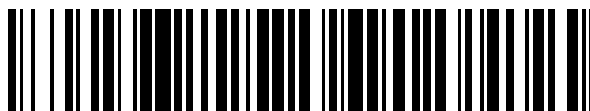


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 748 561**

51 Int. Cl.:

H04N 21/845 (2011.01)

H04N 21/438 (2011.01)

H04N 21/2343 (2011.01)

H04N 19/70 (2014.01)

H04N 19/30 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **04.04.2014** **PCT/US2014/032926**

87 Fecha y número de publicación internacional: **09.10.2014** **WO14165730**

96 Fecha de presentación y número de la solicitud europea: **04.04.2014** **E 14720460 (6)**

97 Fecha y número de publicación de la concesión europea: **03.07.2019** **EP 2982129**

54 Título: **Unidades de acceso IRAP y conmutación y empalme de flujos de bits**

30 Prioridad:

05.04.2013 US 201361809063 P

15.04.2013 US 201361812225 P

03.04.2014 US 201414244743

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
17.03.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

5775 Morehouse Drive

San Diego, CA 92121-1714, US

72 Inventor/es:

RAMASUBRAMONIAN, ADARSH KRISHNAN;

WANG, YE-KUI;

RAPAKA, KRISHNAKANTH y

CHEN, YING

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 748 561 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

Unidades de acceso IRAP y conmutación y empalme de flujos de bits

5 CAMPO TÉCNICO

[0001] Esta divulgación se refiere al procesamiento de vídeo.

ANTECEDENTES

10 [0002] Las capacidades del vídeo digital pueden incorporarse a una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en tiempo real y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital con más eficacia implementando dichas técnicas de compresión de vídeo.

25 [0003] Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques próximos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques próximos en la misma imagen o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden denominarse tramas de referencia.

35 [0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y de acuerdo con los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una formación bidimensional, pueden explorarse con el fin de generar un vector unidimensional de coeficientes de transformada, y puede aplicarse codificación por entropía para lograr aún más compresión.

45 [0005] En el documento EP2304955A1 de Thomson Licensing, un aparato codifica una señal de vídeo para proporcionar una señal SVC que comprende una señal codificada de vídeo de capa base y una señal codificada de vídeo de capa de mejora, donde la señal codificada de vídeo de capa base tiene más puntos de acceso aleatorio, por ejemplo, fragmentos IDR, que la capa de mejora, y en aquellas unidades de acceso donde la capa de mejora tiene un fragmento IDR, la capa base tiene un fragmento que no es IDR. La transmisión de la SVC se realiza en forma de paquete usando el protocolo de transporte en tiempo real (RTP), por lo que fragmentos de puntos de acceso no aleatorios se transmiten en paquetes de agregación de tiempo simples (STAP), donde cada paquete de agregación de tiempo simple comprende una unidad NAL de información de escalabilidad de contenido de carga útil (PACSI).

SUMARIO

55 [0006] Esta divulgación describe técnicas relacionadas con la codificación de soporte (por ejemplo, codificación y descodificación) de imágenes en y después de unidades de acceso de punto de acceso intraaleatorio (IRAP) parcialmente alineadas. Las técnicas de esta divulgación pueden usarse para admitir técnicas para realizar acceso aleatorio a partir de unidades de acceso IRAP parcialmente alineadas. Una unidad de acceso IRAP parcialmente alineada puede ser una unidad de acceso que incluya al menos una imagen IRAP y al menos una imagen que no sea IRAP. Un codificador de vídeo puede codificar datos que indican que las imágenes en capas que incluyen imágenes que no son IRAP en la unidad de acceso IRAP parcialmente alineada no se pueden descodificar correctamente cuando se realiza dicho acceso aleatorio. De esta manera, un descodificador de vídeo puede omitir la descodificación de imágenes que no se pueden descodificar correctamente, cuando se realiza un acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada.

