derecha para conseguir un efecto de vídeo tridimensional. Por ejemplo, un usuario puede llevar gafas pasivas polarizadas que filtran la vista izquierda de la vista derecha. Como alternativa, las imágenes de las dos vistas pueden mostrarse en sucesión rápida, y el usuario puede llevar gafas activas que obturen rápidamente el ojo izquierdo y derecho a la misma frecuencia, pero con un desplazamiento de fase de 90 grados.

40

RESUMEN

En general, esta divulgación describe técnicas para codificación de vídeo multicapa. En particular, esta divulgación se refiere a la gestión de la memoria intermedia de imágenes decodificadas (DPB) para codificación de vídeo multicapa.

45

En uno o más ejemplos de la divulgación, puede configurarse un codificador de vídeo (por ejemplo, un codificador de vídeo o un decodificador de vídeo) para realizar los siguientes procedimientos.

50

En un ejemplo de la divulgación, un procedimiento de codificación de datos de vídeo comprende decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada, almacenar los componentes de capa decodificada en una o más sub-unidades de una DPB y realizar un proceso de gestión de la DPB en las una o más sub-unidades, en el que el proceso de gestión de la DPB se gestiona por separado para cada una entre las una o más sub-unidades.

55

En otro ejemplo de la divulgación, un aparato configurado para codificar datos de vídeo comprende un codificador de vídeo configurado para decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada, almacenar los componentes de capa decodificada en una o más sub-unidades de una DPB, y realizar un proceso de gestión de la DPB en las una o más sub-unidades, en el que el proceso de gestión de la DPB se gestiona por separado para cada una de las una o más sub-unidades.

60

En otro ejemplo de la divulgación, un aparato configurado para codificar datos de vídeo comprende medios para decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada, medios para almacenar los componentes de capa decodificada en una o más sub-unidades de una DPB, y medios para realizar un proceso de gestión de la DPB en las una o más sub-unidades, en el que el proceso de gestión de la DPB se

65

gestiona por separado para cada una de las una o más sub-unidades.

5 En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, al ejecutarse, hacen que uno o más procesadores de un dispositivo configurado para codificar datos de vídeo decodifiquen datos de vídeo para producir una pluralidad de componentes de capa decodificada, almacenar los componentes de capa decodificada en una o más sub-unidades de una DPB, y realizar un proceso de gestión de la DPB en las una o más sub-unidades, en el que el proceso de gestión de la DPB se gestiona por separado para cada una de las una o más sub-unidades.

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

BREVE DESCRIPCIÓN DE LOS DIBUJOS

15 La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo que puede utilizar las técnicas descritas en esta divulgación.

La figura 2 es un diagrama conceptual que ilustra un orden ejemplar de decodificación multi-vista.

20 La figura 3 es un diagrama conceptual que ilustra una estructura de predicción ejemplar para codificación multi-vista.

La figura 4 es un diagrama conceptual que ilustra capas ejemplares de codificación de vídeo ajustable a escala.

25 La figura 5 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar que puede implementar las técnicas descritas en esta divulgación.

La figura 6 es un diagrama de bloques que ilustra un decodificador de vídeo ejemplar que puede implementar las técnicas descritas en esta divulgación.

30 La figura 7 es un diagrama conceptual que muestra una DPB ejemplar de acuerdo a un primer ejemplo de la divulgación.

35 La figura 8 es un diagrama conceptual que muestra una DPB ejemplar de acuerdo a un segundo ejemplo de la divulgación.

La figura 9 es un diagrama conceptual que muestra una DPB ejemplar de acuerdo a un tercer ejemplo de la divulgación.

40 La figura 10 es un diagrama conceptual que muestra una DPB ejemplar de acuerdo a un cuarto ejemplo de la divulgación.

La figura 11 es un diagrama de flujo que muestra un procedimiento ejemplar de acuerdo a las técnicas de esta divulgación.

45

DESCRIPCIÓN DETALLADA

50 En general, esta divulgación describe técnicas para la gestión de imágenes decodificadas en la codificación de vídeo multicapa, donde diferentes capas pueden tener diferentes resoluciones espaciales. En el contexto de esta divulgación, una capa puede ser una capa en un proceso de codificación de vídeo ajustable a escala (por ejemplo, el estándar H.264/SVC o extensiones ajustables a escala de la norma incipiente de codificación de vídeo de alta eficacia (HEVC)), una vista de texturas en codificación de vídeo multi-vista o tridimensional, o una vista en profundidad en la codificación de vídeo en 3 dimensiones. Como otro ejemplo, una capa puede corresponder a una única vista que incluye tanto componentes de vista de textura como componentes de vista en profundidad. Por lo tanto, el término "capa" puede usarse generalmente en esta divulgación para referirse a una capa en el sentido de la SVC o una vista en el sentido de la MVC. Las técnicas de esta divulgación pueden aplicarse a cualquier escenario de codificación de vídeo de ese tipo, incluyendo extensiones de multi-vista, extensiones de vídeo tridimensional y extensiones ajustables a escala de la HEVC y la H.264/AVC.

60 Las técnicas que se describen a continuación pueden aplicarse a la codificación de vídeo ajustable a escala, la de multi-vista y la tridimensional basada en códecs avanzados, incluyendo la codificación de dos o más vistas de una imagen con mapas de profundidad. Las normas de codificación de vídeo incluyen las ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo ajustable a escala (SVC) y de codificación de vídeo de multi-vista (MVC). Además, actualmente, un nuevo estándar de codificación de vídeo, concretamente, la codificación de vídeo de alta eficacia (HEVC), está siendo desarrollado por el Equipo de

65

Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC. Se describe un borrador de trabajo reciente de la HEVC en el documento JCTVC-K1003, "High Efficiency Video Coding (HEVC) text specification draft 9" ["Especificación textual de la Codificación de Vídeo de Alta Eficacia, borrador 9"], 11ª reunión: Shanghai, China, 10 al 19 de octubre de 2012, y a 17 de diciembre de 2012, está disponible para su descarga en http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v12.zip, cuyo contenido en su totalidad se incorpora en el presente documento por referencia.

Se describe un borrador más reciente de la HEVC en el documento ITU-T H.265, SERIE H: SISTEMAS AUDIOVISUALES Y DE MULTIMEDIOS, Infraestructura de Servicios Audiovisuales – Codificación de Vídeo en Movimiento, "Codificación de Vídeo de Alta Eficacia", abril de 2013 (en lo sucesivo en el presente documento, la "HEVC"). La HEVC se incorpora por referencia en el presente documento en su totalidad. Se han propuesto diversas extensiones para la HEVC. Una de tales extensiones es la extensión de rango de la HEVC, que se describe en el documento "Especificación textual de Extensiones de Gama de la Codificación de Vídeo de Alta Eficacia (HEVC): Borrador 4", JCTVC-N1005_v1, abril de 2013 (en lo sucesivo en el presente documento, "JCTVC-N1005"). Un borrador de trabajo (WD) reciente de la HEVC ajustable a escala (SHEVC), titulado "Extensión ajustable a escala de la Codificación de Vídeo de Alta Eficacia (HEVC), borrador 3", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 y ISO/IEC JTC 1/SC 29/WG 11, 14ª reunión: Viena, Austria, 25 de julio al 2 de agosto de 2013, y mencionado como SHEVC WD3 en lo sucesivo en el presente documento, está disponible en http://phenix.it-sudparis.eu/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1008-v3.zip, que se incorpora por referencia en el presente documento en su totalidad.

Las soluciones actuales para la gestión de memorias intermedias de imágenes decodificadas (DPB) se orientan únicamente a situaciones en las que se almacenan diferentes capas con la misma resolución. Es decir, las técnicas actuales para la gestión de las DPB suponen que cada capa incluye el mismo número de píxeles (es decir, resolución), dando como resultado ineficacias cuando las capas tienen un número diferentes de píxeles. En vista de este inconveniente, esta divulgación describe diversos procedimientos y técnicas para la gestión de las DPB cuando han de almacenarse múltiples componentes de capa decodificada con diferentes resoluciones.

La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo que puede utilizar las técnicas para la gestión de memorias intermedias de imágenes decodificadas en una codificación de vídeo multicapa descrita en esta divulgación. Como se muestra en la figura 1, el sistema incluye un dispositivo de origen 12 que genera datos de vídeo codificado a decodificar en un momento posterior por el dispositivo de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos de mano, tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de flujos de vídeo, o similares. En algunos casos, el dispositivo de origen 12 o el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

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

Como alternativa, los datos codificados pueden emitirse desde la interfaz de salida 22 a un dispositivo de almacenamiento 34. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 34 mediante una interfaz de entrada. El dispositivo de almacenamiento 34 puede incluir cualquiera entre una diversidad de medios de almacenamiento de datos de acceso distribuido o local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, una memoria flash, una memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento 34 puede corresponder a un servidor de ficheros u otro dispositivo de almacenamiento intermedio que pueda contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento 34 mediante flujo de transmisión o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y de transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de ficheros ejemplares incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP,

dispositivos de almacenamiento adosados a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, de DSL, de módem por cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 34 puede ser una transmisión por flujo, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación para la gestión de memorias intermedias de imágenes decodificadas en la decodificación de vídeo de multicapa no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo en apoyo de cualquiera entre una diversidad de aplicaciones de multimedios, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en flujo, por ejemplo, a través de Internet, la codificación de vídeo digital para el almacenamiento en un medio de almacenamiento de datos, la decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para prestar soporte a una transmisión de vídeo unidireccional o bidireccional para prestar soporte a aplicaciones tales como los flujos de transmisión de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la telefonía de vídeo.

En el ejemplo de la figura 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo de origen 12, el origen de vídeo 18 puede incluir un origen tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo capturado previamente, una interfaz de suministro de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo y/o un sistema de gráficos de ordenador para generar datos gráficos de ordenador como el vídeo de origen, o una combinación de dichos orígenes. Como un ejemplo, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden aplicarse a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas.

El vídeo grabado, pregrabado o generado por ordenador puede ser codificado por el codificador de vídeo 20. Los datos de vídeo codificados pueden transmitirse directamente al dispositivo de destino 14 a través de una interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados también (o como alternativa) pueden almacenarse en el dispositivo de almacenamiento 34 para un acceso posterior por el dispositivo de destino 14 u otros dispositivos, para decodificación y/o reproducción.

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

El dispositivo de pantalla 32 puede integrarse con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para mantener interfaces con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario y puede comprender cualquiera entre una amplia variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a un estándar de compresión de vídeo, tal como el estándar de codificación de vídeo de alta eficacia (HEVC), actualmente en desarrollo, y puede ser conforme al modelo de prueba de la HEVC (HM). En particular, en algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo pueden funcionar de acuerdo a extensiones de la HEVC que presten soporte a la codificación de vídeo de multi-vista, o de multi-vista más profundidad. En otros ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a otros estándares patentados o industriales, tales como el estándar ITU-T H.264, alternativamente denominado MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), o extensiones de dichos estándares, incluyendo el H.264/SVC. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de estándares de compresión de vídeo incluyen el MPEG-2 y el ITU-T H.263. En particular, de acuerdo a las técnicas de esta divulgación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a un estándar de codificación de vídeo con capacidad para codificación 3DV y/o multi-vista (por ejemplo, el 3D-HEVC, el H.264/MVC, etc.).

Aunque no se muestra en la figura 1, en algunos aspectos el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar ambos integrados en un codificador y un descodificador de audio, y pueden incluir unidades adecuadas de multiplexación y desmultiplexación, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos diferentes. Si procede, en algunos ejemplos, las unidades de multiplexación y desmultiplexación pueden ser conformes al protocolo de multiplexación ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una amplia variedad de sistemas de circuitos de codificación adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio no transitorio adecuado, legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden incluirse en uno o más codificadores o descodificadores, donde cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador combinado (códec) en un dispositivo respectivo.

La siguiente sección de la divulgación proporcionará un antecedente del estándar de la HEVC. Los esfuerzos de la normalización de la HEVC se basaron en un modelo de evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de la HEVC (HM). El HM presupone varias capacidades adicionales de los dispositivos de codificación de vídeo con respecto a los dispositivos existentes de acuerdo, por ejemplo, a la ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modalidades de codificación de intra-predicción, el HM puede proporcionar hasta treinta y tres modalidades de codificación de intra-predicción.

En general, el modelo de trabajo del HM describe que una trama o imagen de vídeo puede dividirse en una secuencia de bloques arbolados o unidades máximas de codificación (LCU) que incluyen tanto muestras de luma como de croma. Un bloque arbolado tiene un fin similar al de un macrobloque del estándar H.264. Una tajada incluye un cierto número de bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en una o más tajadas. Cada bloque arbolado puede dividirse en unidades de codificación (CU) de acuerdo a un árbol cuádruple. Por ejemplo, un bloque arbolado, como un nodo raíz del árbol cuádruple, puede dividirse en cuatro nodos hijos, y cada nodo hijo puede ser, a su vez, un nodo padre y dividirse en otros cuatro nodos hijos. Un nodo hijo final no dividido, como un nodo de hoja del árbol cuádruple, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces en que un bloque arbolado puede dividirse, y también pueden definir un tamaño mínimo de los nodos de codificación.

Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde generalmente a un tamaño del nodo de codificación y debe tener habitualmente forma cuadrada. El tamaño de la CU puede variar desde 8 x 8 píxeles hasta el tamaño del bloque arbolado con un máximo de 64 x 64 píxeles, o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, el reparto de la CU en una o más PU. Las modalidades de reparto pueden diferir entre si la CU se codifica en modalidad de omisión o modalidad directa, si se codifica en modalidad de intra-predicción o si se codifica en modalidad de inter-predicción. Las PU pueden repartirse para que no tengan forma cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, el reparto de la CU en una o más TU de acuerdo a un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada.

El estándar HEVC permite transformaciones de acuerdo a las TU, que pueden ser diferentes para diferentes CU. Las TU se dimensionan habitualmente en base al tamaño de las PU dentro de una CU determinada definida para una LCU dividida, aunque esto puede no ser siempre el caso. Las TU son habitualmente del mismo tamaño, o menores, que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas usando una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos de hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencia de píxeles asociados a las TU pueden transformarse para producir coeficientes de transformación, que pueden cuantizarse.

En general, una PU incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU está codificada en la modalidad de intra-predicción, la PU puede incluir datos que describen una modalidad de intra-predicción para la PU. Como otro ejemplo, cuando la PU está codificada en la modalidad de inter-predicción, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, una precisión de un cuarto de píxel o una precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento, que puede ser indicada por una dirección de predicción.

- En general, se usa una TU para los procesos de transformación y de cuantización. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Tras la predicción, el codificador de vídeo 20 puede calcular los valores residuales del bloque de vídeo identificado por el nodo de codificación de acuerdo a la PU. Después, el nodo de codificación se actualiza para hacer referencia a los valores residuales, en lugar del bloque de vídeo original. Los valores residuales comprenden valores de diferencia de píxeles que pueden transformarse en coeficientes de transformación, cuantificarse y explorarse usando las transformaciones y otra información de transformación especificada en las TU para producir coeficientes de transformación en serie para una codificación por entropía. El nodo de codificación puede actualizarse de nuevo una vez más para hacer referencia a estos coeficientes de transformación en serie. Esta divulgación usa habitualmente la expresión "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar la expresión "bloque de vídeo" para referirse a un bloque arbolado, es decir, una LCU o una CU, que incluye un nodo de codificación y las PU y TU.
- Una secuencia de vídeo incluye habitualmente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende generalmente una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en la cabecera del GOP, una cabecera de una o más de las imágenes o en otras ubicaciones, que indican un número de imágenes incluidas en el GOP. Cada tajada de imagen puede incluir datos sintácticos de la tajada que describen una modalidad de codificación para la respectiva tajada. Un codificador de vídeo 20 actúa habitualmente sobre bloques de vídeo dentro de tajadas de vídeo individuales, con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden tener un tamaño diferente según una norma de codificación específica.
- Como ejemplo, el HM da soporte a la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el HM da soporte a la intra-predicción en tamaños de PU de $2N \times 2N$ o $N \times N$, y a la inter-predicción en tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también da soporte al reparto asimétrico para la inter-predicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En el reparto asimétrico, una dirección de una CU no se reparte, mientras que la otra dirección se reparte entre el 25 % y el 75 %. La porción de la CU correspondiente al reparto del 25 % se indica por una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Por lo tanto, por ejemplo, " $2N \times nU$ " se refiere a una CU de tamaño $2N \times 2N$ que se divide horizontalmente con una PU de tamaño $2N \times 0,5N$ en la parte superior y una PU de tamaño $2N \times 1,5N$ en la parte inferior.
- En esta divulgación, " $N \times N$ " y " N por N " pueden usarse de manera intercambiable para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en lo que respecta a la dimensión vertical y la dimensión horizontal, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de tamaño 16×16 tendrá 16 píxeles en la dirección vertical ($y = 16$) y 16 píxeles en la dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ tiene generalmente N píxeles en la dirección vertical y N píxeles en la dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no necesitan tener necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .
- Tras una codificación intra-predictiva o inter-predictiva, usando las PU de una CU, el codificador de vídeo 20 puede calcular los datos residuales a los que se aplican las transformaciones especificadas por las TU de la CU. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las CU. El codificador de vídeo 20 puede formar los datos residuales para la CU, y después transformar los datos residuales para producir coeficientes de transformación.
- Tras cualquier transformación para producir coeficientes de transformación, el codificador de vídeo 20 puede realizar una cuantización de los coeficientes de transformación. La cuantización se refiere generalmente a un proceso en el que los coeficientes de transformación se cuantizan para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos de, o a todos, los coeficientes. Por ejemplo, un valor de n bits puede redondearse por lo bajo a un valor de m bits durante la cuantización, donde n es mayor que m .
- En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantizados para producir un vector en serie que pueda codificarse por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de la exploración de los coeficientes de transformación cuantizados, para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo a una codificación de longitud variable adaptativa según el contexto (CAVLC), una codificación aritmética binaria adaptativa según el contexto (CABAC), una codificación aritmética binaria adaptativa según el contexto basada en la sintaxis (SBAC), una codificación por entropía de reparto del intervalo de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede codificar por entropía elementos sintácticos asociados a los datos de vídeo codificados para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

Para realizar una CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo a transmitir. El contexto puede referirse, por ejemplo, a si los valores adyacentes del símbolo son distintos de cero o no. Para realizar una CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras clave en la VLC pueden construirse de tal forma que códigos relativamente más cortos correspondan a símbolos más probables, mientras que códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede conseguir un ahorro de bits, por ejemplo, sobre el uso de palabras de código de igual longitud para cada símbolo a transmitir. La determinación de la probabilidad puede basarse en un contexto asignado al símbolo.

Las extensiones de la HEVC también están actualmente en desarrollo en el JCT-3V y el JCT-VC. En el JCT-3V, se están desarrollando dos extensiones de la HEVC, la extensión multi-vista (MV-HEVC) y la extensión de vídeo tridimensional (3D-HEVC). Además, se están desarrollando dos extensiones de la AVC, la MVC+D y la 3D-AVC.