[0007] En un ejemplo, se proporciona un procedimiento para descodificar datos de vídeo de codificación de vídeo de alta eficiencia, HEVC, comprendiendo los datos de vídeo una pluralidad de capas que incluyen una capa base y una o más capas de mejora, comprendiendo el procedimiento: descodificar datos que indican, cuando se realiza el acceso aleatorio desde una unidad de acceso de punto de acceso intraaleatorio, IRAP, parcialmente alineada, al menos una imagen de una capa de mejora que no se puede descodificar correctamente, en el que la unidad de acceso IRAP parcialmente alineada incluye una imagen de acceso de enlace roto, BLA, en una capa base y una imagen que no es IRAP en una capa de mejora; y descodificar la imagen BLA en la capa base de la unidad de acceso IRAP parcialmente alineada, en el que, para la al menos una imagen que no se puede descodificar correctamente, se invoca un proceso para generar imágenes de referencia no disponibles y la al menos una imagen no se proporciona, en el que la al menos una imagen de la capa de mejora que no se puede descodificar correctamente comprende la imagen que no es IRAP en la capa de mejora en la unidad de acceso IRAP parcialmente alineada.

[0008] En otro ejemplo, un dispositivo para descodificar datos de vídeo de codificación de vídeo de alta eficiencia, HEVC, comprendiendo los datos de vídeo una pluralidad de capas que incluyen una capa base y una o más capas de mejora, comprendiendo el dispositivo: medios para descodificar datos que indican, cuando se realiza el acceso aleatorio desde una unidad de acceso IRAP parcialmente alineada, al menos una imagen de una capa de mejora que no se puede descodificar correctamente, en el que la unidad de acceso IRAP parcialmente alineada incluye una imagen de acceso de enlace roto, BLA, en una capa base y una imagen que no es IRAP en una capa de mejora; y medios para descodificar la imagen BLA en la capa base de la unidad de acceso IRAP parcialmente alineada, en el que, para la al menos una imagen que no se puede descodificar correctamente, se invoca un proceso para generar imágenes de referencia no disponibles y la al menos una imagen no se proporciona, en el que la al menos una imagen de la capa de mejora que no se puede descodificar correctamente comprende la imagen que no es IRAP en la capa de mejora en la unidad de acceso IRAP parcialmente alineada.

[0009] En otro ejemplo, un medio de almacenamiento legible por ordenador tiene instrucciones almacenadas en el mismo que, al ser ejecutadas, hacen que un procesador de un dispositivo de descodificación de datos de vídeo ejecute el procedimiento descrito anteriormente.

[0010] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0011]

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

La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 3 es un diagrama de bloques que ilustra un descodificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 4 es un diagrama de bloques que ilustra un conjunto de ejemplo de dispositivos que forman parte de una red de comunicación de datos de vídeo.

La FIG. 5 muestra un ejemplo de alineación diferente de imágenes de puntos de acceso intraaleatorios (IRAP) en unidades de acceso.

La FIG. 6 muestra un ejemplo de empalme en una unidad de acceso (AU) IRAP parcialmente alineada.

La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de procedimiento para la codificación de datos de vídeo de acuerdo con las técnicas de esta divulgación.

La FIG. 8 es un diagrama de flujo que ilustra un ejemplo de procedimiento de descodificación de datos de vídeo de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0012] Esta divulgación describe técnicas para admitir las operaciones de acceso aleatorio y empalme en unidades de acceso de un flujo de bits de vídeo que tienen múltiples capas. En particular, las técnicas de esta divulgación se refieren a situaciones en las que un flujo de bits de múltiples capas incluye unidades de acceso de punto de acceso intraaleatorio (IRAP) no alineadas. Una unidad de acceso IRAP no alineada, también denominada en el presente documento unidad de acceso IRAP parcialmente alineada, incluye al menos una imagen IRAP y al menos una imagen

no IRAP. La imagen IRAP puede comprender, por ejemplo, una imagen de actualización instantánea de descodificador (IDR), una imagen de acceso aleatorio limpio (CRA) o una imagen de acceso de enlace roto (BLA). Las técnicas de esta divulgación proporcionan soporte para casos en los que no todas las imágenes en una unidad de acceso IRAP son imágenes IRAP. Una unidad de acceso incluye todas las imágenes codificadas pertenecientes a un tiempo de salida particular, junto con sus unidades NAL no VCL asociadas. Una capa puede ser una capa escalable, una vista, una componente de profundidad de una vista, una componente de textura de una vista, etc.