Las últimas versiones de los estándares en curso se enumeran como se indica a continuación:

- T. Suzuki, M. M. Hannuksela, Y. Chen, S. Hattori, G. Sullivan, "Extensión de la MVC para la inclusión de Mapas de Profundidades, Texto Borrador 6", JCT3V-C1001, Equipo de Colaboración Conjunta sobre Desarrollo de Extensiones de Codificación de Vídeo Tridimensional de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 4ª reunión, disponible en http://phenix.int-evry.fr/jct2/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1001-v3.zip.
- M. M. Hannuksela, Y. Chen, T. Suzuki, J.-R. Ohm, G. Sullivan, "AVC Tridimensional, Texto Borrador 8", JCT3V-F1002, Equipo de Colaboración Conjunta sobre Desarrollo de Extensiones de Codificación de Vídeo Tridimensional de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 6ª reunión, disponible en http://phenix.int-evry.fr/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1002-v3.zip.
- JCT3V-F1004, "MV-HEVC, Texto Borrador 6", G. Tech, K. Wegner, Y. Chen, M. Hannuksela, Equipo de Colaboración Conjunta sobre Desarrollo de Extensiones de Codificación de Vídeo Tridimensional de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 6ª reunión, disponible en http://phenix.int-evry.fr/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1004-v6.zip.
- Gerhard Tech, Krzysztof Wegner, Ying Chen, Sehoon Yea, "HEVC Tridimensional, Texto Borrador 2", JCT3V-F1001, Equipo de Colaboración Conjunta sobre Desarrollo de Extensiones de Codificación de Vídeo Tridimensional de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 6ª reunión, disponible en http://phenix.int-evry.fr/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1001-v2.

Ahora se analizarán técnicas de codificación de vídeo multi-vista de extensiones del estándar H.264/codificación de vídeo avanzada (AVC). Sin embargo, las técnicas de esta divulgación pueden ser aplicables con cualquier estándar de codificación de vídeo que dé soporte a la codificación multi-vista y/o propuestas multi-vista de codificación tridimensional para el estándar de HEVC incipiente (por ejemplo, HEVC multi-vista y HEVC tridimensional).

La codificación de vídeo multi-vista (MVC) es una extensión de la H.264/AVC. Se muestra un orden típico de decodificación de MVC (es decir, orden de flujo de bits) en la figura 2. La disposición del orden de decodificación se denomina codificación con el tiempo primero. Obsérvese que el orden de decodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización. En la figura 2, cada uno de S0 a S7 se refiere a diferentes vistas del vídeo multi-vista. Cada uno de T0 a T8 representa una instancia de tiempo de salida. Una unidad de acceso puede incluir las imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Por ejemplo, una primera unidad de acceso puede incluir todas las vistas S0 a S7 para una instancia de tiempo T0, una segunda unidad de acceso puede incluir todas las vistas S0 a S7 para la instancia de tiempo T1, etc.

Con fines de brevedad, la divulgación puede usar las siguientes definiciones:

componente de vista: Una *representación codificada* de una *vista* en una *única unidad de acceso*. Cuando una vista incluye tanto representaciones de textura como de profundidad codificadas, un componente de vista puede incluir un componente de vista de textura y un componente de vista de profundidad.

componente de vista de textura: Una *representación codificada* de la textura de una vista en una *única unidad de acceso*.

componente de vista de profundidad: Una *representación codificada* de la profundidad de una vista en una *única unidad de acceso*.

Como se ha analizado anteriormente, en el contexto de esta divulgación, el componente de vista, el componente de vista de textura y el componente de vista de profundidad pueden mencionarse generalmente como una capa. En la figura 2, cada una de las vistas incluye conjuntos de imágenes. Por ejemplo, la vista S0 incluye un conjunto de

imágenes 0, 8, 16, 24, 32, 40, 48, 56 y 64, la vista S1 incluye el conjunto de imágenes 1, 9, 17, 25, 33, 41, 49, 57 y 65, etc. Cada conjunto incluye dos imágenes: una imagen se denomina un componente de vista de textura, y la otra imagen se denomina un componente de vista de profundidad. El componente de vista de textura y el componente de vista de profundidad dentro de un conjunto de imágenes de una vista pueden considerarse como correspondientes entre sí. Por ejemplo, el componente de vista de textura dentro de un conjunto de imágenes de una vista se considera como correspondiente al componente de vista de profundidad en el conjunto de las imágenes de la vista, y viceversa (es decir, el componente de vista de profundidad corresponde a su componente de vista de textura en el conjunto, y viceversa). Como se usa en esta divulgación, un componente de vista de textura que corresponde a un componente de vista de profundidad puede considerarse como el componente de vista de textura y el componente de vista de profundidad que son parte de una misma vista de una única unidad de acceso.

El componente de vista de textura incluye el contenido de imagen real que se muestra. Por ejemplo, el componente de vista de textura puede incluir los componentes de luma (Y) y croma (Cb y Cr). El componente de vista de profundidad puede indicar profundidades relativas de los píxeles en su componente de vista de textura correspondiente. Como un ejemplo, el componente de vista de profundidad es una imagen en escala de grises que incluye únicamente valores de luma. En otras palabras, el componente de vista de profundidad puede no transportar ningún contenido de imagen, sino, en cambio, proporcionar una medida de las profundidades relativas de los píxeles en el componente de vista de textura.

Por ejemplo, un píxel puramente blanco en el componente de vista de profundidad indica que su píxel o píxeles correspondientes en el componente de vista de textura correspondiente está más cerca desde la perspectiva del espectador, y un píxel puramente negro en el componente de vista de profundidad indica que su píxel o píxeles correspondientes en el componente de vista de textura correspondiente está más alejado desde la perspectiva del espectador. Los diversos tonos de gris entre negro y blanco indican diferentes niveles de profundidad. Por ejemplo, un píxel muy gris en el componente de vista de profundidad indica que su píxel correspondiente en el componente de vista de textura está más alejado que un píxel ligeramente gris en el componente de vista de profundidad. Dado que únicamente es necesaria la escala de grises para identificar la profundidad de los píxeles, el componente de vista de profundidad no necesita incluir componentes de croma, ya que los valores de color para el componente de vista de profundidad pueden no servir para ningún fin.

El componente de vista de profundidad que usa únicamente valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona con fines de ilustración y no debería considerarse como limitante. En otros ejemplos, puede utilizarse cualquier técnica para indicar las profundidades relativas de los píxeles en el componente de vista de textura.

Una estructura de predicción de MVC típica (incluyendo tanto la predicción entre imágenes dentro de cada vista como la predicción entre vistas) para una codificación de vídeo multi-vista se muestra en la figura 3. Las direcciones de predicción se indican con flechas, usando el objeto al que apuntan el objeto desde el que se apunta como la referencia de predicción. En la MVC, la predicción entre vistas dispone de soporte por parte de la compensación de movimiento de disparidad, que usa la sintaxis de la compensación de movimiento de la H.264/AVC, pero permite que se use una imagen en una vista diferente como una imagen de referencia.

En el ejemplo de la figura 3, se ilustran seis vistas (que tienen los Identificadores de vista "S0" a "S5"), y se ilustran doce ubicaciones temporales ("T0" a "T11") para cada vista. Es decir, cada fila en la figura 3 corresponde a una vista, mientras que cada columna indica una ubicación temporal.

Aunque la MVC tiene una vista denominada base, que es decodificable por decodificadores de la H.264/AVC, y los pares de vistas estéreo también podrían disponer de soporte por parte de la MVC, la ventaja de la MVC es que podría dar soporte a un ejemplo que usa más de dos vistas como una entrada de vídeo tridimensional y decodifica este vídeo tridimensional representado por las múltiples vistas. Un representador de un cliente que tiene un decodificador de MVC puede esperar contenido de vídeo tridimensional con múltiples vistas.

Las imágenes en la figura 3 se indican en la intersección de cada fila y cada columna. El estándar H.264/AVC puede usar el término 'trama' para representar una parte del vídeo. Esta divulgación puede usar el término imagen y trama de forma intercambiable.

Las imágenes en la figura 3 se ilustran usando un bloque que incluye una letra, designando la letra si la imagen correspondiente está intra-codificada (es decir, es una imagen I), o inter-codificada en una dirección (es decir, como una imagen P) o en múltiples direcciones (es decir, como una imagen B). En general, las predicciones se indican con flechas, donde las imágenes a las que apuntan usan la imagen desde la que apuntan para la referencia de predicción. Por ejemplo, la imagen P de la vista S2 en la ubicación temporal T0 se predice a partir de la imagen I de la vista S0 en la ubicación temporal T0.

Tal como con la codificación de vídeo de vista individual, las imágenes de una secuencia de vídeo de codificación de vídeo multi-vista pueden codificarse de forma predictiva con respecto a imágenes en diferentes ubicaciones temporales. Por ejemplo, la imagen b de la vista S0 en la ubicación temporal T1 tiene una flecha que le apunta

desde la imagen I de la vista S0 en la ubicación temporal T0, indicando que la imagen b se predice a partir de la imagen I. Adicionalmente, sin embargo, en el contexto de una codificación de vídeo multi-vista, las imágenes pueden predecirse entre vistas. Es decir, un componente de vista puede usar los componentes de vista en otras vistas como referencia. En la MVC, por ejemplo, la predicción entre vistas se realiza como si el componente de vista en otra vista fuese una referencia de inter-predicción. Las referencias entre vistas potenciales se señalizan en la extensión de MVC del conjunto de parámetros de secuencia (SPS) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, lo que permite un ordenamiento flexible de las referencias de inter-predicción o de predicción entre vistas. La predicción entre vistas es también una característica de la extensión multi-vista propuesta de la HEVC, incluyendo la HEVC tridimensional (multi-vista más profundidad).

La figura 3 proporciona diversos ejemplos de predicción entre vistas. Las imágenes de la vista S1, en el ejemplo de la figura 3, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como predichas entre vistas a partir de imágenes de las vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la imagen b de la vista S1 en la ubicación temporal T1 se predice a partir de cada una de las imágenes B de la vista S1 en las ubicaciones temporales T0 y T2, así como las imágenes b de las vistas S0 y S2 en la ubicación temporal T1.

En algunos ejemplos, la figura 3 puede visualizarse como ilustrando los componentes de vista de textura. Por ejemplo, las imágenes I, P, B y b ilustradas en la figura 2 pueden considerarse como componentes de vista de textura para cada una de las vistas. De acuerdo a las técnicas descritas en esta divulgación, para cada uno de los componentes de vista de textura ilustrados en la figura 3, hay un componente de vista de profundidad correspondiente. En algunos ejemplos, los componentes de vista de profundidad pueden predecirse de una manera similar a la ilustrada en la figura 3 para los componentes de vista de textura correspondientes.

La codificación de dos vistas también puede disponer de soporte por parte de la MVC. Una de las ventajas de la MVC es que un codificador de MVC puede tomar más de dos vistas en forma de una entrada de vídeo tridimensional y un decodificador de la MVC puede decodificar tal representación multi-vista. Como tal, cualquier representador con un decodificador de MVC puede decodificar contenido de vídeo tridimensional con más de dos vistas.

Como se ha analizado anteriormente, en la MVC, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso (lo que significa, en algunos casos, con la misma instancia de tiempo). Al codificar una imagen en una de las vistas no de base, puede añadirse una imagen en una lista de imágenes de referencia, si está en una vista diferente pero dentro de una misma instancia de tiempo. Una imagen de referencia de predicción entre vistas puede ponerse en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de inter-predicción. Como se muestra en la figura 3, un componente de vista puede usar los componentes de vista en otras vistas como referencia. En la MVC, la predicción entre vistas se realiza como si el componente de vista en otra vista fuese una referencia de inter-predicción.

En la MVC, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso (es decir, con la misma instancia de tiempo). Al codificar una imagen en una de las vistas no de base, puede añadirse una imagen en una lista de imágenes de referencia, si está en una vista diferente pero con una misma instancia de tiempo. Una imagen de referencia de predicción entre vistas puede ponerse en cualquier posición de una lista de imágenes de referencia, tal como cualquier imagen de referencia de inter-predicción.

Como se muestra en la figura 3, un componente de vista puede usar los componentes de vista en otras vistas como referencia. Esto se denomina predicción entre vistas. En la MVC, la predicción entre vistas se realiza como si el componente de vista en otra vista fuese una referencia de inter-predicción. Sin embargo, las referencias potenciales entre vistas se señalizan en la extensión de la MVC del conjunto de parámetros de secuencia (SPS) (como se muestra en la Tabla 1) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, lo que permite un ordenamiento flexible de las referencias de inter-predicción o de predicción entre vistas.

Tabla 1 - Tabla de sintaxis de la extensión de la MVC del SPS

	Descriptor
extensión_mvc_conjunto_parámetros_sec() {	
núm_vistas_menos1	ue(v)
for(i = 0; i <= núm_vistas_menos1; i++)	
id_vista[i]	ue(v)
for(i = 1; i <= núm_vistas_menos1; i++) {	
núm_refs_ancla_l0[i]	ue(v)
for(j = 0; j < núm_refs_ancla_l0[i]; j++)	
ref_ancla_l0[i][j]	ue(v)
núm_refs_ancla_l1[i]	ue(v)
for(j = 0; j < núm_refs_ancla_l1 [i]; j++)	

ref_ancla_l1[i][j]		ue(v)
}		
for(i = 1; i <= núm_vistas_menos1; i++) {		
núm_refs_no_ancla_l0[i]		ue(v)
for(j = 0; j < núm_refs_no_ancla_l0[i]; j++)		
ref_no_ancla_l0[i][j]		ue(v)
núm_refs_no_ancla_l1[i]		ue(v)
for(j = 0; j < núm_refs_no_ancla_l1[i]; j++)		
ref_no_ancla_l1[i][j]		ue(v)
}		
núm_nivel_valores_señalizados_menos1		ue(v)
for(i = 0; i <= núm_nivel_valores_señalizados_menos1; i++) {		
nivel_idc[i]		u(8)
núm_ops_aplicables_menos1[i]		ue(v)
for(j = 0; j <= núm_ops_aplicables_menos1[i]; j++) {		
id_temporal_op_aplicable[i][j]		u(3)
op_aplicable_núm_vistas_objetivo_menos1[i][j]		ue(v)
for(k = 0; k <=		
op_aplicable_núm_vistas_objetivo_menos1[i][j]; k++)		
op_aplicable_id_vista_objetivo[i][j][k]		ue(v)
op_aplicable_núm_vistas_menos1[i][j]		ue(v)
}		
}		
}		

5 En la extensión de la MVC del SPS, para cada vista, se señalizan el número de vistas que puede usarse para formar la lista 0 de imágenes de referencia y la lista 1 de imágenes de referencia. Las relaciones de predicción para una imagen de anclaje, como se señala en la extensión de la MVC del SPS, pueden ser diferentes de la relación de predicción para una imagen no de anclaje (señalizada en la extensión de la MVC del SPS) de la misma vista.

10 La siguiente sección analizará la codificación de vídeo multi-vista y tridimensional con respecto a la HEVC. En particular, las técnicas ejemplares de esta divulgación son aplicables al codificar dos o más vistas, cada una con un componente de vista de textura y un componente de vista de profundidad. La pluralidad de imágenes de vídeo para cada vista pueden denominarse componentes de vista de textura. Cada componente de vista de textura tiene un componente de vista de profundidad correspondiente. Los componentes de vista de textura incluyen contenido de vídeo (por ejemplo, componentes de luma y croma de valores de píxel), y los componentes de vista de profundidad pueden indicar profundidades relativas de los píxeles dentro de los componentes de vista de textura.

15 Las técnicas de esta divulgación se refieren a la codificación de datos de vídeo tridimensional, codificando datos de textura y de profundidad. En general, el término "textura" se usa para describir valores de luminancia (es decir, brillo o "luma") de una imagen y valores de crominancia (es decir, color o "croma") de la imagen. En algunos ejemplos, una imagen de textura puede incluir un conjunto de datos de luminancia y dos conjuntos de datos de crominancia, para matices azules (Cb) y matices rojos (Cr). En ciertos formatos de croma, tales como 4:2:2 o 4:2:0, los datos de croma se sub-muestran con respecto a los datos de luma. Es decir, la resolución espacial de los componentes de crominancia es menor que la resolución espacial de los componentes de luminancia correspondientes, por ejemplo, un medio o un cuarto de la resolución de luminancia.

25 Los datos de profundidad generalmente describen valores de profundidad para los datos de textura correspondientes. Por ejemplo, una imagen de profundidad puede incluir un conjunto de píxeles de profundidad que describen, cada uno, la profundidad para los datos de textura correspondientes. Los datos de profundidad pueden usarse para determinar la disparidad horizontal para los datos de textura correspondientes. Por lo tanto, un dispositivo que recibe los datos de textura y profundidad puede mostrar una primera imagen de textura para una vista (por ejemplo, una vista del ojo izquierdo) y usar los datos de profundidad para modificar la primera imagen de textura, para generar una segunda imagen de textura para la otra vista (por ejemplo, una vista del ojo derecho)

30 desplazando valores de píxel de la primera imagen en los valores de disparidad horizontal determinados en base a los valores de profundidad. En general, la disparidad horizontal (o simplemente "disparidad") describe el desplazamiento espacial horizontal de un píxel en una primera vista con respecto a un píxel correspondiente en una segunda vista, donde los dos píxeles corresponden a la misma parte del mismo objeto, como se representa en las

35 dos vistas.

En otros ejemplos más, los datos de profundidad pueden definirse para los píxeles en una dimensión z perpendicular

al plano de imagen, de tal forma que una profundidad asociada a un píxel dado se defina con respecto a un plano de disparidad nula definido para la imagen. Dicha profundidad puede usarse para crear disparidad horizontal para mostrar el píxel, de tal forma que el píxel se muestre de forma diferente para el ojo izquierdo y el derecho, según el valor de profundidad de la dimensión z del píxel con respecto al plano de disparidad nula. El plano de disparidad nula puede cambiar para diferentes partes de una secuencia de vídeo, y la magnitud de la profundidad relativa al plano de disparidad nula también puede cambiar. Los píxeles situados en el plano de disparidad nula pueden definirse de forma similar para el ojo izquierdo y el derecho. Los píxeles situados delante del plano de disparidad nula pueden mostrarse en diferentes ubicaciones para el ojo izquierdo y el derecho (por ejemplo, con disparidad horizontal) para crear una percepción de que el píxel parece salir de la imagen en la dirección z perpendicular al plano de imagen. Los píxeles situados tras el plano de disparidad nula pueden mostrarse ligeramente borrosos, para presentar una ligera percepción de profundidad, o pueden mostrarse en ubicaciones diferentes para el ojo izquierdo y derecho (por ejemplo, sin disparidad horizontal que esté opuesta a la de los píxeles situados delante del plano de disparidad nula). También pueden usarse muchas otras técnicas para transportar o definir datos de profundidad para una imagen.

Para cada píxel en el componente de vista de profundidad, puede haber uno o más píxeles correspondiente en el componente de vista de textura. Por ejemplo, si las resoluciones espaciales del componente de vista de profundidad y del componente de vista de textura son iguales, cada píxel en el componente de vista de profundidad corresponde a un píxel en el componente de vista de textura. Si la resolución espacial del componente de vista de profundidad es menor que la del componente de vista de textura, entonces cada píxel en el componente de vista de profundidad corresponde a múltiples píxeles en el componente de vista de textura. El valor del píxel en el componente de vista de profundidad puede indicar la profundidad relativa de los uno o más píxeles correspondientes en la vista de textura.