[0013] En general, las técnicas de esta divulgación admiten situaciones en las que el acceso aleatorio se realiza a partir de una unidad de acceso IRAP parcialmente alineada. Por ejemplo, la unidad de acceso IRAP parcialmente alineada puede incluir una imagen IRAP en una capa base y una imagen no IRAP en una capa de mejora. Debido a que el acceso aleatorio se realiza a partir de la unidad de acceso IRAP parcialmente alineada, suponiendo que la imagen no IRAP de la capa de mejora se predice al menos parcialmente usando interpredicción temporal, la imagen no IRAP de la capa de mejora no será descodificable. Es decir, no se habrá recuperado una imagen anterior de la que depende la imagen que no es IRAP. Además, una o más imágenes subsiguientes a la imagen que no es IRAP en la capa de mejora también pueden no ser descodificables. Las imágenes que no se pueden descodificar correctamente pueden denominarse imágenes descartables de capa de mejora (ELDISC). Debe entenderse que la expresión "no se puede descodificar correctamente" se refiere a imágenes que, cuando se trata de un descodificador de vídeo, no son las mismas que las reproducidas por un codificador de vídeo. Esto puede ocurrir cuando, por ejemplo, no se reciben imágenes de referencia para estas imágenes, por ejemplo, cuando las imágenes de referencia preceden a una unidad de acceso IRAP correspondiente en orden de descodificación, y cuando la unidad de acceso IRAP se usa para el acceso aleatorio.

[0014] De acuerdo con las técnicas de esta divulgación, un codificador de vídeo puede señalar que ciertas imágenes no se pueden descodificar correctamente cuando se realiza un acceso aleatorio desde una unidad de acceso IRAP parcialmente alineada. Dichas imágenes no descodificables pueden incluir aquellas que están en una capa de mejora para la cual la unidad de acceso IRAP no incluye una imagen IRAP y que están en la misma secuencia de vídeo codificada (CVS) que la unidad de acceso IRAP. De esta manera, un descodificador de vídeo puede determinar qué imágenes no se pueden descodificar correctamente cuando se realiza un acceso aleatorio desde una unidad de acceso IRAP parcialmente alineada, y omitir la descodificación de las imágenes no descodificables. En cambio, el descodificador de vídeo puede simplemente analizar sintácticamente los datos de las imágenes no descodificables, sin intentar descodificar estas imágenes. De esta manera, el descodificador de vídeo puede determinar de forma simple, rápida y eficiente que dichas imágenes no se pueden descodificar correctamente a partir de los datos señalizados, de modo que el descodificador de vídeo simplemente puede determinar el llevar a cabo un procedimiento para imágenes no existentes para estas imágenes u omitir la descodificación de estas imágenes.

[0015] Los datos que se señalizan para las imágenes no descodificables pueden incluir, por ejemplo, un tipo de unidad de capa de abstracción de red (NAL) particular para las unidades NAL de las imágenes. Adicionalmente o de forma alternativa, los datos pueden incluir una bandera u otro elemento sintáctico que indique si una imagen se puede descodificar cuando se accede aleatoriamente a una secuencia de vídeo codificada que incluye la imagen, comenzando con una unidad de acceso parcialmente alineada. La bandera o el elemento sintáctico pueden ser, por ejemplo, datos de cabecera de fragmento, datos incluidos en una cabecera de unidad NAL, datos incluidos en un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de secuencia (SPS) u otros datos similares.

[0016] Debe entenderse que los datos que indican que una imagen particular no se puede descodificar correctamente están separados de los propios datos de imagen codificados, es decir, los datos de los bloques codificados de la imagen. Es decir, los datos que indican que una imagen en particular no se puede descodificar correctamente se pueden analizar sintácticamente e interpretar sin intentar realmente descodificar la imagen. De esta manera, los datos que indican que una imagen no se puede descodificar correctamente se pueden interpretar antes de intentar descodificar la imagen, de modo que un descodificador puede evitar intentar descodificar una imagen que no es descodificable y que se señala que es no descodificable. En cambio, el descodificador de vídeo puede simplemente analizar sintácticamente los datos de la capa de mejora no descodificable sin intentar descodificar los datos de la imagen. De forma alternativa, el descodificador de vídeo puede realizar un proceso de imagen no existente para descodificar estas imágenes.

[0017] Permitir unidades de acceso IRAP parcialmente alineadas puede proporcionar varios beneficios. Por ejemplo, una capa base puede incluir imágenes IRAP más frecuentes que una capa de mejora. Esto puede reducir el retardo de ajuste de un dispositivo cliente, ya que el dispositivo cliente puede comenzar a descodificar y mostrar datos de vídeo de la capa base, incluso si los datos de la capa de mejora aún no son descodificables. Además, esto permite una selección de granularidad fina de puntos de acceso aleatorio. Además, debido a que la capa de mejora no necesita incluir imágenes IRAP tan frecuentemente como la capa base, se puede reducir la velocidad de bits para la capa de mejora.

[0018] Las normas de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo de múltiples vistas (MVC).

[0019] Además, existe una nueva norma de codificación de vídeo, concretamente la codificación de vídeo de alta eficiencia (HEVC), desarrollada por el Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) del Grupo de Expertos sobre Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos sobre Imágenes en Movimiento (MPEG) de ISO/IEC. Un borrador de trabajo (WD) reciente de HEVC y denominado HEVC WD10 en adelante en el presente documento, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip.

[0020] El JCT-3V también está desarrollando la ampliación multivista para HEVC, en concreto, la MV-HEVC. Un borrador de trabajo (WD) reciente de MV-HEVC (documento: JCT3V-C1004_d3) denominado MV-HEVC WD3 en adelante en el presente documento, está disponible en http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1004-v4.zip.

[0021] El JCT-VC también está desarrollando la ampliación escalable de HEVC, denominada SHVC. Un borrador de trabajo (WD) reciente de SHVC (documento: OJCTVC-L1008) y denominado SHVC WD1 en adelante en el presente documento, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1008-v1.zip.

[0022] A continuación se describen las imágenes IRAP y las unidades de acceso en HEVC. En HEVC con codificación de una sola capa, se aplican los siguientes conceptos de imagen IRAP y AU IRAP.

[0023] Las imágenes IRAP son imágenes que solo contienen fragmentos I, y tienen la propiedad de que un proceso de descodificación que comienza en la imagen IRAP en orden de descodificación puede descodificar correctamente todas las imágenes (no RASL) iniciales omitidas de acceso no aleatorio que suceden a la imagen IRAP en orden de descodificación, sin realizar la descodificación de ninguna imagen que preceda al IRAP en orden de descodificación. Una imagen IRAP puede ser una imagen de acceso de enlace roto (BLA), una imagen de acceso aleatorio limpio (CRA) o una imagen de actualización instantánea de descodificación (IDR). Puede haber imágenes RASL asociadas a una imagen IRAP que no se pueden descodificar correctamente al realizar un acceso aleatorio a partir de la imagen IRAP.

[0024] Una unidad de acceso que contiene una imagen IRAP es una unidad de acceso IRAP que se describe a continuación.

[0025] A continuación se describe el empalme y la conmutación de flujo de bits.

[0026] Las operaciones de conmutación y empalme de flujo de bits consisten en el proceso de unir dos flujos de bits para obtener un flujo de bits, con cambios mínimos en el flujo de bits en el punto de empalme y una transición casi perfecta. Por lo general, los dos flujos de bits originales comienzan en un punto de acceso aleatorio, típicamente, una unidad de acceso IRAP.

[0027] Un caso de uso típico del empalme es cuando se insertan anuncios de televisión entre los datos de medios. Otro caso de uso típico de empalme es la conmutación de flujo de bits en la transmisión de vídeo adaptable.

[0028] Las imágenes IRAP en MV-HEVC y SHVC se describen a continuación.

[0029] En MV-HEVC o SHVC, se puede usar el término componente de capa para especificar una colección de las unidades NAL VCL codificadas de la unidad de acceso con un id de capa nuh único, igual que el término de imagen en MV-HEVC WD3 y SHVC WD1. El término componente de capa también se puede usar para especificar las unidades NAL VCL descodificadas, que es una representación descodificada del vídeo en una determinada instancia de tiempo de una capa única.