En algunos ejemplos, un codificador de vídeo señala los datos de vídeo para los componentes de vista de textura y los componentes de vista de profundidad correspondientes para cada una de las vistas. Un decodificador de vídeo utiliza tanto los datos de vídeo de componentes de vista de textura como los componentes de vista de profundidad para decodificar el contenido de vídeo de las vistas para la visualización. Después, una pantalla muestra el vídeo de multi-vista para producir vídeo tridimensional.

También se están desarrollando extensiones ajustables a escala de la HEVC por parte del JCT-VC. La figura 4 es un diagrama conceptual que ilustra un ejemplo de codificación de vídeo ajustable a escala. Aunque la figura 4 se describe con respecto a la H.264/AVC y la SVC, debería entenderse que pueden codificarse capas similares usando otros esquemas de codificación de vídeo multicapa, incluyendo la extensión ajustable a escala de la HEVC. El ejemplo de la figura 4 muestra capas codificadas usando el mismo códec. En otros ejemplos, las capas pueden codificarse usando un códec de múltiples estándares. Por ejemplo, una capa base puede codificarse usando la H.264/AVC, mientras que una capa de realce puede codificarse usando una extensión ajustable a escala para la HEVC. Por lo tanto, las referencias a la SVC a continuación pueden aplicarse a la codificación de vídeo ajustable a escala en general, y no se restringen a la H.264/SVC.

En la SVC, la ajustabilidad a escala puede habilitarse en tres dimensiones incluyendo, por ejemplo, la espacial, la temporal y la de calidad (que se representa como una tasa de bits o relación entre señal y ruido (SNR)). En general, normalmente puede conseguirse una mejor representación por añadidura a una representación en cualquier dimensión. Por ejemplo, en el ejemplo de la figura 4, la capa 0 se codifica en un formato intermedio común de un cuarto (QCIF) que tiene una velocidad de tramas de 7,5 Hz y una velocidad de bits de 64 kilo-octetos por segundo (KBPS). Además, la capa 1 se codifica en QCIF con una velocidad de tramas de 15 Hz y una velocidad de bits de 64 KBPS, la capa 2 se codifica en CIF con una velocidad de tramas de 15 Hz y una velocidad de bits de 256 KBPS, la capa 3 se codifica en QCIF con una velocidad de tramas de 7,5 Hz y una velocidad de bits de 512 KBPS, y la capa 4 se codifica en 4CIF que tiene una velocidad de tramas de 30 Hz y una velocidad de bits en Mega-octetos por segundo (MBPS). Se ha de apreciar que el número particular, el contenido y la disposición de las capas mostradas en la figura 4 se proporcionan únicamente con fines de ejemplo.

En cualquier caso, una vez que un codificador de vídeo (tal como el codificador de vídeo 20) ha codificado el contenido de tal manera ajustable a escala, un decodificador de vídeo (tal como el decodificador de vídeo 30) puede usar una herramienta extractora para adaptar el contenido entregado real de acuerdo a los requisitos de la aplicación, lo que puede depender, por ejemplo, del cliente o del canal de transmisión.

En la SVC, las imágenes que tienen la capa espacial y de calidad más baja son típicamente compatibles con la H.264/AVC. En el ejemplo de la figura 4, las imágenes con la más baja capa espacial y de calidad (imágenes en la capa 0 y la capa 1, con resolución QCIF) pueden ser compatibles con la H.264/AVC. Entre ellas, aquellas imágenes con el más bajo nivel temporal forman la capa base temporal (capa 0). Esta capa base temporal (capa 0) puede mejorarse con imágenes de niveles temporales más altos (capa 1).

Además de la capa compatible con la H.264/AVC, pueden añadirse varias capas de mejora espacial y/o de calidad para proporcionar ajustabilidades a escala, espaciales y/o de calidad. Cada capa de mejora espacial o de calidad puede ser temporalmente ajustable a escala en sí misma, con la misma estructura de ajustabilidad temporal a escala

que la capa compatible con la H.264/AVC.

Una vez que se ha decodificado cada una de las capas, por ejemplo, mediante el decodificador de vídeo 30 o el bucle de reconstrucción del codificador de vídeo 20, la capa decodificada se almacena en una DPB. Una DPB es una memoria intermedia o memoria que se usa para almacenar imágenes y, en el ejemplo de esta divulgación, las capas o imágenes decodificadas al usar técnicas de codificación de vídeo multicapa. Las capas decodificadas almacenadas en la DPB pueden usarse como referencias para la inter-predicción (incluyendo compensación de movimiento, predicción entre vistas y entre capas), para el reordenamiento de la salida y el retardo de la salida. En la HEVC, y otros estándares de codificación de vídeo, la operación de la DPB a menudo se especifica con relación a un decodificador de referencia hipotético (HRD). El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para gestionar la DPB para realizar diversas acciones, incluyendo marcar imágenes decodificadas almacenadas en la DPB como no usadas para referencia (es decir, incapacitadas para ser usadas como una imagen de referencia para un proceso de inter predicción), marcar una imagen decodificada para su salida (por ejemplo, al dispositivo de visualización 32), y marcar una imagen para su eliminación de la DPB (también denominada batacazo). Una imagen se elimina habitualmente (es decir, se despeña) de la DPB cuando ya no es necesaria como una imagen de referencia para una inter-predicción y ya no es necesaria para la salida.

Ya sea la codificación usando técnicas de codificación de vídeo ajustable a escala, multi-vista o tridimensional, las diferentes capas, vistas de textura y/o vistas de profundidad pueden tener diferentes resoluciones espaciales. Es decir, los componentes (por ejemplo, componentes de vista o componentes de capa, correspondientes a un mapa de imágenes o profundidades) de diferentes capas o vistas pueden tener diferentes resoluciones espaciales. En las técnicas de gestión de DPB existentes, las operaciones relacionadas con la salida se realizan para cada vista o capa por separado. Esto significa que, para cada vista, el marcado para la salida o eliminación de una imagen de referencia (de una capa/vista) se hace por separado, aunque pueden aplicarse restricciones o alineaciones.

Sin pérdida de generalidad, en esta divulgación, un componente de representación de capa decodificada, o de vista de textura/profundidad decodificada, de una unidad de acceso también se denomina un componente de capa decodificada de la unidad de acceso. En la codificación de vídeo multicapa que implica múltiples resoluciones espaciales en diferentes capas, en particular, cuando se aplica una decodificación multi-bucle, y en la que al menos dos capas más pueden ser completamente reconstruidas durante el proceso de decodificación, sería necesario el almacenamiento de los componentes de capa decodificada de diferentes capas en la memoria o memorias intermedias de imágenes decodificadas.

En la SVC, se aplica una decodificación de bucle único, por lo tanto, únicamente la capa más alta puede reconstruirse completamente y únicamente puede ser necesario almacenar el componente de capa decodificada de la capa más alta en cada unidad de acceso. En las técnicas existentes para la MVC, aunque se aplique una decodificación de múltiples bucles, se requieren diferentes vistas para tener la misma resolución espacial. Por lo tanto, aunque han de almacenarse múltiples componentes de capa decodificada en cada unidad de acceso, todos tienen la misma resolución. Actualmente, no hay ningún mecanismo para la gestión de la DPB cuando es necesario almacenar múltiples componentes de capa decodificada con diferentes resoluciones.

En vista de estos inconvenientes, esta divulgación describe diversos mecanismos y técnicas para la gestión de la DPB cuando han de almacenarse múltiples componentes de capa decodificada con diferentes resoluciones.

La figura 5 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar 20 que puede implementar las técnicas para la gestión de la DPB descritas en esta divulgación. El codificador de vídeo 20 puede configurarse para codificar los datos de vídeo de acuerdo a cualquier técnica de codificación de vídeo, incluyendo la HEVC y la H.264/AVC, así como extensiones ajustables a escala, de multi-vista y tridimensionales de dichos estándares. El ejemplo de la figura 5 se explicará con referencia a la HEVC. A este respecto, el bucle de codificación de vídeo mostrado en la figura 5 puede aplicarse a cada capa de un proceso de codificación de vídeo ajustable a escala (es decir, capas de base y capas de mejora), a cada vista de un proceso de codificación de vídeo multi-vista, o tanto a las vistas de textura como de profundidad de un proceso de codificación de vídeo tridimensional.

El codificador de vídeo 20 puede realizar la intra-codificación y la inter-codificación de bloques de vídeo dentro de las tajadas de vídeo. La intra-codificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en vídeo dentro de una trama o imagen de vídeo determinadas. La inter-codificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. La intra-modalidad (modalidad I) puede referirse a cualquiera de varias modalidades de compresión basadas en el espacio. Las inter-modalidades, tales como una predicción unidireccional (modalidad P) o una bi-predicción (modalidad B), pueden referirse a cualquiera de varias modalidades de compresión basadas en el tiempo. Además, el codificador de vídeo 20 puede realizar una predicción entre vistas y/o una predicción entre capas, entre diferentes vistas o capas, como se ha descrito anteriormente.

En el ejemplo de la figura 5, el codificador de vídeo 20 incluye una unidad de procesamiento de predicción 41, una DPB 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantización 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 41 incluye una unidad de

estimación de movimiento y disparidad 42, una unidad de compensación de movimiento y disparidad 44 y una unidad de procesamiento de intra-predicción 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de cuantificación inversa 58, una unidad de procesamiento de transformación inversa 60 y un sumador 62. También puede incluirse un filtro de desbloqueo (no mostrado en la figura 5) para filtrar límites de bloques para eliminar distorsiones de formación de bloques de un vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará habitualmente la salida del sumador 62. También pueden usarse filtros de bucle adicionales (en bucle o post-bucle) además del filtro de desbloqueo.

Como se muestra en la figura 5, el codificador de vídeo 20 recibe datos de vídeo y puede configurarse para repartir los datos en bloques de vídeo. Este reparto también incluye repartir en tajadas, losas u otras unidades más grandes, así como el reparto de bloques de vídeo, por ejemplo, de acuerdo a una estructura de árbol cuádruple de las LCU y CU. El codificador de vídeo 20 ilustra generalmente los componentes que codifican bloques de vídeo dentro de una tajada de vídeo que se va a codificar. El fragmento puede dividirse en múltiples bloques de vídeo (y posiblemente en conjuntos de bloques de vídeo denominados losas). La unidad de procesamiento de predicción 41 puede seleccionar una entre una pluralidad de modalidades de codificación posibles, tales como una entre una pluralidad de modalidades de intra-codificación o una entre una pluralidad de modalidades de inter-codificación o modalidades de codificación entre vistas, para el bloque de vídeo actual en base a resultados de error (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intra-codificado o inter-codificado resultante al sumador 50 para generar datos de bloques residuales, y al sumador 62 para reconstruir el bloque codificado para su uso como una imagen de referencia.

La unidad de procesamiento de intra-predicción 46 dentro de la unidad de procesamiento de predicción 41 puede realizar una codificación intra-predictiva del bloque de vídeo actual con respecto a uno o más bloques vecinos en la misma trama o tajada que el bloque actual que se va a codificar para proporcionar compresión espacial. La unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44 en la unidad de procesamiento de predicción 41 realizan la codificación inter-predictiva y/o la codificación entre vistas del bloque de vídeo actual con respecto a uno o más bloques predictivos en una o más imágenes de referencia, capas de imágenes de referencia y/o vistas de referencia para proporcionar predicción temporal y entre vistas.

La unidad de estimación de movimiento y disparidad 42 puede configurarse para determinar la modalidad de inter-predicción y/o la modalidad de predicción entre vistas para una tajada de vídeo de acuerdo a un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar tajadas de vídeo en la secuencia como las tajadas P o las tajadas B. La unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44 pueden estar estrechamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, llevada a cabo por la unidad de estimación de movimiento y disparidad 42, es el proceso de generar vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo en una trama o imagen de vídeo actual con respecto a un bloque predictivo dentro de una imagen de referencia. La estimación de la disparidad, realizada por la unidad de estimación de movimiento y disparidad 42, es el proceso de generación de vectores de disparidad, que pueden usarse para predecir un bloque actualmente codificado a partir de un bloque en una vista diferente.

Un bloque predictivo es un bloque del que se sabe que se corresponde estrechamente con la PU del bloque de vídeo que va a codificarse, en lo que respecta a la diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), una suma de diferencia de cuadrados (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la DPB 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxeles de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede llevar a cabo una búsqueda de movimiento con respecto a las posiciones de píxeles completos y las posiciones fraccionarias de píxel y proporcionar un vector de movimiento con precisión fraccionaria de píxeles.

La unidad de estimación de movimiento y disparidad 42 calcula un vector de movimiento (para la predicción compensada de movimiento) y/o un vector de disparidad (para la predicción compensada de disparidad) para una PU de un bloque de vídeo en una tajada predicha inter-codificada o de entre vistas, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse a partir de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la DPB 64. Para la predicción entre vistas, la imagen de referencia está en una vista diferente. La unidad de estimación de movimiento y disparidad 42 envía el vector de movimiento y/o el vector de disparidad calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.

La compensación de movimiento y/o la compensación de disparidad, realizadas por la unidad de compensación de movimiento y disparidad 44, pueden implicar coger o generar el bloque predictivo en base al vector de movimiento determinado por la estimación de movimiento y/o la estimación de disparidad, posiblemente, realizando interpolaciones con respecto a una precisión de sub-píxeles. Tras recibir el vector de movimiento y/o la disparidad

para la PU del bloque de vídeo actual, la unidad de compensación de movimiento y disparidad 44 puede localizar el bloque predictivo al que apunta el vector de movimiento y/o el vector de disparidad en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo a los valores de píxeles del bloque de vídeo actual que está codificándose, generando valores de diferencia de píxeles. Los valores de diferencia de píxeles forman datos residuales para el bloque, y pueden incluir tanto componentes de diferencia de luma como de croma. El sumador 50 representa el componente o los componentes que llevan a cabo esta operación de sustracción. La unidad de compensación de movimiento y disparidad 44 también puede generar elementos sintácticos asociados a los bloques de vídeo y a la tajada de vídeo, para su uso por parte del descodificador de vídeo 30 a la hora de descodificar los bloques de vídeo de la tajada de vídeo.

La unidad de procesamiento de intra-predicción 46 puede intra-predecir un bloque actual, como alternativa a la inter-predicción llevada a cabo por la unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44, como se ha descrito anteriormente. En particular, la unidad de procesamiento de intra-predicción 46 puede determinar una modalidad de intra-predicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intra-predicción 46 puede codificar un bloque actual usando varias modalidades de intra-predicción, por ejemplo, durante diferentes pasadas de codificación, y la unidad de procesamiento de intra-predicción 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar una modalidad de intra-predicción adecuada a usar, entre las modalidades probadas. Por ejemplo, la unidad de procesamiento de intra-predicción 46 puede calcular valores de velocidad y distorsión usando un análisis de velocidad y distorsión para las diversas modalidades de intra-predicción probadas, y seleccionar la modalidad de intra-predicción que tenga las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad y distorsión determina generalmente una magnitud de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para generar el bloque codificado, así como una tasa de bits (es decir, el número de bits) usada para generar el bloque codificado. La unidad de procesamiento de intra-predicción 46 puede calcular razones a partir de las distorsiones y velocidades de los diversos bloques codificados para determinar qué modalidad de intra-predicción presenta el mejor valor de velocidad y distorsión para el bloque.

En cualquier caso, tras seleccionar una modalidad de intra-predicción para un bloque, la unidad de procesamiento de intra-predicción 46 puede proporcionar información que indica la modalidad de intra-predicción seleccionada para el bloque a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica a la modalidad de intra-predicción seleccionada de acuerdo a las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir datos de configuración en el flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modalidades de intra-predicción y una pluralidad de tablas de índices de modalidades de intra-predicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para varios bloques, e indicaciones de la modalidad de intra-predicción más probable, una tabla de índices de modalidades de intra-predicción y una tabla modificada de índices de modalidades de intra-predicción, a usar en cada uno de los contextos.

Después de que la unidad de procesamiento de predicción 41 genere el bloque predictivo para el bloque de vídeo actual mediante la inter-predicción o la intra-predicción, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo del bloque de vídeo actual. Los datos de vídeo residuales en el bloque residual pueden incluirse en una o más TU y se aplicarse a la unidad de procesamiento de transformación 52. La unidad de procesamiento de transformación 52 transforma los datos de vídeo residuales en coeficientes de transformación residuales usando una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar. La unidad de procesamiento de transformación 52 puede convertir los datos de vídeo residuales de un dominio de píxeles en un dominio de transformación, tal como un dominio de frecuencia.

La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantización 54. La unidad de procesamiento de cuantización 54 cuantiza los coeficientes de transformación para reducir adicionalmente la velocidad de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos de, o todos, los coeficientes. El grado de cuantización puede modificarse ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de cuantización 54 puede luego llevar a cabo una exploración de la matriz, incluyendo los coeficientes de transformación cuantizados. Como alternativa, la unidad de codificación por entropía 56 puede llevar a cabo la exploración.

Tras la cuantización, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede llevar a cabo una codificación de longitud variable adaptable al contexto (CAVLC), una codificación aritmética binaria adaptable al contexto (CABAC), una codificación aritmética binaria adaptable al contexto y basada en la sintaxis (SBAC), una codificación por entropía de la división en intervalos de probabilidad (PIPE) u otra metodología o técnica de codificación por entropía. Tras la codificación por entropía realizada por la unidad de codificación por entropía 56, el flujo de bits codificado puede transmitirse al descodificador de vídeo 30 o archivarse para su posterior transmisión o recuperación por el descodificador de vídeo 30. La unidad de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos sintácticos para la tajada de vídeo actual que se codifica.

La unidad de cuantización inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican una cuantización inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, para su uso posterior como un bloque de referencia de una imagen de referencia. Esta parte del codificador de vídeo 20 a veces se denomina el bucle de reconstrucción, y decodifica eficazmente un bloque de vídeo codificado para su uso como una imagen de referencia en la inter-predicción. Las imágenes reconstruidas se almacenan en la DPB 64.

La unidad de compensación de movimiento y disparidad 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento y disparidad 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores fraccionarios de píxeles para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, generado por la unidad de compensación de movimiento 44, para generar un bloque de referencia para su almacenamiento en la DPB 64. El bloque de referencia puede ser usado por la unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44 como un bloque de referencia para inter-predecir un bloque en una trama o imagen de vídeo posterior. Como se analizará en más detalle a continuación, el codificador de vídeo 20 puede configurarse para realizar las técnicas de gestión de DPB de esta divulgación que admiten el almacenamiento y la gestión de múltiples capas de datos de vídeo en la DPB a diferentes resoluciones.

La figura 6 es un diagrama de bloques que ilustra un decodificador de vídeo ejemplar 30 que puede implementar las técnicas de gestión de DPB descritas en esta divulgación. En el ejemplo de la figura 6, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 80, una unidad de procesamiento de predicción 81, una unidad de cuantización inversa 86, una unidad de transformación inversa 88, un sumador 90 y una DPB 92. La unidad de procesamiento de predicción 81 incluye una unidad de compensación de movimiento y disparidad 82 y una unidad de procesamiento de intra-predicción 84. En algunos ejemplos, el decodificador de vídeo 30 puede llevar a cabo una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la figura 5.

Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de una tajada de vídeo codificado y elementos sintácticos asociados del codificador de vídeo 20. La unidad de decodificación por entropía 80 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantizados, vectores de movimiento, vectores de disparidad y otros elementos sintácticos. La unidad de decodificación por entropía 80 envía los vectores de movimiento, los vectores de disparidad y otros elementos sintácticos a la unidad de procesamiento de predicción 81. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de tajada de vídeo y/o el nivel de bloque de vídeo.

Cuando el fragmento de vídeo se codifica como una tajada intra-codificado (I), la unidad de procesamiento de intra-predicción 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de vídeo de la tajada de vídeo actual en base a una modalidad de intra-predicción señalizada y a datos de bloques decodificados previamente de la trama o imagen actual. Cuando la trama de vídeo se codifica como una tajada inter-codificada (es decir, B o P) o una tajada predicha entre vistas, la unidad de compensación de movimiento y disparidad 82 de la unidad de procesamiento de predicción 81 produce bloques predictivos para un bloque de vídeo de la tajada de vídeo actual, en base a los vectores de movimiento, los vectores de disparidad y otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 80. Los bloques predictivos pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de tramas de referencia, Lista 0 y Lista 1, usando técnicas de construcción por omisión en base a las imágenes de referencia almacenadas en la DPB 92.

La unidad de compensación de movimiento y disparidad 82 determina la información de predicción para un bloque de vídeo de la tajada de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento y disparidad 82 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, de intra-predicción o de inter-predicción) usada para codificar los bloques de vídeo de la tajada de vídeo, un tipo de tajada de inter-predicción o predicción entre vistas (por ejemplo, tajada B o tajada P), la información de construcción para una o más de las listas de imágenes de referencia para la tajada, los vectores de movimiento y/o los vectores de disparidad para cada bloque de vídeo inter-codificado de la tajada, un estado de inter-predicción para cada bloque de vídeo inter-codificado de la tajada y otra información para decodificar los bloques de vídeo en la tajada de vídeo actual.

La unidad de compensación de movimiento y disparidad 82 también puede realizar interpolación en base a filtros de interpolación. La unidad de compensación de movimiento 82 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para fracciones de píxeles 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 vídeo 20 para los elementos sintácticos recibidos y usar los filtros de interpolación para producir bloques predictivos.

5 La unidad de cuantización inversa 86 cuantiza de manera inversa, es decir, des-cuantiza, los coeficientes de transformación cuantizados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización calculado por el codificador de vídeo 20 para cada bloque de vídeo en la tajada de vídeo para determinar un grado de cuantización y, asimismo, un grado de cuantización inversa que debería aplicarse. La unidad de procesamiento de transformación inversa 88 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación entera inversa o un
10 proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación con el fin de generar bloques residuales en el dominio de píxeles.

15 Después de que la unidad de compensación de movimiento y disparidad 82 genere el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y/o los vectores de disparidad y otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformación inversa 88 con los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados a fin de eliminar las distorsiones por formación de bloques.
20 También pueden usarse otros filtros de bucle (en el bucle de codificación o bien después del bucle de codificación) para allanar las transiciones de píxeles, o mejorar de otro modo la calidad del vídeo.

25 Los bloques de vídeo decodificados en una trama o imagen dadas se almacenan entonces en la DPB 92, que almacena las imágenes de referencia usadas para una compensación de movimiento posterior. Como se describirá en más detalle a continuación, el decodificador de vídeo 30 puede configurarse para realizar las técnicas de esta divulgación para la gestión de las DPB al almacenar múltiples capas de datos de vídeo en la DPB a diferentes resoluciones. La DPB 92 almacena también vídeo decodificado para su presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1.

30 Un codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30) puede configurarse para realizar cualquier combinación de las siguientes técnicas para la gestión de las DPB. En general, cada una de las siguientes técnicas muestra las siguientes características. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para decodificar (por ejemplo, en el bucle de reconstrucción para el codificador de vídeo 20) datos de vídeo para producir una pluralidad de componentes de capa decodificada. El codificador de vídeo
35 20 y/o el decodificador de vídeo 30 pueden configurarse para almacenar los componentes de capa decodificada en una o más sub-unidades de una DPB. En este contexto, una "unidad" de la DPB es un área accesible por separado de la DPB, que contiene datos de vídeo reconstruidos o decodificados que tienen alguna característica en común. Además, una sub-unidad de una DPB puede considerarse un área accesible por separado de la DPB que se gestiona y se trata ella misma como una DPB individual.

40 El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse adicionalmente para realizar un proceso de gestión de las DPB en las una o más sub-unidades, en donde el proceso de gestión de las DPB se gestiona por separado para cada una de las una o más sub-unidades. Los procesos de gestión de las DPB pueden incluir uno o más entre eliminar un componente de capa decodificada de la sub-unidad, o marcar un componente de
45 capa decodificada en la sub-unidad como no usado para referencia. De esta manera, en algunos ejemplos, las capas decodificadas y/o reconstruidas que tienen diferentes características (por ejemplo, resolución espacial, tipo de capa) pueden gestionarse por separado.

50 En ejemplos adicionales de la divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para realizar el proceso de gestión de las DPB en las una o más sub-unidades, realizando el proceso de gestión de las DPB en un primer componente de capa decodificada en una unidad de acceso, y realizando el mismo proceso de gestión de las DPB en otros componentes de capa decodificada en la misma unidad de acceso.

55 En un ejemplo de la divulgación, un codificador de vídeo puede configurarse para decodificar o reconstruir datos de vídeo para producir una pluralidad de componentes de capa decodificada, y almacenar los componentes de capa decodificada en una única sub-unidad de una DPB, de tal forma que la única sub-unidad de la DPB contenga una unión de componentes de capa decodificada para una capa completamente reconstruida dentro de una unidad de acceso. Es decir, en este ejemplo, la DPB consiste solamente en una única sub-unidad.

60 En este ejemplo de la divulgación, la única sub-unidad de la DPB almacena la unión de componentes de capa decodificada para una capa completamente reconstruida dentro de una unidad de acceso. En algunos ejemplos, la unión de componentes de capa decodificada para una capa completamente reconstruida puede ser la unión de tanto los componentes de vista de textura como los componentes de vista de profundidad, múltiples componentes de vista de textura, o una capa de base y una o más capas de mejora. Una unidad de acceso es un conjunto de unidades de
65 capa de abstracción de red (NAL) que son consecutivos en el orden de decodificación y contienen una imagen codificada. El tamaño de las sub-unidades individuales de la DPB puede determinarse por la suma de varias

muestras de todos los componentes en todos los componentes de capa decodificada. Es decir, el tamaño de la sub-unidad individual de la DPB puede determinarse en base a las resoluciones de todos los componentes de capa decodificada almacenados en la sub-unidad. De esta manera, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para dimensionar de forma flexible la DPB de acuerdo a la resolución de las capas que se codifican realmente.

La figura 7 es un diagrama conceptual que muestra una DPB 700 configurada para almacenar la unión de componentes de capa decodificada de todas las capas completamente reconstruidas en una unidad de acceso. Cada una de las sub-unidades 710A a 710D representa una ubicación de almacenamiento para la unión de los componentes de capa decodificada. El codificador de vídeo puede configurarse adicionalmente para eliminar un componente de capa decodificada de la DPB 700. En este ejemplo de la divulgación, la eliminación de un componente particular de capa decodificada de la DPB (por ejemplo, la eliminación de un componente de vista de textura) también elimina toda una unidad de acceso decodificado relacionada con el componente de capa decodificada de la DPB (por ejemplo, también elimina otros componentes de vista de textura y/o componentes de vista de profundidad). El codificador de vídeo puede configurarse adicionalmente para marcar un componente de capa decodificada en la DPB como no usado para referencia. Asimismo, en este ejemplo, el marcado de un componente particular de capa decodificada (por ejemplo, un componente de vista de textura) como no usado para referencia también marca toda una unidad de acceso decodificada, relacionada con el componente de capa decodificada (por ejemplo, marcando también otros componentes de vista de textura y/o componentes de vista de profundidad) como no usado para referencia.

En otro ejemplo de la divulgación, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30) puede configurarse para decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada. En algunos casos, los componentes de capa decodificada tienen al menos dos resoluciones espaciales diferentes. El codificador de vídeo puede configurarse adicionalmente para almacenar los componentes de capa decodificada en una entre una pluralidad de sub-unidades de una memoria intermedia de imágenes decodificadas (DPB), según la resolución espacial de los componentes de capa decodificada. Por ejemplo, cada una entre la pluralidad de sub-unidades se asocia a una resolución espacial diferente. De esta manera, un codificador de vídeo puede configurarse para almacenar los componentes de capa decodificada (por ejemplo, un componente de vista de textura, un componente de vista de profundidad, una capa de base, una capa de mejora, etc.) en ciertas sub-unidades en base a su resolución. De esta manera, cada resolución puede manipularse por separado.

La figura 8 es un diagrama conceptual que muestra una DPB 800 configurada para almacenar los componentes de capa decodificada a diferentes resoluciones en diferentes sub-unidades. Cada una de las sub-unidades 810A a 810D representa ubicaciones de almacenamiento para componentes de capa decodificada a diferentes resoluciones. Por ejemplo, cada una de las sub-unidades 810A a 810D está diseñada para el almacenamiento de una resolución diferente 1 a 4. Cada una de las sub-unidades 810A a 810D contiene unidades separadas que están configuradas para almacenar un componente de capa decodificada asociado a una imagen. Como una ilustración ejemplar, un componente de vista de textura de una imagen puede almacenarse en una sub-unidad correspondiente a la resolución del componente de vista de textura (por ejemplo, la sub-unidad 810A), mientras que un componente de vista de profundidad de la misma imagen puede almacenarse en diferentes sub-unidades correspondientes a la resolución (habitualmente inferior) del componente de vista de profundidad (por ejemplo, la sub-unidad 810B).

El codificador de vídeo puede configurarse adicionalmente para eliminar los componentes de capa decodificada de la pluralidad de sub-unidades 810A a 810D. En este ejemplo de la divulgación, la eliminación de los componentes de capa decodificada se gestiona por separado para cada sub-unidad. Por ejemplo, la eliminación de un componente de vista de textura de la sub-unidad 810A no eliminará el componente de vista de profundidad asociado que tiene una resolución diferente (por ejemplo, un componente de vista de profundidad asociado, almacenado en la sub-unidad 810B). Asimismo, el codificador de vídeo puede configurarse adicionalmente para marcar los componentes de capa decodificada como no usados para referencia en la pluralidad de sub-unidades, donde el marcado se gestiona por separado para cada sub-unidad.

En un ejemplo adicional, el codificador de vídeo también puede configurarse para eliminar un componente de capa decodificada de una entre la pluralidad de sub-unidades 810A a 810D, donde la eliminación del componente de capa decodificada también elimina toda una unidad de acceso decodificado, relacionada con el componente de capa decodificada, de una entre la pluralidad de sub-unidades 810A a 810D. El codificador de vídeo puede configurarse adicionalmente para marcar un componente de capa decodificada en una entre la pluralidad de sub-unidades como no usado para referencia, donde el marcado del componente de capa decodificada también marca toda una unidad de acceso decodificado, relacionada con el componente de capa decodificada, como no usado para referencia.

Si bien las sub-unidades ejemplares en el ejemplo de la figura 8 se categorizaron por resolución espacial, otros ejemplos de la divulgación pueden usar diferente categorización. Por ejemplo, cada una entre la pluralidad de sub-unidades puede asociarse a una resolución espacial, formato de muestreo de croma o profundidad de bits diferente, o a cualquier combinación de resolución espacial, formato de muestreo de croma y profundidad de bits.

En otro ejemplo de la divulgación, un codificador de vídeo puede configurarse para decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada. El codificador de vídeo puede configurarse adicionalmente para almacenar los componentes de capa decodificada en una entre una pluralidad de sub-unidades, donde cada una entre la pluralidad de sub-unidades está asociada a una capa completamente reconstruida diferente. Por ejemplo, para la codificación de vídeo tridimensional, puede usarse una sub-unidad para almacenar componentes de vista de textura, mientras que puede usarse otra sub-unidad para almacenar componentes de vista de profundidad. Dado que los componentes de vista de textura y los componentes de vista de profundidad habitualmente tienen diferentes resoluciones, tal técnica permite la gestión independiente de capas decodificadas a diferentes resoluciones. Es decir, como el ejemplo de la figura 8, cada sub-unidad puede gestionarse por separado.

La figura 9 es un diagrama conceptual que muestra una DPB 900 configurada para almacenar diferentes componentes de capa decodificada en diferentes sub-unidades. Cada una de las sub-unidades 910A a 910D representa ubicaciones de almacenamiento para diferentes tipos de componentes de capa decodificada a diferentes resoluciones. Por ejemplo, la sub-unidad 910A puede usarse para almacenar componentes de vista de textura, mientras que la sub-unidad 910B puede usarse para almacenar componentes de vista de profundidad. De forma similar, la sub-unidad 910C puede usarse para almacenar las capas de base en un proceso de codificación de vídeo ajustable a escala, mientras que la sub-unidad 910D puede usarse para almacenar un nivel de capas de mejora en un proceso de codificación de vídeo ajustable a escala. Las sub-unidades adicionales pueden configurarse para el almacenamiento de niveles adicionales de capas de mejora o componentes adicionales de vista de textura. Cada una de las sub-unidades 910A a 910D contiene unidades individuales que están configuradas para almacenar un componente de capa decodificada asociado a una imagen.

El codificador de vídeo puede configurarse adicionalmente para eliminar los componentes de capa decodificada de la pluralidad de sub-unidades 910A a 910D, donde la eliminación se gestiona por separado para cada sub-unidad. El codificador de vídeo puede configurarse adicionalmente para marcar los componentes de capa decodificada como no usados para referencia en la pluralidad de sub-unidades 910A a 910D, donde el marcado se gestiona por separado para cada sub-unidad. El codificador de vídeo puede configurarse adicionalmente para eliminar un componente de capa decodificada de una entre la pluralidad de sub-unidades 910A a 910D, donde la eliminación del componente de capa decodificada también elimina la totalidad de una unidad de acceso decodificado, relacionada con el componente de capa decodificada, de una entre la pluralidad de sub-unidades 910A a 910D. El codificador de vídeo puede configurarse adicionalmente para marcar un componente de capa decodificada, en una entre la pluralidad de sub-unidades 910A a 910D, como no usado para referencia, donde el marcado del componente de capa decodificada también marca toda la unidad de acceso decodificado, relacionada con el componente de capa decodificada, como no usada para referencia.

En otro ejemplo de la divulgación, un codificador de vídeo puede configurarse para decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada, y almacenar los componentes de capa decodificada en una entre la pluralidad de sub-unidades de una memoria intermedia de imágenes decodificadas (DPB), de tal forma que cada sub-unidad de la DPB corresponda a un componente de capa decodificada que tenga la más alta resolución espacial. Es decir, el tamaño de cada una de las sub-unidades de la DPB se configura para que sea igual al componente de capa decodificada que tenga la más alta resolución. Cada una entre la pluralidad de sub-unidades de la DPB puede almacenar un componente de capa decodificada, independientemente de la resolución del componente de capa decodificada.

La figura 10 es un diagrama conceptual que muestra una DPB 1000 configurada para almacenar componentes de capa decodificada. Cada una de las sub-unidades 1010A a 1010D representa una ubicación de almacenamiento para un componente de capa decodificada, correspondiendo el tamaño de cada una de las sub-unidades a la resolución espacial del componente de capa decodificada que tenga la más alta resolución. El codificador de vídeo puede configurarse adicionalmente para eliminar un componente de capa decodificada de la DPB 1000, donde la eliminación del componente de capa decodificada también elimina la totalidad de la unidad de acceso decodificado, relacionada con el componente de capa decodificada, de la DPB 1000. El codificador de vídeo puede configurarse adicionalmente para marcar un componente de capa decodificada en la DPB 1000 como no usado para referencia, donde el marcado del componente de capa decodificada también marca la totalidad de la unidad de acceso decodificado, relacionada con el componente de capa decodificada, como no usada para referencia.

Para resumir, en un primer ejemplo de la divulgación, una única sub-unidad de una DPB almacena la unión de los componentes de capa decodificada de todas las capas completamente reconstruidas dentro de una unidad de acceso. Este primer ejemplo puede incluir una o más de las siguientes técnicas y/o estructuras:

- El tamaño de la sub-unidad individual de una DPB puede determinarse por la suma del número de muestras de todos los componentes en todos los componentes de capa decodificada.
- La eliminación de una imagen decodificada incluye la eliminación de toda una unidad de acceso decodificada.
- El marcado de la imagen decodificada como "no usada para referencia" incluye el marcado de toda la unidad de acceso decodificada como "no usada para referencia".