[0030] En MV-HEVC WD3 y SHVC WD1, los componentes de capa IDR y BLA (imágenes) están alineados a través de las capas, mientras que las imágenes CRA no lo están. Por ejemplo, cuando se requiere que los componentes de capa IDR estén alineados, si alguna imagen en una AU es una imagen IDR, entonces todas las imágenes en la AU son componentes de capa IDR, es decir, con un tipo de unidad NAL de IDR_W_RADL o IDR_N_LP. De manera similar, cuando se requiere que las imágenes BLA estén alineadas, si algún componente de capa en una unidad de acceso es un componente de capa BLA, entonces todos los componentes de capa en la AU son imágenes BLA.

[0031] En esta divulgación, se supone que una AU consiste en todas las imágenes codificadas que pertenecen a un tiempo de salida particular, junto con sus unidades NAL asociadas que no son VCL.

[0032] En las ampliaciones de HEVC, un flujo de bits puede tener una o más capas. De acuerdo con el último borrador de trabajo, los componentes de capa IDR y BLA (imágenes) tienen que alinearse a través de las diferentes capas en una unidad de acceso, mientras que no existe tal restricción en las imágenes CRA.

[0033] Actualmente no hay ningún mecanismo que admita el acceso aleatorio simple o las operaciones de empalme en ciertas AU que no tengan todas las imágenes como imágenes IRAP, por lo que es necesario, a lo sumo, cambiar los valores de tipo de unidad NAL en las cabeceras de unidad NAL de algunos componentes de capa. Cuando el acceso aleatorio o la conmutación de flujo de bits se realiza en una unidad de acceso donde solo la capa base es una imagen CRA, no es posible descodificar correctamente y proporcionar imágenes de capa superior hasta que aparezca otro componente de capa IRAP en esa capa. Para admitir el acceso aleatorio simple anterior u operaciones de empalme del flujo de bits, se necesita un manejo adecuado de dichas imágenes de capa superior.

[0034] Esta divulgación describe varios ejemplos que proporcionan procedimientos para proporcionar medios para realizar el acceso aleatorio y el empalme de flujo de bits en AU en las que las imágenes IRAP no están alineadas (es decir, no todas las imágenes en la AU son imágenes IRAP). Algunos de los ejemplos son los siguientes:

1. Definir AU IRAP para incluir AU con alineación parcial de imágenes IRAP (es decir, las AU en las que algunos componentes de capa son imágenes IRAP, mientras que otros componentes de capa no son imágenes IRAP). Dichas AU IRAP se denominan AU IRAP parcialmente alineadas.
2. Permitir que un flujo de bits válido comience con una AU IRAP parcialmente alineada.
3. Proporcionar una indicación de los componentes de capa que no se pueden descodificar cuando se realiza el empalme en una AU IRAP parcialmente alineada.
4. Indicar para cada componente de capa no base (es decir, un componente de capa que no pertenece a una capa con id de capa nuh igual a 0) si se puede descodificar cuando se realiza el empalme a partir de una AU IRAP.
5. Especificar un proceso de descodificación de imágenes no descodificables para requerir que su sintaxis y otras características cumplan con los requisitos de conformidad de flujo de bits.

[0035] Un rasgo característico de esta divulgación implica la idea de incluir algunas unidades de acceso, donde no todas las imágenes son IRAP, como una unidad de acceso IRAP. Esto se logra definiendo unidades de acceso IRAP parcialmente alineadas. Una ilustración de la alineación de imágenes IRAP en diferentes unidades de acceso se ofrece en la FIG. 5, y las descripciones detalladas se ofrecen en las siguientes secciones.

[0036] La FIG. 5 muestra un ejemplo de diferente alineación de imágenes IRAP en unidades de acceso. La alineación de izquierda a derecha de las AU no indica su orden de descodificación. La FIG. 6 muestra un ejemplo de empalme en una AU IRAP parcialmente alineada. Las imágenes ELDISC son imágenes que no se pueden descodificar debido a la operación de empalme.