- 5 En un segundo ejemplo de la divulgación, la DPB puede incluir múltiples sub-unidades, cada una de las cuales está asociada a una resolución espacial diferente. Los componentes de capa decodificada para cada capa se gestionan por separado (incluyendo tanto el marcado como la eliminación). Además, durante la invocación del proceso de gestión de las DPB para una unidad de acceso en una capa específica, los componentes de capa decodificada de otras capas en la misma unidad de acceso pueden marcarse como "no usados para referencia" y eliminarse de las sub-unidades para esos componentes de capa decodificada. Como alternativa, además, cada sub-unidad puede asociarse a una combinación diferente de resoluciones espaciales, formatos de muestreo de croma y profundidades de bits.
- 10 En un tercer ejemplo de la divulgación, una DPB puede configurarse para incluir múltiples sub-unidades, cada una de las cuales está asociada a una capa completamente reconstruida diferente. Los componentes de capa decodificada para cada capa se gestionan por separado (incluyendo tanto el marcado como la eliminación). Además, durante la invocación del proceso de gestión de las DPB para una unidad de acceso en una capa específica, los componentes de capa decodificada de otras capas en la misma unidad de acceso pueden marcarse como "no usados para referencia" y eliminarse de las sub-unidades para esos componentes de capa decodificada.
- 15 En un cuarto ejemplo de la divulgación, cada sub-unidad en una DPB corresponde a un componente de capa decodificada de la capa con la más alta resolución. Cada sub-unidad se usa para almacenamiento de un componente de capa decodificada, independientemente de la resolución del componente de capa decodificada. Además, durante la invocación del proceso de gestión de las DPB para una unidad de acceso en una capa específica, los componentes de capa decodificada de otras capas en la misma unidad de acceso pueden marcarse como "no usados para referencia" y eliminarse de la DPB.
- 20 La figura 11 es un diagrama de flujo que muestra un procedimiento ejemplar de acuerdo a las técnicas de esta divulgación. Las técnicas de la figura 11 pueden ser ejecutadas por una o más unidades funcionales del codificador de vídeo 20 y/o del decodificador de vídeo 30, incluyendo, por ejemplo, la DPB 64 y la DPB 92.
- 25 En un ejemplo de la divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada (1100), almacenar los componentes de capa decodificada en una o más sub-unidades de una DPB (1110) y realizar un proceso de gestión de la DPB en las una o más sub-unidades, donde el proceso de gestión de la DPB se gestiona por separado para cada una de las una o más sub-unidades (1120). En un ejemplo de la divulgación, el proceso de gestión de la DPB comprende uno o más entre eliminar un componente de capa decodificada de la sub-unidad o marcar un componente de capa decodificada en la sub-unidad como no usado para referencia. En otro ejemplo de la divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 se configuran adicionalmente para realizar el mismo proceso de gestión de la DPB en un primer componente de capa decodificada en una unidad de acceso, y realizar el mismo proceso de gestión de la DPB en otros componentes de capa decodificada en la misma unidad de acceso.
- 30 En un ejemplo de la divulgación, los componentes de capa decodificada tienen al menos dos resoluciones espaciales diferentes. En este ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 están configurados adicionalmente para almacenar los componentes de capa decodificada en una de las una o más sub-unidades de la DPB, en base a la resolución especial de los componentes de capa decodificada, donde cada una de las una o más sub-unidades está asociada a una resolución espacial diferente. En otro ejemplo, cada una de las una o más sub-unidades está asociada a una combinación particular de resolución espacial, formato de muestreo de croma y profundidad de bits.
- 35 En otro ejemplo de la divulgación, los componentes de capa decodificada tienen al menos dos resoluciones espaciales diferentes, y cada una de las una o más sub-unidades está asociada a una capa completamente reconstruida diferente.
- 40 En otro ejemplo de la divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 están configurados adicionalmente para almacenar los componentes de capa decodificada en una de las una o más de las sub-unidades de la DPB, de tal forma que cada sub-unidad de la DPB corresponda al componente de capa decodificada que tenga la más alta resolución espacial. En un ejemplo, cada una de las una o más sub-unidades de la DPB almacena un componente de capa decodificada.
- 45 En otro ejemplo de la divulgación, las una o más sub-unidades comprenden una única sub-unidad, y el codificador de vídeo 20 y/o el decodificador de vídeo 30 están configurados adicionalmente para almacenar los componentes de capa decodificada en la única sub-unidad de la DPB, de tal forma que la única sub-unidad de la DPB contenga una unión de componentes de capa decodificada para una capa completamente reconstruida dentro de una unidad de acceso. En otro ejemplo, un tamaño de la única sub-unidad de la DPB se determina por la suma de varias muestras de todos los componentes en todos los componentes de capa decodificada.
- 50 La siguiente sección de la divulgación describirá una implementación ejemplar de un decodificador de referencia hipotético (HRD) (por ejemplo, para la HEVC, incluyendo extensiones ajustables a escala, de multi-vistas y
- 55
- 60
- 65

tridimensionales) para las primeras técnicas ejemplares de esta divulgación. Es decir, la siguiente sección se aplica a las técnicas de la divulgación, por lo que la única sub-unidad de una DPB almacena la unión de los componentes de capa decodificada de todas las capas completamente reconstruidas dentro de una unidad de acceso. Las cláusulas y sub-cláusulas mencionadas en esta especificación se refieren a la especificación del HRD en el Anexo C de la HEVC WD9 precitada.

4.1.1 HRD: Decodificador de referencia hipotético (Ejemplo N° 1)

A.1 Funcionamiento de la memoria intermedia de imágenes decodificadas (DPB)

Las especificaciones en esta sub-cláusula se aplican independientemente a cada conjunto de parámetros de DPB seleccionados como se especifica en la sub-cláusula C.1.

La memoria intermedia de imágenes decodificadas contiene memorias intermedias de almacenamiento de imágenes. Cada una de las memorias intermedias de almacenamiento de imágenes puede contener una imagen decodificada que se marca como "usada para referencia" o se guarda para una salida futura. Antes de la inicialización, la DPB está vacía (la plenitud de la DPB se fije en cero). Las siguientes etapas de las sub-cláusulas de esta sub-cláusula tienen lugar en la secuencia según se enumera a continuación.

Cada una de las memorias intermedias de almacenamiento de imágenes contiene todos los componentes de capa decodificada (también denominados como representaciones de capa en un códec ajustable a escala, todos los componentes de vista decodificados en la MV-HEVC o todos los componentes de vista de textura y de profundidad decodificados en la 3D-HEVC) de una unidad de acceso. Por lo tanto, cada imagen decodificada es la unidad de acceso decodificada. El tamaño de memoria de una memoria intermedia de almacenamiento de imágenes corresponde a la unidad de acceso decodificada que requiere el tamaño máximo de memoria intermedia para el almacenamiento de todos los componentes de capa decodificada entre todas las unidades de acceso en una secuencia de vídeo codificada. Por ejemplo, si diferentes unidades de acceso tienen diferentes números de componentes de capa, entonces el tamaño máximo de memoria intermedia puede ser necesario para el almacenamiento de las unidades de acceso que tengan, cada una, el número máximo de componentes de capa entre todas las unidades de acceso.

Una imagen decodificada se marca como "usada para referencia" si y sólo si uno o más componentes de capa están marcados como "usados para referencia".

Una imagen decodificada se marca como "no usada para referencia" si y sólo si todos los componentes de capa están marcados como "no usados para referencia".

Además, pueden tener aplicación restricciones, como se describe en la sección 4.1.3.

Como alternativa:

Una imagen decodificada se marca como "usada para referencia" si y sólo si el componente de capa con el mayor valor de `id_capa` está marcado como "usado para referencia".

Una imagen decodificada se marca como "no usada para referencia" si y sólo si el componente de capa con el mayor valor de `id_capa` está marcado como "no usado para referencia".

Además, pueden tener aplicación restricciones, como se describe en la sección 4.1.3.

En el caso en que los componentes de múltiples capas de la misma unidad de acceso pueden tener diferentes estados de marcado, el estado de marcado de cada componente de capa se conoce después de aplicar un proceso de marcado de imagen decodificada para cada componente de capa, de un modo similar al proceso de marcado de imagen decodificada en la HEVC.

En una alternativa, también pueden marcarse conjuntamente múltiples componentes de capa; en este caso, se invoca un proceso similar al proceso de marcado de imagen decodificada en la HEVC para toda la unidad de acceso.

Una imagen es refresco de decodificador instantáneo (IDR) si todos los componentes de capa tienen `tipo_unidad_nal` igual a `IDR_W_DLP` o `IDR_N_LP`.

Una imagen es un acceso a enlace roto (BLA) si todos los componentes de capa tienen `tipo_unidad_nal` igual a `BLA_W_LP`, `BLA_W_DLP` o `BLA_N_LP`.

Cuando `tipo_unidad_nal` tiene un valor en el intervalo de 16 a 23, inclusive (imagen de acceso aleatorio (RAP)), el `tipo_tajada` será igual a 2 para la capa base (con `id_capa` igual a 0), pero puede ser igual a otros valores si `id_capa`

es mayor que 0.

A.1.1 Eliminación de imágenes de la DPB

5 La eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente el encabezado de tajada de la primera tajada de la imagen actual) tiene lugar instantáneamente en el momento de la eliminación de la CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene la imagen actual) y avanza como se indica a continuación.

10 Se invoca el proceso de decodificación para el conjunto de imágenes de referencia como se especifica en la sub-cláusula 8.3.2.

Si la imagen actual es una imagen de IDR o una imagen de BLA, se aplica lo siguiente:

15 Cuando la imagen de IDR o BLA no es la primera imagen decodificada y el valor de ancho_imagen_en_muestras_luma o altura_imagen_en_muestras_luma o sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid] para cualquier capa de la unidad de acceso actual es diferente al valor del ancho de la imagen en muestras de luma o a la altura de la imagen en muestras de luma o a sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid], obtenidas al decodificar la unidad de acceso anterior, respectivamente, el HRD deduce que el indicador_nº_salida_de_imágenes_anteriores es igual a 1, independientemente del valor real del indicador_nº_salida_de_imágenes_anteriores.

20 Cuando el indicador_nº_salida_de_imágenes_anteriores es igual a 1 o se deduce que es igual a 1, todas las memorias intermedias de almacenamiento de imágenes en la DPB son vaciadas sin salida de las imágenes que contienen, y la plenitud de la DPB se fija en 0.

Todas las imágenes k en la DPB, para las que todas las siguientes condiciones son verdaderas, se eliminan de la DPB:

- 30 - la imagen k está marcada como "no usada para referencia",
- la imagen k tiene IndicadorSalidaImagen igual a 0 o su hora de salida de la DPB es menor o igual a la hora de eliminación de la CPB de la primera unidad de decodificación (representada como unidad de decodificación m) de la imagen actual n; es decir $to,dpb(k) \leq tr(m)$

35 Cuando una imagen se elimina de la DPB, la plenitud de la DPB se reduce en uno.

A.1.2 Salida de imagen

40 Lo siguiente sucede instantáneamente en el momento de eliminación de la CPB de la unidad de acceso n, $tr(n)$.

Cuando la imagen n tiene el IndicadorSalidaImagen igual a 1, su hora de salida de la DPB $to,dpb(n)$ se obtiene por

$$45 \quad to,dpb(n) = tr(n) + tc * retardo_salida_dpb_imagen(n) \quad (C 15)$$

donde $retardo_salida_dpb_imagen(n)$ es el valor de $retardo_salida_dpb_imagen$ especificado en el mensaje SEI de temporización de imagen asociado a la unidad de acceso n.

La salida de la imagen actual se especifica como se indica a continuación.

- 50 - Si IndicadorSalidaImagen es igual a 1 y $to,dpb(n) = tr(n)$, la imagen actual se emite.
- Por el contrario, si IndicadorSalidaImagen es igual a 0, la imagen actual no se emite, sino que se almacenará en la DPB como se especifica en la sub-cláusula C.3.4.
- 55 - De otro modo, (IndicadorSalidaImagen es igual a 1 y $to,dpb(n) > tr(n)$), la imagen actual se emite posteriormente y se almacenará en la DPB (como se especifica en la sub-cláusula C.3.4) y se emite en el momento $to,dpb(n)$, a menos que se indique que no se emita, por la decodificación o la deducción de que el indicador_nº_salida_de_imágenes_anteriores es igual a 1 en un momento que antecede a $to,dpb(n)$.

60 Al emitirse, la imagen se recortará, usando la ventana de recorte de conformidad especificada en el conjunto activo de parámetros de secuencia.

65 Cuando la imagen n es una imagen que se emite y no es la última imagen del flujo de bits que se emite, el valor de $Dto,dpb(n)$ se define como:

$$\text{Dto,dpb}(n) = \text{to,dpb}(nn) - \text{to,dpb}(n) \quad (\text{C } 16)$$

donde nn indica la imagen que sigue a la imagen n en orden de salida y tiene el IndicadorSalidaImagen igual a 1.

5 **A.1.3 Marcado y almacenamiento de imágenes decodificadas actuales**

Lo siguiente sucede instantáneamente en el momento de eliminación de la CPB de la unidad de acceso n, tr(n).

10 La imagen decodificada actual se almacena en la DPB en una memoria intermedia de almacenamiento de imágenes vacía, la plenitud de la DPB se aumenta en uno y la imagen actual se marca como "no usada para referencia a corto plazo".

A.2 Conformidad del flujo de bits

15 Tienen aplicación las especificaciones en la sub-cláusula C.2.

A.3 Conformidad del decodificador

A.3.1 General

20 Las especificaciones en C.5.1 tienen aplicación con los siguientes agregados.

25 Cada una de las memorias intermedias de almacenamiento de imágenes contiene todos los componentes de capa decodificada (también denominados representaciones de capa en un códec ajustable a escala, todos los componentes de vista decodificados en la MV-HEVC o todos los componentes de vista de textura y de profundidad decodificados en la 3D-HEVC) de una unidad de acceso. Por lo tanto, cada imagen decodificada es la unidad de acceso decodificada. El tamaño de memoria de una memoria intermedia de almacenamiento de imágenes corresponde a la unidad de acceso decodificada que requiere el tamaño máximo de memoria intermedia para el almacenamiento de todos los componentes de capa decodificada entre todas las unidades de acceso en una secuencia de vídeo codificada. Por ejemplo, si diferentes unidades de acceso tienen diferentes números de componentes de capa, entonces el tamaño máximo de memoria intermedia puede ser necesario para el almacenamiento de las unidades de acceso que tengan, cada una, el número máximo de componentes de capa entre todas las unidades de acceso.

35 Lo siguiente es un ejemplo de una alternativa para el estado de marcado de los componentes de capa. Una imagen decodificada se marca como "usada para referencia" si y sólo si uno o más componentes de capa están marcados como "usados para referencia". Una imagen decodificada se marca como "no usada para referencia" si y sólo si todos los componentes de capa están marcados como "no usados para referencia". Además, pueden tener aplicación restricciones, como se describe en la sección 4.1.3.

40 Lo siguiente es un ejemplo de otra alternativa para el estado de marcado de los componentes de capa. Una imagen decodificada se marca como "usada para referencia" si y sólo si el componente de capa con el mayor valor de id_capa está marcado como "usado para referencia". Una imagen decodificada se marca como "no usada para referencia" si y sólo si el componente de capa con el mayor valor de id_capa está marcado como "no usado para referencia". Además, pueden tener aplicación restricciones, como se describe en la sección 4.1.3.

Una imagen es un IDR si todos los componentes de capa tienen tipo_unidad_nal igual a IDR_W_DLP o IDR_N_LP.

50 Una imagen es un BLA si todos los componentes de capa tienen tipo_unidad_nal igual a BLA_W_LP, BLA_W_DLP o BLA_N_LP.

Cuando tipo_unidad_nal tiene un valor en el intervalo de 16 a 23, inclusive (imagen RAP), el tipo_tajada será igual a 2 para la capa base (con id_capa igual a 0), pero puede ser igual a otros valores si el Id_capa es mayor que 0.

55 **A.3.2 Funcionamiento de la DPB de orden de salida**

60 La memoria intermedia de imágenes decodificadas contiene memorias intermedias de almacenamiento de imágenes. Cada una de las memorias intermedias de almacenamiento de imágenes contiene una imagen decodificada que se marca como "usada para referencia" o se guarda para una salida futura. En el inicio del HRD, la DPB está vacía. Las siguientes etapas tienen lugar en el orden según se enumera a continuación.

A.3.3 Salida y eliminación de imágenes de la DPB

65 La salida y eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente el encabezado de tajada de la primera tajada de la imagen actual) tiene lugar instantáneamente cuando la primera unidad de decodificación de la unidad de acceso que contiene la imagen actual

se elimina de la CPB, y avanza como se indica a continuación.

Se invoca el proceso de decodificación para el conjunto de imágenes de referencia como se especifica en la sub-cláusula 8.3.2.

- 5 - Si la imagen actual es una imagen de IDR o una imagen de BLA, se aplica lo siguiente.
- 10 1. Cuando la imagen de IDR o de BLA no es la primera imagen decodificada y el valor de ancho_imagen_en_muestras_luma o altura_imagen_en_muestras_luma o sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid] de cada capa de la unidad de acceso actual es diferente al valor del ancho de la imagen en muestras de luma o a la altura de la imagen en muestras de luma o a sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid], de cada capa obtenida para la anterior unidad de acceso, respectivamente, el HRD deduce que el indicador_nº_salida_de_imágenes_anteriores es igual a 1, independientemente del valor real del indicador_nº_salida_de_imágenes_anteriores.
- 15 2. NOTA - Las implementaciones del decodificador deberían intentar manipular la imagen o cambios de tamaño de la DPB más gentilmente que el HRD con respecto a cambios en ancho_imagen_en_muestras_luma, altura_imagen_en_muestras_luma o sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid].
- 20 3. Cuando el indicador_nº_salida_de_imágenes_anteriores es igual a 1 o se deduce que es igual a 1, todas las memorias intermedias de almacenamiento de imágenes en la DPB son vaciadas sin salida de las imágenes que contienen.
- 25 4. Cuando el indicador_nº_salida_de_imágenes_anteriores no es igual a 1 y no se deduce que sea igual a 1, las memorias intermedias de almacenamiento de imágenes que contienen una imagen que se marca como "no necesaria para la salida" y "no marcada para referencia" se vacían (sin salida), y todas las memorias intermedias de almacenamiento de imágenes no vacías en la DPB se vacían invocando repetidamente el proceso de "batacazo" especificado en la sub-cláusula C.5.3.1.
- 30 - De otro modo (la imagen actual no es una imagen de IDR o una imagen de BLA), las memorias intermedias de almacenamiento de imágenes que contienen una imagen que se marca como "no necesaria para la salida" y "no usada para referencia" se vacían (sin salida). Cuando una o más de las siguientes condiciones son verdaderas, el proceso de "batacazo" especificado en la sub-cláusula C.5.3.1 se invoca repetidamente hasta que haya una memoria intermedia de almacenamiento de imágenes vacía para almacenar la imagen decodificada actual.
- 35 1. El número de imágenes en la DPB que se marcan como "necesarias para la salida" es mayor que sps_máx_núm_reordenamiento_imágenes [MáximoTid],
- 40 2. El número de imágenes en la DPB es igual a sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid].
- 45 Ha de apreciarse que el ancho_imagen_en_muestras_luma o la altura_imagen_en_muestras_luma o la sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid] de cada capa de una unidad de acceso puede señalizarse en los conjuntos de parámetros de secuencia o bien en el conjunto de parámetros de vídeo activo mencionado por la unidad de acceso actual.

50 A.3.3.1 Proceso de "Batacazo"

El proceso de "batacazo" se invoca en los siguientes casos.

- 55 - La imagen actual es una imagen de IDR o una imagen de BLA y el indicador_nº_salida_de_imágenes_anteriores no es igual a 1 y no se deduce que sea igual a 1, como se especifica en la sub-cláusula C.5.2.
- El número de imágenes en la DPB que están marcadas como "necesarias para salida" es mayor que sps_máx_núm_reordenamiento_imágenes [IdTemporal], como se especifica en la sub-cláusula C.5.2.
- 60 - El número de imágenes en la DPB con IdTemporal menor o igual que el IdTemporal de la imagen actual es igual a sps_máx_imagen_dec_almacenamiento_temporal [IdTemporal], como se especifica en la sub-cláusula C.5.2.

El proceso de "batacazo" consiste en las siguientes etapas ordenadas:

- 65 1. La imagen que está primero para la salida se selecciona como la única que tiene el valor más pequeño de

ValCntOrdenImagen de todas las imágenes en la DPB marcadas como "necesaria para la salida".

2. La imagen se recorta, usando el rectángulo de recorte especificado en el conjunto activo de parámetros de secuencia para la imagen, la imagen recortada se emite y la imagen se marca como "no necesaria para la salida".

3. Si la memoria intermedia de almacenamiento de imágenes que incluía la imagen que se recortó y se emitió contiene una imagen marcada como "no usada para referencia", la memoria intermedia de almacenamiento de imágenes se vacía.

A.3.4 Decodificación, marcado y almacenamiento de imágenes

Lo siguiente tiene lugar instantáneamente cuando la última unidad de decodificación de la unidad de acceso n que contiene la imagen actual se elimina de la CPB.

La imagen actual se considera como decodificada después de que se decodifique la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en una memoria temporal vacía de almacenamiento de imágenes en la DPB, y tiene aplicación lo siguiente.

- Si la imagen decodificada actual tiene el IndicadorSalidaImagen igual a 1, se marca como "necesaria para la salida".
- De lo contrario (la imagen decodificada actual tiene el IndicadorSalidaImagen igual a 0), se marca como "no necesaria para la salida".

La imagen decodificada actual se marca como "utilizada para referencia a corto plazo".

4.1.2 Tamaño máximo de imagen de luma en el Anexo A: Perfiles, escalones y niveles

MaxLumaPS, como se especifica en la Tabla A1, puede corresponder a la suma máxima del tamaño de imagen de luma (AUMaxLumaPS) de cada capa en la unidad de acceso.

Como alternativa, puede usarse el mismo valor del nivel en la especificación de base de la HEVC para indicar extensiones, donde una capa requiere el mismo nivel. En este caso, pueden tener aplicación los siguientes dos aspectos:

- MaxLumaPS corresponde al tamaño máximo de imagen de luma de la capa más alta en una extensión ajustable a escala o a la vista de textura con la más alta resolución espacial en una extensión de multi-vista o 3DV;
- Un factor de ajuste a escala se introduce y se calcula en base al AUMaxLumaPS/MaxLumaPS y teniendo en cuenta al obtener, por ejemplo, el tamaño de la DPB y otras restricciones relacionadas con el nivel.

Como se ha mencionado anteriormente, el factor de ajuste a escala puede obtenerse como la suma de números de muestra de todos los componentes de capa, dividida entre los números de muestra máximos de un componente de capa. Como alternativa, el factor de ajuste a escala puede calcularse teniendo en cuenta el formato de muestreo de croma y los valores de profundidad de bits tanto para los componentes de luma como para los de croma.

Pueden usarse diversos enfoques para definir el tamaño de la DPB que puede estar relacionado con el factor de ajuste a escala.

En un enfoque, el "tamaño de la DPB" es indicativo del tamaño de la memoria física y, por lo tanto, el factor de ajuste a escala rebaja en escala el número de imágenes que pueden almacenarse.

Como alternativa, en un enfoque diferente, el "tamaño de la DPB" es indicativo de cuántas AU pueden almacenarse; por lo tanto, el factor de ajuste a escala aumenta a escala el tamaño de la memoria física (en comparación con tener solamente la capa de base).

4.1.3 Restricciones para marcar el estado de los componentes de capa del conjunto de imágenes de referencia

Alternativa N° 1

En una extensión escalable de la HEVC, el conjunto de imágenes de referencia, excluyendo las imágenes que se generan durante la decodificación de la misma unidad de acceso, de cada capa se restringe como se indica a continuación:

El RPS de cada componente de capa i , representado como RPS_i , es un superconjunto de cualquier RPS_j , en el que el componente de capa j (de la misma unidad de acceso) tiene un id_capa más pequeño, lo que significa que cualquier identificación de imagen (POC) que se incluya en el RPS_j también se incluirá en el RPS_i .

Como alternativa, ese RPS para cada componente de capa debería ser igual. En este caso, la señalización basada únicamente en la AU del RPS es necesaria para referencias de inter-predicción. La señalización basada en la AU puede hacerse de manera que únicamente el componente de capa de base contenga los elementos sintácticos del RPS. Como alternativa, la señalización basada en la AU puede hacerse de manera que solamente cada componente de capa independiente contenga elementos sintácticos del RPS.

Ha de apreciarse que un conjunto de imágenes de referencia incluye $RefPicSetLtCurr$, $RefPicSetLtFoll$, $RefPicSetStCurrBefore$, $RefPicSetStCurrAfter$ y $RefPicSetStFoll$.

De forma similar, en la MV-HEVC, el conjunto de imágenes de referencia, excluyendo las imágenes de referencia que se generaron durante la decodificación de la misma unidad de acceso, de cada componente de vista se restringe como se indica a continuación:

El RPS del componente de capa i , representado como RPS_i , es un superconjunto de cualquier RPS_j , donde el componente de capa j (de la misma unidad de acceso) tiene un id_capa más pequeño, lo que significa que cualquier identificación de imagen (POC) que esté en el RPS_j no se incluirá en el RPS_i .

Como alternativa, ese RPS para cada componente de capa debería ser igual. En este caso, la señalización basada únicamente en la AU del RPS es necesaria para referencias de inter-predicción. La señalización basada en la AU puede hacerse de manera que solamente el componente de capa de base contenga los elementos sintácticos del RPS. Como alternativa, la señalización basada en la AU puede hacerse de manera que solamente cada componente de capa independiente contenga elementos sintácticos del RPS.

En la 3D-HEVC, pueden tener aplicación las siguientes restricciones:

El RPS de cada componente de vista de textura i , representado como RPS_i , es un superconjunto de cualquier RPS_j , donde el componente de vista de textura j (de la misma unidad de acceso) tiene un id_capa más pequeño.

El RPS de cada componente de vista de profundidad i , representado como RPS_i , es un superconjunto de cualquier RPS_j , donde el componente de vista de profundidad j (de la misma unidad de acceso) tiene un id_capa más pequeño.

El RPS de un componente de vista de textura, representado como $RPSt$, es un superconjunto del $RPSd$, el RPS del componente de vista de profundidad de la misma vista.

Como alternativa, independientemente de si un componente de vista es de textura o profundidad, el RPS del componente de capa i , representado como RPS_i , es un superconjunto de cualquier RPS_j , en el que el componente de capa j (de la misma unidad de acceso) tiene un id_capa más pequeño.

Alternativa N° 2

Todos los componentes de capa de la misma unidad de acceso comparten el mismo RPS.

Por ejemplo, el RPS para cada componente de capa debería ser igual. Además, como alternativa, solamente la señalización basada en la AU del RPS es necesaria para referencias de inter-predicción. La señalización basada en la AU puede hacerse de manera que solamente el componente de capa de base contenga los elementos sintácticos del RPS. Como alternativa, la señalización basada en la AU puede hacerse de manera que solamente cada componente de capa independiente contenga elementos sintácticos del RPS.

La siguiente sección de la divulgación describirá otra implementación ejemplar del decodificador de referencia hipotética (HRD) (por ejemplo, para la HEVC, incluyendo extensiones ajustables a escala, de multi-vista y tridimensionales) para las segundas técnicas ejemplares de esta divulgación. Es decir, la siguiente sección se aplica a las técnicas de la divulgación, por las que la DPB consiste en múltiples sub-DPB, cada una está asociada a una resolución espacial diferente y los componentes de capa decodificada para cada capa se gestionan por separado. Las cláusulas y sub-cláusulas mencionadas en esta especificación se refieren a la especificación del HRD en el Anexo C de la HEVC WD9 precitada.

4.2.1 HRD: Decodificador de referencia hipotético (Ejemplo N° 2)

A.4 Funcionamiento de la memoria intermedia de imágenes decodificadas (DPB)

Las especificaciones en esta sub-cláusula se aplican independientemente a cada conjunto de parámetros de la DPB seleccionados como se especifica en la sub-cláusula C.1.

5 La memoria intermedia de imágenes decodificadas contiene memorias intermedias de almacenamiento de imágenes. Cada una de las memorias intermedias de almacenamiento de imágenes puede contener un componente de capa decodificada que se marca como "usada para referencia" o se guarda para una salida futura.

10 La memoria intermedia de imágenes decodificadas consiste en una o más sub-memorias intermedias de imágenes decodificadas (SDPB), cada una asociada a una resolución espacial diferente. Antes de la inicialización, cada SDPB está vacía (la plenitud de la SDPB se fija en cero).

15 Las siguientes etapas de las sub-cláusulas de esta sub-cláusula suceden en la secuencia como se indica a continuación, y se invocan repetidamente, cada vez para una capa, en el orden de decodificación, y durante la invocación, "DPB" se reemplaza con "SDPB" y la "imagen decodificada" se reemplaza con el "componente de capa decodificada".

Una imagen es un IDR si todos los componentes de capa tienen el tipo_unidad_nal igual a IDR_W_DLP o a IDR_N_LP.

20 Una imagen es un BLA si todos los componentes de capa tienen el tipo_unidad_nal igual a BLA_W_LP, BLA_W_DLP o BLA_N_LP.

25 Cuando el tipo_unidad_nal tiene un valor en el intervalo de 16 a 23, inclusive (imagen RAP), el tipo_tajada será igual a 2 para la capa base (con id_capa igual a 0), pero puede ser igual a otros valores si el id_capa es mayor que 0.

A.4.1 Eliminación de imágenes de la DPB

30 La eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente el encabezado de tajada de la primera tajada de la imagen actual) tiene lugar instantáneamente en el momento de la eliminación de la CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene la imagen actual) y avanza como se indica a continuación.

35 Se invoca el proceso de decodificación para el conjunto de imágenes de referencia, como se especifica en la sub-cláusula 8.3.2.

Si la imagen actual es una imagen de IDR o una imagen de BLA, tiene aplicación lo siguiente:

40 1. Cuando la imagen de IDR o de BLA no es la primera imagen decodificada y el valor del ancho_imagen_en_muestras_luma o la altura_imagen_en_muestras_luma o la sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid] para cualquier capa de la unidad de acceso actual es diferente al valor del ancho_imagen_en_muestras_luma o de la altura_imagen_en_muestras_luma o del sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid] que se obtienen al decodificar la unidad de acceso anterior, respectivamente, el HRD deduce que el indicador_nº_salida_de_imágenes_anteriores es igual a 1, independientemente del valor real del indicador_nº_salida_de_imágenes_anteriores.

50 NOTA - Las implementaciones del decodificador deberían intentar manipular la imagen o los cambios de tamaño de la DPB más gentilmente que el HRD con respecto a cambios en el ancho_imagen_en_muestras_luma o la altura_imagen_en_muestras_luma o el sps_máx_imagen_dec_almacenamiento_temporal [i].

55 2. Cuando el indicador_nº_salida_de_imágenes_anteriores es igual a 1 o se deduce que es igual a 1, todas las memorias intermedias de almacenamiento de imagen en la DPB son vaciadas sin salida de las imágenes que contienen, y la plenitud de la DPB se fija en 0.

Todas las imágenes k en la DPB, para las cuales todas las siguientes condiciones son verdaderas, se eliminan de la DPB:

- 60 - la imagen k se marca como "no usada para referencia",
- la imagen k tiene el IndicadorSalidaImagen igual a 0 o su hora de salida de la DPB es menor o igual a la hora de eliminación de la CPB de la primera unidad de decodificación (representada como unidad de decodificación m) de la imagen actual n; es decir $to,dpb(k) \leq tr(m)$

65 Cuando una imagen se elimina de la DPB, la plenitud de la DPB se reduce en uno.

A.4.2 Salida de imágenes

Lo siguiente sucede instantáneamente en el momento de la eliminación de la CPB de la unidad de acceso n , $tr(n)$.

5 Cuando la imagen n tiene el `IndicadorSalidaImagen` igual a 1, su hora de salida de la DPB $to,dpb(n)$ se obtiene por
 $to,dpb(n) = tr(n) + tc * retardo_salida_dpb_imagen(n)$ (C 15)

10 donde el `retardo_salida_dpb_imagen(n)` es el valor del `retardo_salida_dpb_imagen(n)` especificado en el mensaje SEI de temporización de imagen asociado a la unidad de acceso n .

La salida de la imagen actual se especifica como se indica a continuación.

- 15 - Si `IndicadorSalidaImagen` es igual a 1 y $to,dpb(n) = tr(n)$, la imagen actual se emite.
- Por el contrario, si `IndicadorSalidaImagen` es igual a 0, la imagen actual no se emite, sino que se almacenará en la DPB como se especifica en la sub-cláusula C.3.4.
- 20 - De otro modo, (`IndicadorSalidaImagen` es igual a 1 y $to,dpb(n) > tr(n)$), la imagen actual se emite posteriormente y se almacenará en la DPB (como se especifica en la sub-cláusula C.3.4) y se emite en el momento $to,dpb(n)$, a menos que se indique que no se emita, por la decodificación o la deducción de que el `indicador_nº_salida_de_imágenes_anteriores` es igual a 1 en un momento que antecede a $to,dpb(n)$.

25 Al emitirse, la imagen se recortará, usando la ventana de recorte de conformidad especificada en el conjunto activo de parámetros de secuencia.

Cuando la imagen n es una imagen que se emite y no es la última imagen del flujo de bits que se emite, el valor de $Dto,dpb(n)$ se define como:

30 $Dto,dpb(n) = to,dpb(nn) - to,dpb(n)$ (C 16)

donde nn indica la imagen que sigue a la imagen n en orden de salida y que tiene el `IndicadorSalidaImagen` igual a 1.

35 **A.4.3 Marcado y almacenamiento de imágenes decodificadas actuales**

Lo siguiente sucede instantáneamente en el momento de la eliminación de la CPB de la unidad de acceso n , $tr(n)$.

40 La imagen decodificada actual se almacena en la DPB en una memoria intermedia vacía de almacenamiento de imágenes, la plenitud de la DPB se aumenta en uno y la imagen actual se marca como "no usada para referencia a corto plazo".

A.5 Conformidad del flujo de bits

45 Se aplican las especificaciones en la sub-cláusula C.2.

A.6 Conformidad del decodificador

50 **A.6.1. General**

Las especificaciones en C.5.1 se aplican con los siguientes agregados.

55 La memoria intermedia de imágenes decodificadas contiene memorias intermedias de almacenamiento de imágenes. Cada una de las memorias intermedias de almacenamiento de imágenes puede contener un componente de capa decodificada que se marca como "usada para referencia" o se guarda para una salida futura.

60 La memoria intermedia de imágenes decodificadas puede incluir una o más sub-memorias intermedias de imágenes decodificadas (SDPB), cada una asociada a una resolución espacial diferente. Antes de la inicialización, cada SDPB está vacía (la plenitud de la SDPB se fija en cero).

Las siguientes etapas de las sub-cláusulas de esta sub-cláusula suceden en la secuencia como se indica a continuación, y se invocan repetidamente, cada vez para una capa, en el orden de decodificación, y durante la invocación, "DPB" se reemplaza con "SDPB" y la "imagen decodificada" se reemplaza con el "componente de capa decodificada".

65 Una imagen es un IDR si es un componente de capa que tiene el `tipo_unidad_nal` igual a `IDR_W_DLP` o a

IDR_N_LP.

Una imagen es un BLA si es un componente de capa que tiene el tipo_unidad_nal igual a BLA_W_LP, BLA_W_DLP o BLA_N_LP.

5 Cuando el tipo_unidad_nal tiene un valor en el intervalo de 16 a 23, inclusive (imagen RAP), el tipo_tajada será igual a 2 para la capa base (con id_capa igual a 0), pero puede ser igual a otros valores si el id_capa es mayor que 0.

A.6.2 Funcionamiento de la DPB de orden de salida

10 La memoria intermedia de imágenes decodificadas contiene memorias intermedias de almacenamiento de imágenes. Cada una de las memorias intermedias de almacenamiento de imágenes contiene una imagen decodificada que se marca como "usada para referencia" o se guarda para una salida futura. En el inicio del HRD, la DPB está vacía. Las siguientes etapas tienen lugar en el orden según se enumeran a continuación.

A.6.3 Salida y eliminación de imágenes de la DPB

15 La salida y eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente el encabezado de tajada de la primera tajada de la imagen actual) tiene lugar instantáneamente cuando la primera unidad de decodificación de la unidad de acceso que contiene la imagen actual se elimina de la CPB y avanza como se indica a continuación.

20 Se invoca el proceso de decodificación para el conjunto de imágenes de referencia, como se especifica en la sub-cláusula 8.3.2.

25 - Si la imagen actual es una imagen de IDR o una imagen de BLA, tiene aplicación lo siguiente.

30 1. Cuando la imagen de IDR o de BLA no es la primera imagen decodificada y el valor del ancho_imagen_en_muestras_luma o de la altura_imagen_en_muestras_luma o del sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid] de cada capa de la unidad de acceso actual es diferente al valor del ancho_imagen_en_muestras_luma o de la altura_imagen_en_muestras_luma o del sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid] de cada capa obtenida para la unidad de acceso anterior, respectivamente, el HRD deduce que el indicador_nº_salida_de_imágenes_anteriores es igual a 1, independientemente del valor real del indicador_nº_salida_de_imágenes_anteriores.

35 2. Cuando el indicador_nº_salida_de_imágenes_anteriores es igual a 1 o se deduce que es igual a 1, todas las memorias intermedias de almacenamiento de imágenes en la DPB son vaciadas sin salida de las imágenes que contienen.

40 3. Cuando el indicador_nº_salida_de_imágenes_anteriores no es igual a 1 y no se deduce que sea igual a 1, las memorias intermedias de almacenamiento de imagen que contienen una imagen que está marcada como "no necesaria para la salida" y "no usada para referencia" se vacían (sin salida), y todas las memorias intermedias no vacías de almacenamiento de imágenes en la DPB se vacían invocando repetidamente el proceso de "batacazo" especificado en la sub-cláusula C.5.3.1.

45 - De otro modo (la imagen actual no es una imagen de IDR o una imagen de BLA), las memorias intermedias de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no usada para referencia" se vacían (sin salida). Cuando una o más de las siguientes condiciones son verdaderas, el proceso de "batacazo" especificado en la sub-cláusula C.5.3.1 se invoca repetidamente hasta que haya una memoria intermedia vacía de almacenamiento de imágenes para almacenar la imagen decodificada actual.

50 4.El número de imágenes en la DPB que se marcan como "necesarias para la salida" es mayor que el sps_máx_núm_reordenamiento_imágenes [MáximoTid],

55 5.El número de imágenes en la DPB es igual al sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid].

60 Ha de apreciarse que el ancho_imagen_en_muestras_luma o la altura_imagen_en_muestras_luma o el sps_máx_imagen_dec_almacenamiento_temporal [MáximoTid] de cada capa de una unidad de acceso puede señalizarse en los conjuntos de parámetros de secuencia o bien en el conjunto activo de parámetros de vídeo mencionado por la unidad de acceso anterior.

A.6.3.1 Proceso de "Batacazo"

65 El proceso de "batacazo" se invoca en los siguientes casos.

- La imagen actual es una imagen de IDR o una imagen de BLA y el indicador_nº_salida_de_imágenes_anteriores no es igual a 1 y no se deduce que sea igual a 1, como se especifica en la sub-cláusula C.5.2.
- El número de imágenes en la DPB que están marcadas como "necesarias para salida" es mayor que el sps_máx_núm_reordenamiento_imágenes [IdTemporal], como se especifica en la sub-cláusula C.5.2.
- El número de imágenes en la DPB con IdTemporal menor o igual al IdTemporal de la imagen actual es igual a sps_máx_imagen_dec_almacenamiento_temporal [IdTemporal], como se especifica en la sub-cláusula C.5.2.

El proceso de "batacazo" puede incluir las siguientes etapas ordenadas:

1. La imagen que está primera para la salida se selecciona como la única que tiene el valor más pequeño de ValCntOrdenImagen de todas las imágenes en la DPB marcadas como "necesaria para la salida".
2. La imagen se recorta, usando el rectángulo de recorte especificado en el conjunto activo de parámetros de secuencia para la imagen, la imagen recortada se emite y la imagen se marca como "no necesaria para la salida".
3. Si la memoria intermedia de almacenamiento de imágenes que incluía la imagen que se recortó y se emitió contiene una imagen marcada como "no usada para referencia", la memoria intermedia de almacenamiento de imágenes se vacía.