[0037] Un primer ejemplo se describe a continuación. En este ejemplo, a las imágenes de un flujo de bits que no se pueden descodificar correctamente cuando la descodificación comienza desde la unidad de acceso IRAP anterior (parcialmente alineada) en orden de descodificación, se les asigna un nuevo tipo de unidad NAL. Cuando la descodificación se inicia desde una AU IRAP parcialmente alineada, estas imágenes no se proporcionan y se invoca el proceso para generar imágenes no disponibles. Además, se supone que las imágenes IDR en una AU están alineadas en varias capas, mientras que las imágenes BLA pueden no estar alineadas en varias capas.

[0038] A modo de ejemplo, esta divulgación supone las siguientes definiciones:

Unidad de acceso (AU) IRAP: Una unidad de acceso que contiene una o más imágenes IRAP donde para cada imagen IRAP con ID de capa nuh_layer_id, todas las imágenes de la AU con ID de capa menor que nuh_layer_id también son IRAP.

AU IRAP alineada: Una AU IRAP en la que todas las imágenes de la AU son imágenes IRAP.

AU IRAP parcialmente alineada: Una AU IRAP que no es una AU IRAP alineada.

Imagen ELDISC: Una imagen final que tiene un ID de capa layerId distinto de cero y que pertenece a una AU IRAP parcialmente alineada que tiene un ID de capa de acceso menor que layerId o que pertenece a una AU posterior, en orden de descodificación, que precede, en orden de descodificación, a cualquier AU IRAP parcialmente alineada con ID de capa de acceso mayor que o igual a layerId.

ID de capa de acceso: El ID de capa de acceso de una AU IRAP es el mayor valor de nuh_layer_id de todas las imágenes IRAP en la AU IRAP.

[0039] A continuación se describe el indicar la no descodificabilidad de imágenes EL.

Semántica de cabecera de unidad NAL

[0040] Se presentan dos tipos de unidades NAL, ELDISC_R y ELDISC_N, que se muestran en la tabla que se muestra a continuación. Los cambios en la semántica de los tipos de unidades NAL también se describen a continuación. Los cambios se muestran usando texto *en cursiva* para representar las adiciones y [eliminado: ""] para representar eliminaciones.

forbidden_zero_bit será igual a 0.

nal_unit_type especifica el tipo de estructura de datos RBSP contenida en la unidad NAL como se especifica en la Tabla 7-1.

[0041] Las unidades NAL que tienen nal_unit_type en el intervalo de UNSPEC48..UNSPEC63, inclusive, para las cuales no se especifica la semántica, no afectarán al proceso de descodificación especificado en esta memoria descriptiva.

NOTA 1 - Los tipos de unidades NAL en el intervalo de UNSPEC48..UNSPEC63 se pueden usar según lo determine la aplicación. En esta memoria descriptiva no se especifica ningún proceso de descodificación para estos valores de nal_unit_type. Dado que diferentes aplicaciones podrían usar estos tipos de unidades NAL para diferentes propósitos, se debe tener especial cuidado en el diseño de los codificadores que generan unidades NAL con estos valores nal_unit_type, y en el diseño de descodificadores que interpretan el contenido de las unidades NAL con estos valores nal_unit_type.

[0042] Para propósitos distintos a la determinación de la cantidad de datos en las unidades de descodificación del flujo de bits (como se especifica en el Anexo C de WD 10), los descodificadores ignorarán (eliminarán del flujo de bits y descartarán) el contenido de todas las unidades NAL que usan valores reservados de nal_unit_type.

NOTA 2 - Este requisito permite la definición futura de ampliaciones compatibles con esta memoria descriptiva.