A.6.4 Decodificación, marcado y almacenamiento de imágenes

Lo siguiente tiene lugar instantáneamente cuando la última unidad de decodificación de la unidad de acceso n que contiene la imagen actual se elimina de la CPB.

La imagen actual se considera como decodificada después de que se decodifique la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en una memoria vacía de almacenamiento de imágenes en la DPB, y tiene aplicación lo siguiente.

- Si la imagen decodificada actual tiene el IndicadorSalidaImagen igual a 1, se marca como "necesaria para la salida".
- De lo contrario (la imagen decodificada actual tiene el IndicadorSalidaImagen igual a 0), se marca como "no necesaria para la salida".

La imagen decodificada actual se marca como "utilizada para referencia a corto plazo".

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

A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión se denomina debidamente medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por

ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles, no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos ópticamente con láseres. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones lógicas programables en el terreno (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de hardware dedicado y/o módulos de software configurados para la codificación y la descodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación se pueden implementar en una gran variedad de dispositivos o aparatos, incluyendo un equipo de mano inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Varios componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas descritas, pero no requieren necesariamente la realización mediante diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec, o proporcionarse por una colección de unidades de hardware inter-operativas, incluyendo uno o más procesadores como se ha descrito anteriormente, junto con el software y / o firmware adecuado.

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

REIVINDICACIONES

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

5 decodificar (1100) datos de vídeo para producir una pluralidad de componentes de capa decodificada, siendo cada componente de capa decodificada de un tipo, en el que los tipos de la pluralidad de componentes de capa decodificada incluyen al menos uno entre una vista de textura de datos de vídeo de multi-vista, una vista en profundidad de datos de vídeo de multi-vista, una capa base de datos de vídeo ajustables a escala, o cada una de una o más capas de mejora de datos de vídeo ajustables a escala; caracterizado por

10 almacenar (1110) la pluralidad de componentes de capa decodificada en sub-unidades de una memoria intermedia de imágenes decodificadas (DPB), de tal forma que un tipo diferente de componente de capa decodificada se almacene en cada sub-unidad respectiva; y

15 realizar (1120) un proceso de gestión de la DPB en las sub-unidades, en el que el proceso de gestión de la DPB se gestiona independientemente para cada una de las sub-unidades.
2. El procedimiento de la reivindicación 1, en el que el proceso de gestión de la DPB comprende uno o más entre eliminar un componente de capa decodificada, entre la pluralidad de componentes de capa decodificada, de la sub-unidad respectiva y marcar un componente de capa decodificada, entre la pluralidad de componentes de capa decodificada en la sub-unidad respectiva, como no usado para referencia.
3. El procedimiento de la reivindicación 2, en el que realizar el proceso de gestión de la DPB en las sub-unidades comprende:

25 realizar el proceso de gestión de la DPB en un primer componente de capa decodificada en una unidad de acceso; y

30 realizar el proceso de gestión de la DPB en otros componentes de capa decodificada en la unidad de acceso.
4. Un aparato configurado para procesar datos de vídeo, comprendiendo el aparato:

35 un decodificador de vídeo configurado para:

40 decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada, siendo cada componente de capa decodificada de un tipo, en el que los tipos de la pluralidad de componentes de capa decodificada incluyen al menos uno entre una vista de textura de datos de vídeo de multi-vista, una vista en profundidad de datos de vídeo de multi-vista, una capa base de datos de vídeo ajustables a escala, y cada una de una o más capas de mejora de datos de vídeo ajustables a escala; caracterizado por almacenar la pluralidad de componentes de capa decodificada en sub-unidades de una memoria intermedia de imágenes decodificadas (DPB) de tal forma que un tipo diferente de componente de capa decodificada se almacene en cada sub-unidad respectiva; y

45 realizar un proceso de gestión de la DPB en las sub-unidades, en el que el proceso de gestión de la DPB se gestiona independientemente para cada una de las sub-unidades.
5. El aparato de la reivindicación 4, en el que el proceso de gestión de la DPB comprende uno o más entre eliminar un componente de capa decodificada, entre la pluralidad de componentes de capa decodificada, de la sub-unidad respectiva, o marcar un componente de capa decodificada, entre la pluralidad de componentes de capa decodificada, en la sub-unidad respectiva como no usado para referencia.
6. El aparato de la reivindicación 5, en el que el codificador de vídeo está configurado adicionalmente para:

55 realizar el proceso de gestión de la DPB en un primer componente de capa decodificada en una unidad de acceso; y

60 realizar el proceso de gestión de la DPB en otros componentes de capa decodificada en la unidad de acceso.
7. Un aparato configurado para procesar datos de vídeo, comprendiendo el aparato:

65 medios para decodificar datos de vídeo para producir una pluralidad de componentes de capa decodificada, siendo cada componente de capa decodificada de un tipo, en el que los tipos de la pluralidad de componentes de capa decodificada incluyen al menos uno entre una vista de textura de

datos de vídeo de multi-vista, una vista en profundidad de datos de vídeo de multi-vista, una capa base de datos de vídeo ajustables a escala, o cada una de una o más capas de mejora de datos de vídeo ajustables a escala; caracterizado por

5 medios para almacenar la pluralidad de componentes de capa decodificada en sub-unidades de una memoria intermedia de imágenes decodificadas (DPB), de tal forma que un tipo diferente de componente de capa decodificada se almacene en cada sub-unidad respectiva; y

10 medios para realizar un proceso de gestión de la DPB en las sub-unidades, en el que el proceso de gestión de la DPB se gestiona independientemente para cada una de las sub-unidades.

8. El aparato de la reivindicación 7, en el que el proceso de gestión de la DPB comprende uno o más entre eliminar un componente de capa decodificada de la pluralidad de componentes de capa decodificada de la sub-unidad respectiva y marcar un componente de capa decodificada de la pluralidad de componentes de capa decodificada en la sub-unidad respectiva como no usado para referencia.

15 9. El aparato de la reivindicación 8, en el que los medios para realizar el proceso de gestión de la DPB en las sub-unidades comprenden:

20 medios para realizar el proceso de gestión de la DPB en un primer componente de capa decodificada en una unidad de acceso; y

medios para realizar el proceso de gestión de la DPB en otros componentes de capa decodificada en la unidad de acceso.

25 10. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, al ejecutarse, hacen que uno o más procesadores de un dispositivo configurado para procesar datos de vídeo:

30 decodifiquen datos de vídeo para producir una pluralidad de componentes de capa decodificada, siendo cada componente de capa decodificada de un tipo, en el que los tipos de la pluralidad de componentes de capa decodificada incluyen al menos uno entre una vista de textura de datos de vídeo de multi-vista, una vista en profundidad de datos de vídeo de multi-vista, una capa base de datos de vídeo ajustables a escala, o cada una de una o más capas de mejora de datos de vídeo ajustables a escala; caracterizado por

35 almacenar la pluralidad de componentes de capa decodificada en sub-unidades de una memoria intermedia de imágenes decodificadas (DPB), de tal forma que un tipo diferente de componente de capa decodificada se almacene en cada sub-unidad respectiva; y

40 realizar un proceso de gestión de la DPB en las sub-unidades, en el que el proceso de gestión de la DPB se gestiona independientemente para cada una de las sub-unidades.

45 11. El medio de almacenamiento legible por ordenador de la reivindicación 10, en el que el proceso de gestión de la DPB comprende uno o más entre eliminar un componente de capa decodificada, entre la pluralidad de componentes de capa decodificada, de la sub-unidad y marcar un componente de capa decodificada, entre la pluralidad de componentes de capa decodificada en la sub-unidad, como no usado para referencia.

50 12. El medio de almacenamiento legible por ordenador de la reivindicación 11, en el que las instrucciones hacen adicionalmente que los uno o más procesadores:

realicen el proceso de gestión de la DPB en un primer componente de capa decodificada en una unidad de acceso; y

55 realicen el proceso de gestión de la DPB en otros componentes de capa decodificada en la unidad de acceso.

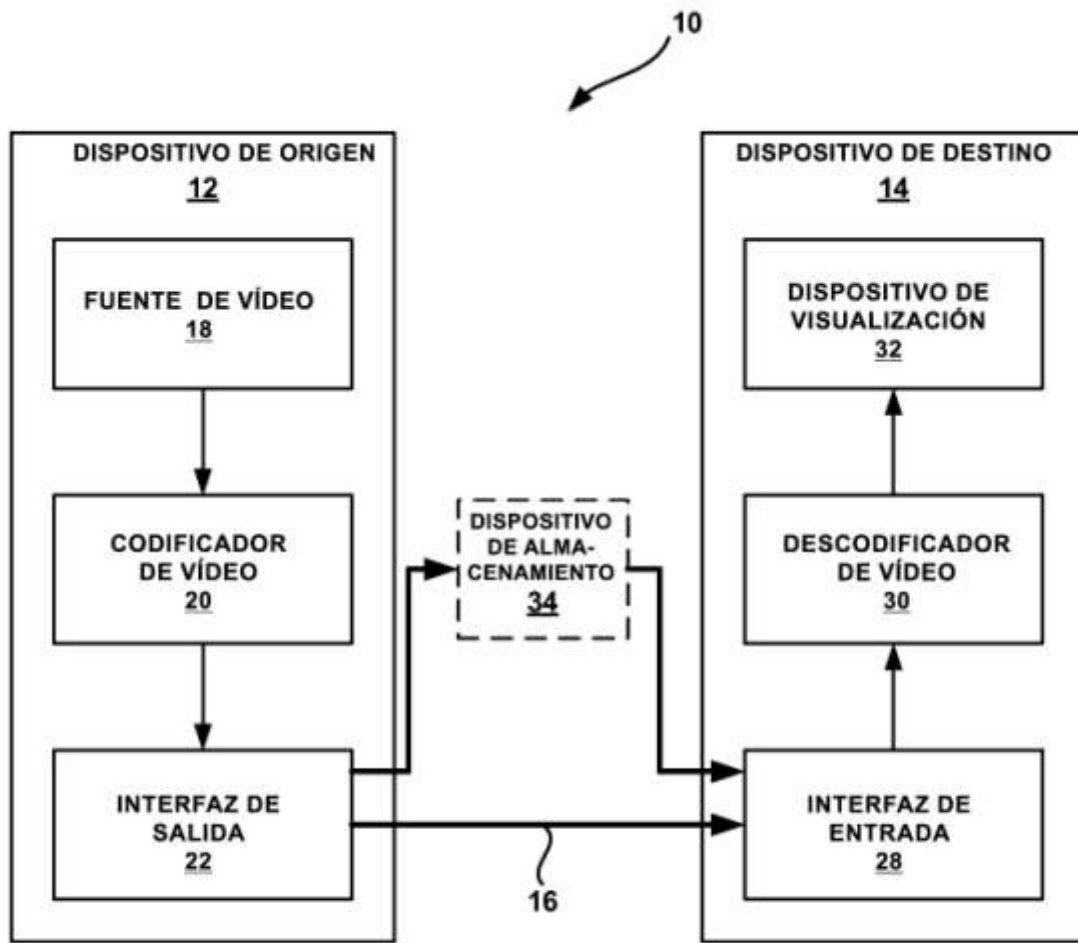


FIG. 1

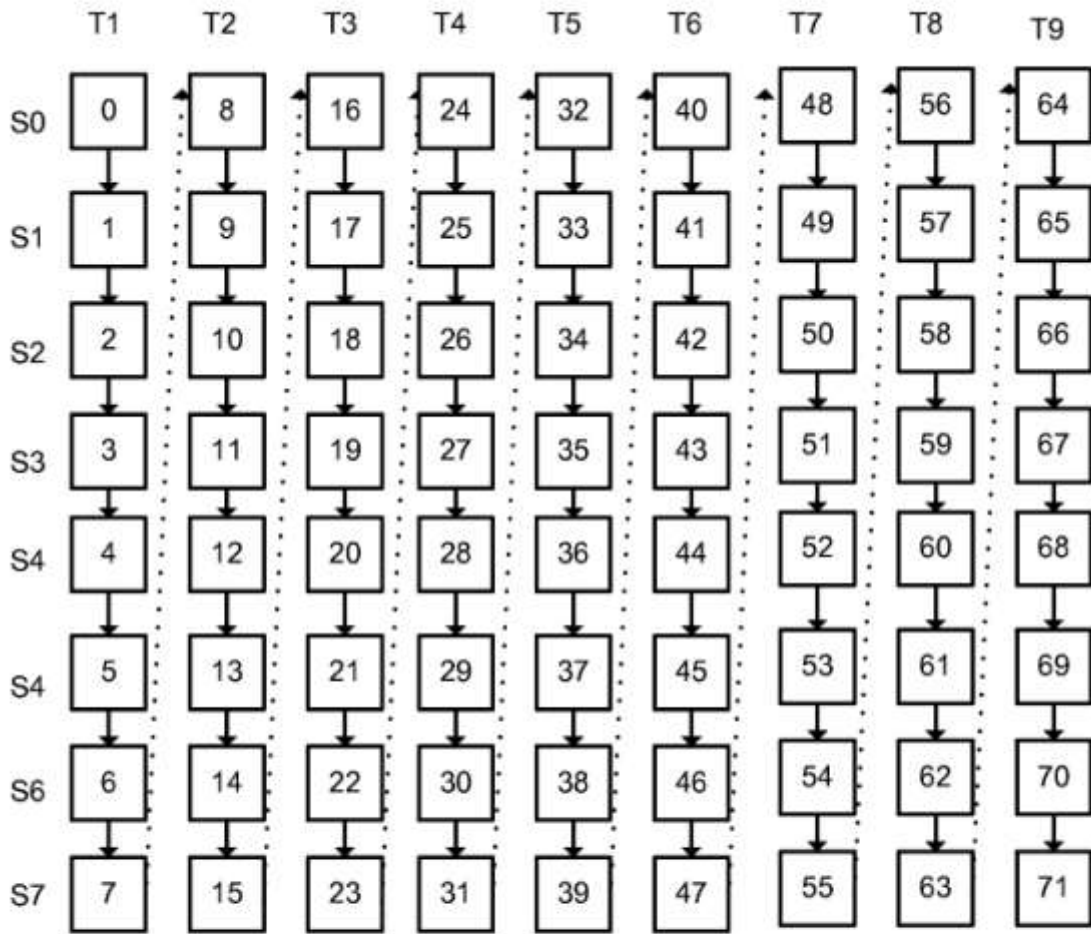


FIG. 2

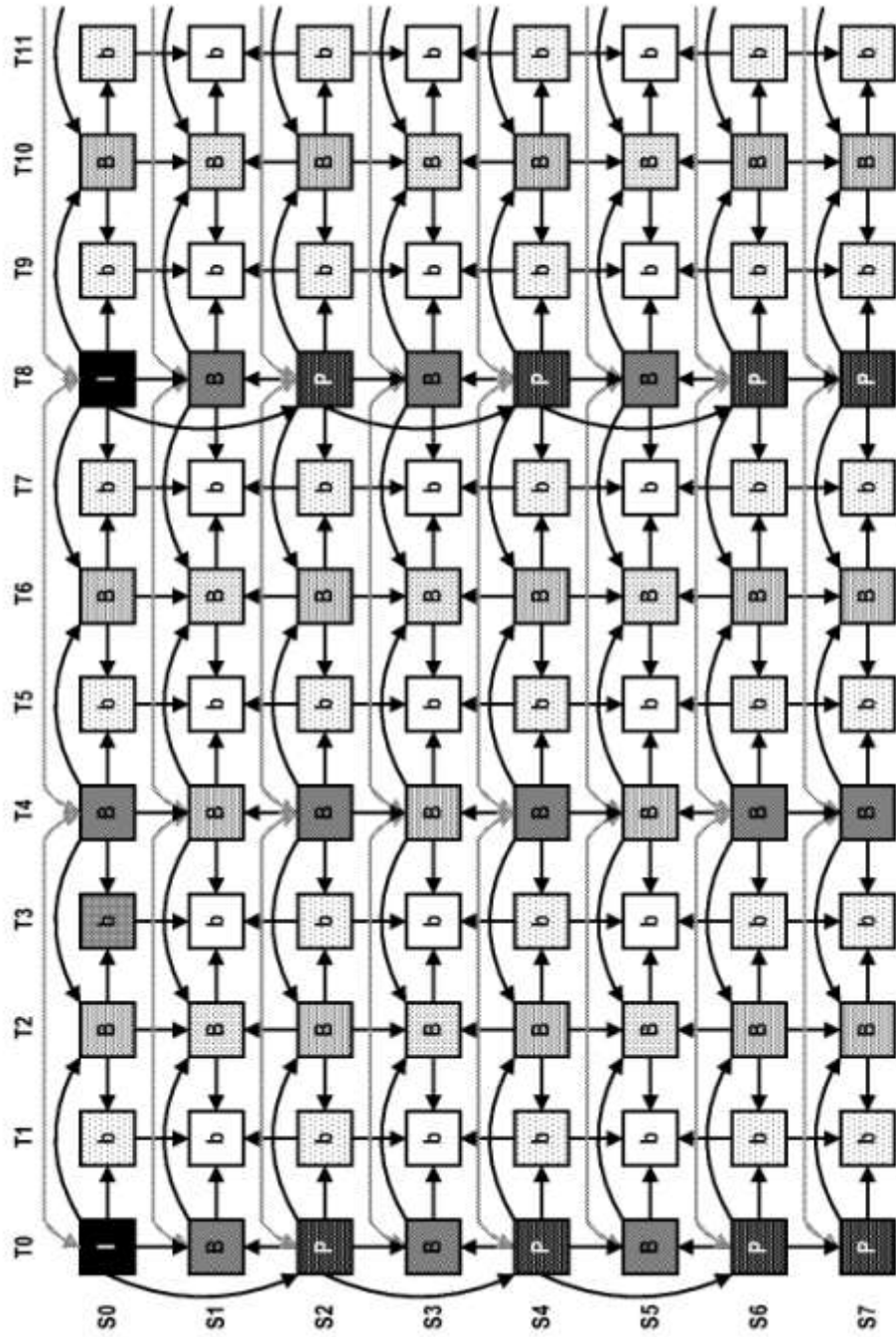


FIG. 3

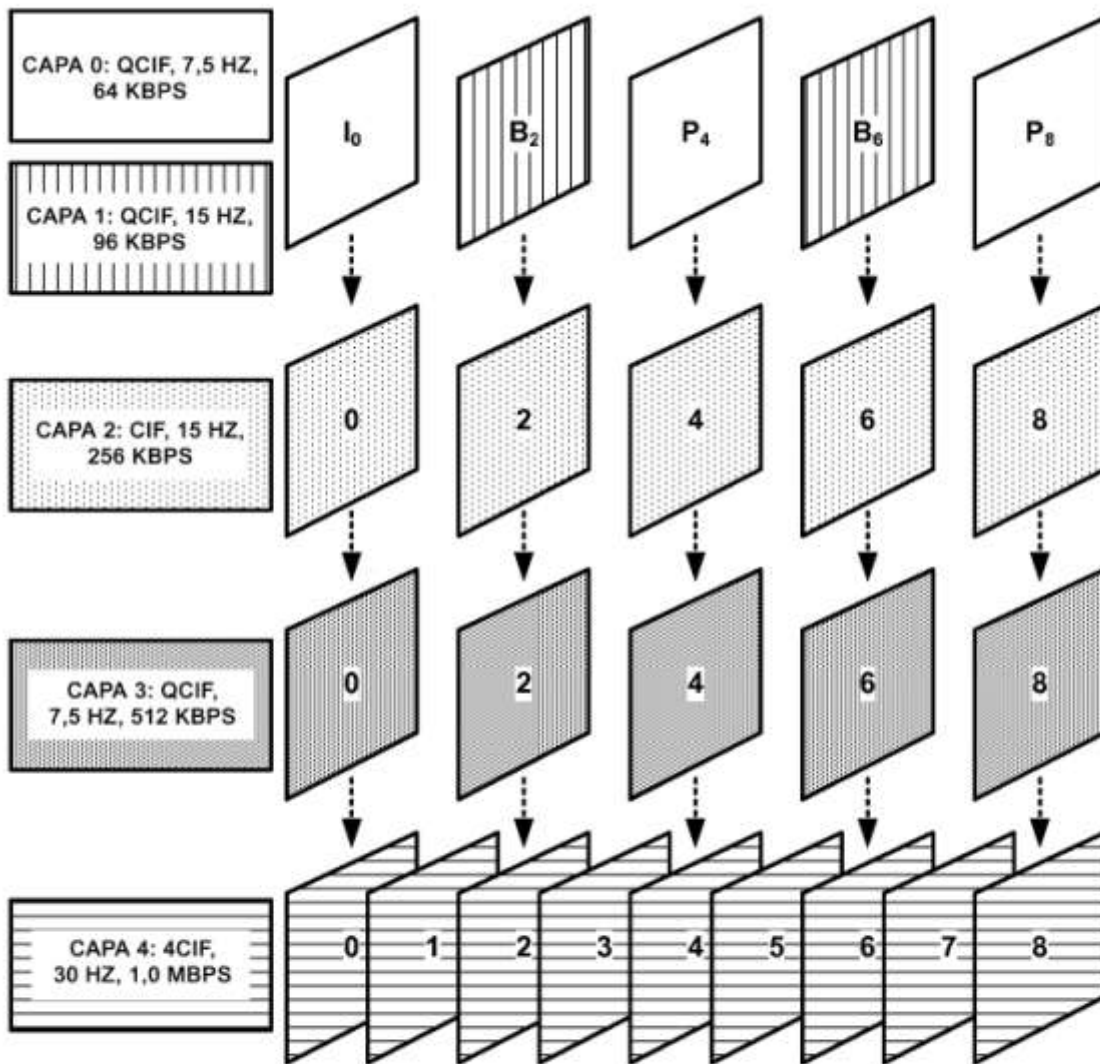


FIG. 4

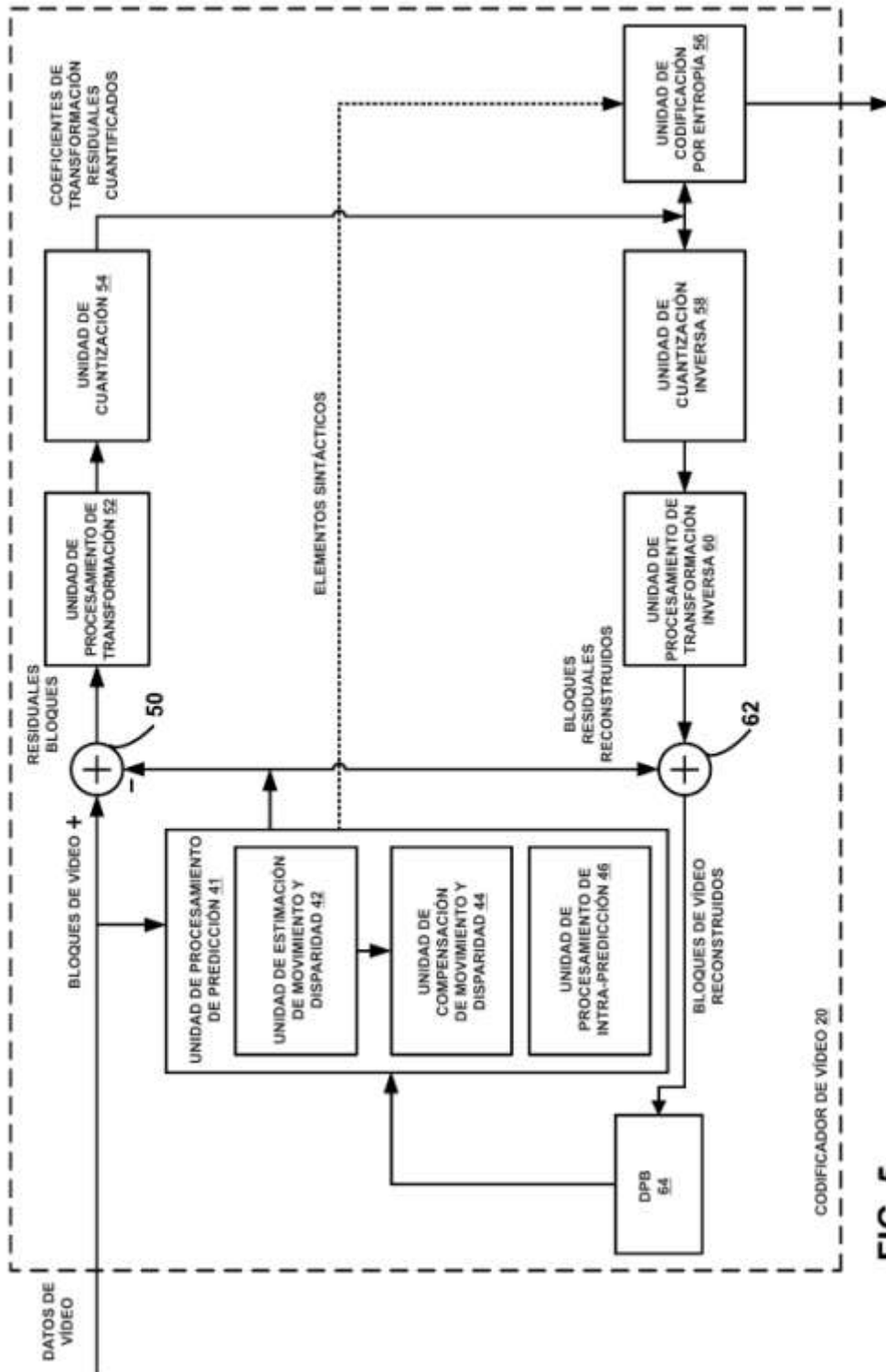


FIG. 5

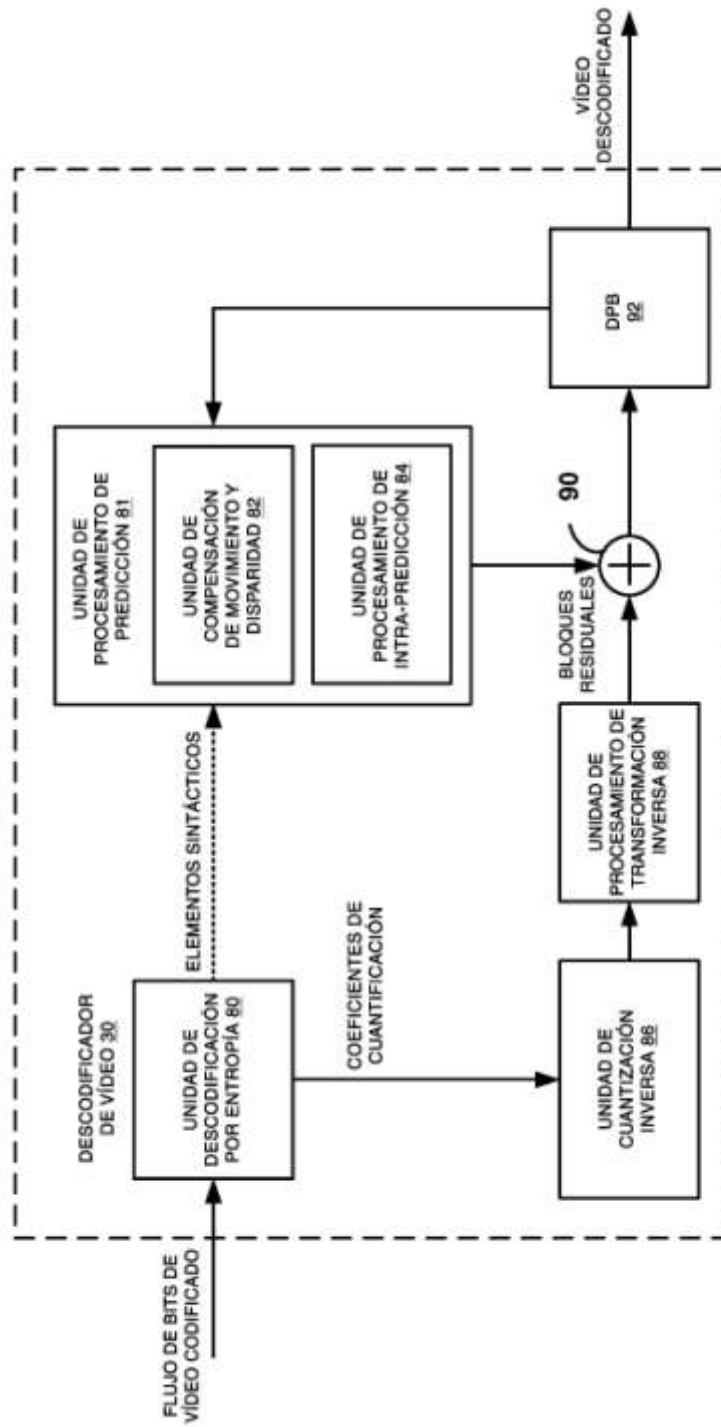
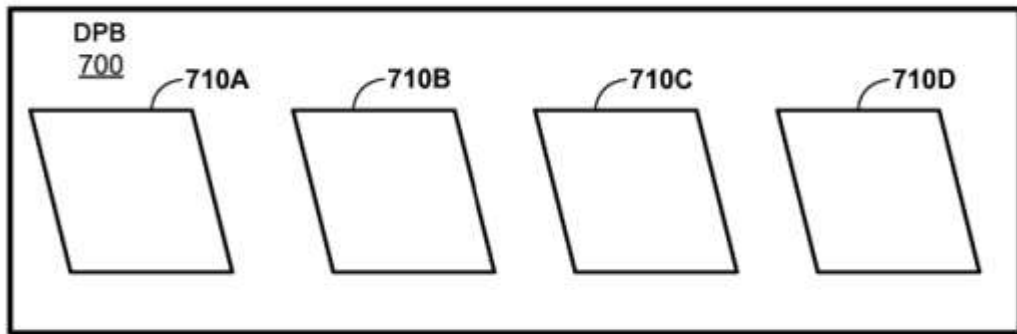


FIG. 6



CADA UNA DE LAS SUB-UNIDADES 710A A 710D INCLUYE LOS COMPONENTES DE CAPA DECODIFICADA DE TODAS LAS CAPAS COMPLETAMENTE RECONSTRUIDAS DENTRO DE UNA UNIDAD DE ACCESO

FIG. 7

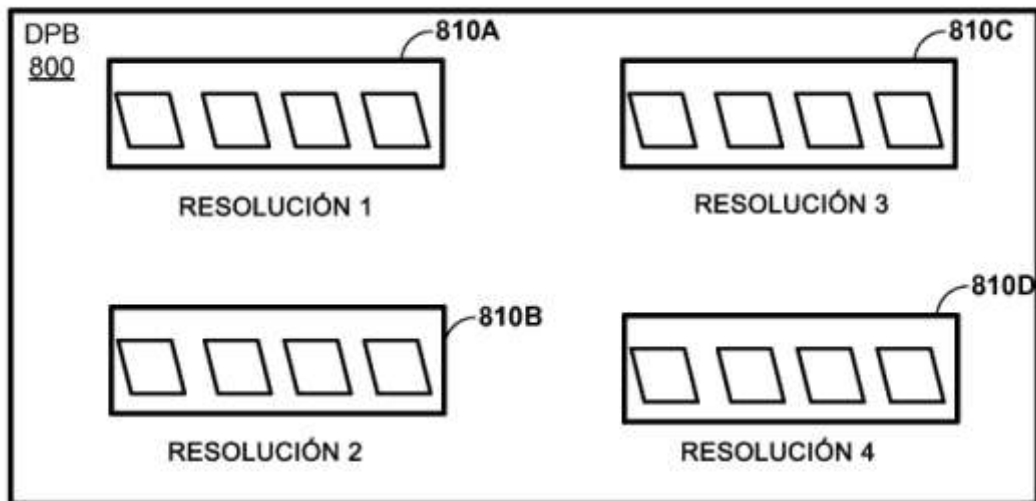


FIG. 8

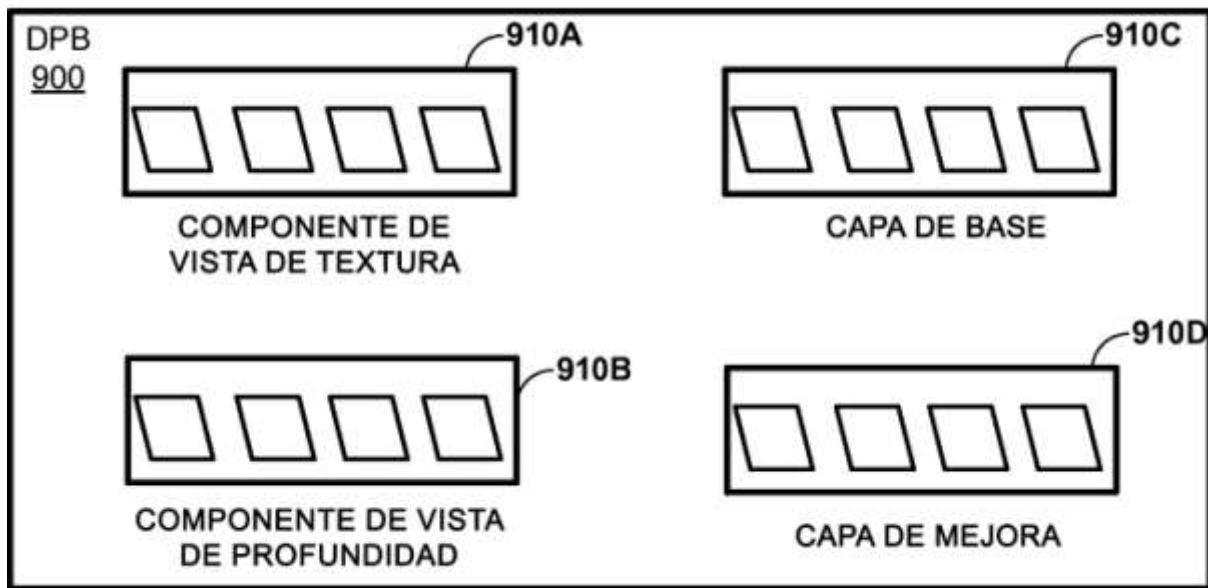
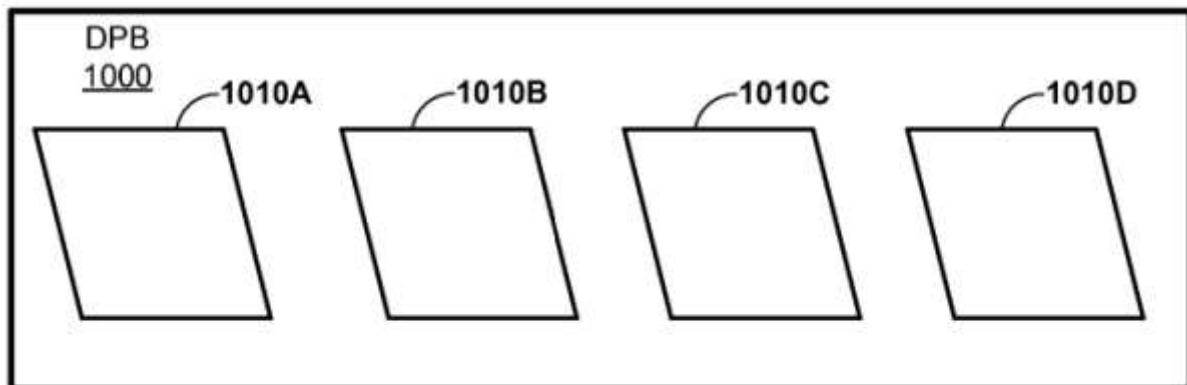


FIG. 9



CADA UNA DE LAS SUB-UNIDADES 1010A A 1010D CORRESPONDE A UN COMPONENTE DE CAPA DECODIFICADA QUE TIENE LA MÁS ALTA RESOLUCIÓN ESPACIAL

FIG. 10

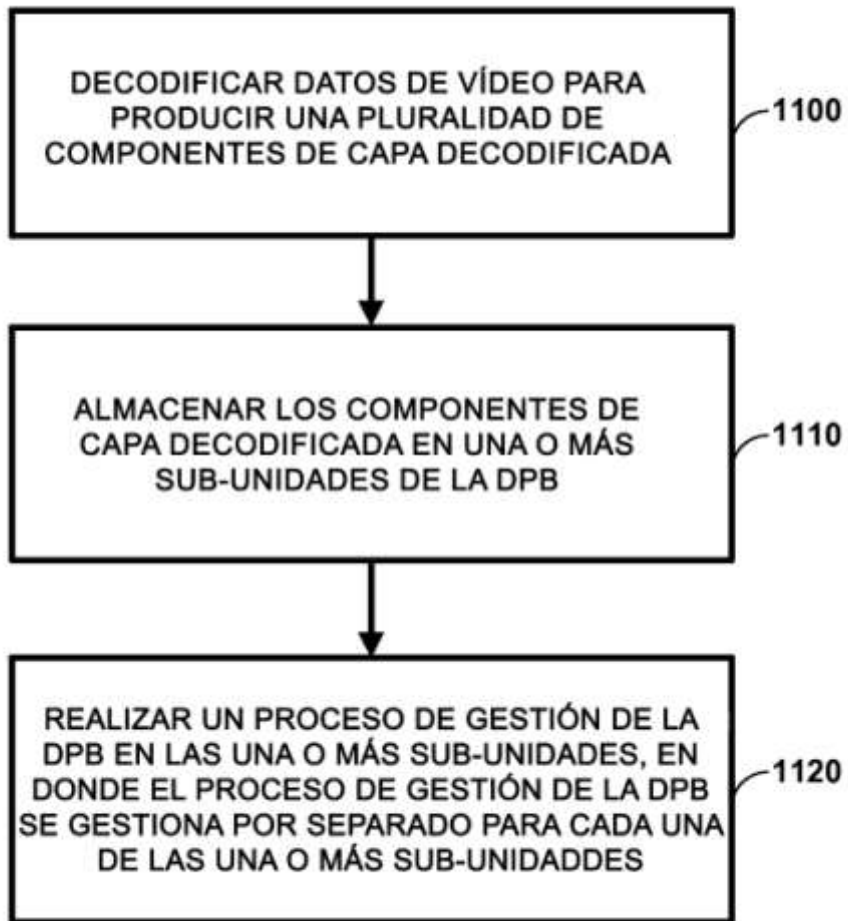


FIG. 11