Tabla 7-1 - Códigos de tipo de unidad NAL y clases de tipo de unidad NAL

nal_unit_type	Nombre nal_unit_type	de Contenido de la unidad NAL y estructura sintáctica RBSP	Clase de tipo de unidad NAL
0	TRAIL_N	Segmento codificado de una	VCL
1	TRAIL_R	slice_segment_layer_rbsp() de imagen final no TSA ni STSA	
2	TSA_N	Segmento codificado de una	VCL
3	TSA_R	slice_segment_layer_rbsp() de imagen TSA	
4	STSA_N	Segmento codificado de una slice_segment_	VCL
5	STSA_R	layer_rbsp() de imagen STSA	
6	RADL_N	Segmento de sector codificado de una	VCL
7	RADL_R	slice_segment_layer_rbsp() de imagen RADL	
8	RASL_N	Segmento codificado de una slice_segment_	VCL
9	RASL_R	layer_rbsp() de imagen RASL	
10	RSV_VCL_N10	Tipos reservados de unidades NAL VCL no de	VCL
12	RS V_VCL_N12	referencia de subcapa no IRAP	
14	RSV_VCL_N14		
11	RSV_VCL_R11	Tipos reservados de unidades NAL VCL de	VCL
13	RSV_VCL_R13	referencia de subcapa no IRAP	
15	RSV_VCL_R15		
16	BLA_W_LP	Segmento codificado de una	VCL
17	BLA_W_RADL	slice_segment_layer_rbsp() de imagen BLA	
18	BLA_N_LP		
19	IDR_W_RADL	Segmento codificado de una slice_segment_	VCL
20	IDR_N_LP	layer_rbsp() de imagen IDR	
21	CRA_NUT	Segmento codificado de una slice_segment_	VCL
		layer_rbsp() de imagen CRA	
22	RSV_IRAP_VCL22	Tipos reservados de unidades NAL VCL IRAP	VCL
23	RSV_IRAP_VCL23		
24	ELDISC_N	Segmento codificado de una imagen ELDISC	VCL

nal_unit_type	Nombre de nal_unit_type	Contenido de la unidad NAL y estructura sintáctica RBSP	Clase de tipo de unidad NAL
25	ELDISC_R		
[eliminado: "24"] 26..31	RSV_VCL24.. RSV_VCL31	Tipos reservados de unidades NAL VCL no IRAP	VCL
32	VPS_NUT	Conjunto de parámetros de vídeo video_parameter_set_rbsp()	no VCL
33	SPS_NUT	Conjunto de parámetros de secuencia seq_parameter_set_rbsp()	no VCL
34	PPS_NUT	Conjunto de parámetros de imagen pic_parameter_set_rbsp()	no VCL
35	AUD_NUT	Delimitador de unidad de acceso access_unit_delimiter_rbsp()	no VCL
36	EOS_NUT	Fin de secuencia end_of_seq_rbsp()	no VCL
37	EOB_NUT	Fin de flujo de bits end_of_bitstream_rbsp()	no VCL
38	FD_NUT	Datos de relleno filler_data_rbsp()	no VCL
39 40	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Información complementaria de mejora sei_rbsp()	no VCL
41..47	RSV_NVCL41.. RSV_NVCL47	Reservado	no VCL
48..63	UNSPEC48.. UNSPEC63	Sin especificar	no VCL

NOTA 3 - Una imagen CRA puede tener imágenes RASL o RADL asociadas presentes en el flujo de bits.

5 NOTA 4 - Una imagen BLA que tenga nal_unit_type igual a BLA_W_LP puede tener imágenes RASL o RADL asociadas presentes en el flujo de bits. Una imagen BLA que tiene nal_unit_type igual a BLA_W_DLP no tiene imágenes RASL asociadas presentes en el flujo de bits, pero puede tener imágenes RADL asociadas en el flujo de bits. Una imagen BLA que tiene nal_unit_type igual a BLA_N_LP no tiene imágenes iniciales asociadas presentes en el flujo de bits.

10 NOTA 5 - Una imagen IDR que tiene nal_unit_type igual a IDR_N_LP no tiene imágenes iniciales asociadas presentes en el flujo de bits. Una imagen IDR que tiene nal_unit_type igual a IDR_W_RADL no tiene imágenes RASL asociadas presentes en el flujo de bits, pero puede tener imágenes RADL asociadas en el flujo de bits.

15 NOTA 6 - No se incluye una imagen de subcapa de no referencia en RefPicSetStCurrBefore, RefPicSetStCurrAfter ni RefPicSetLtCurr de ninguna imagen con el mismo valor de TemporalId, y se puede descartar sin afectar a la descodificación de otras imágenes con el mismo valor de TemporalId.

20 **[0043]** Todas las unidades NAL de segmento codificado de una unidad de acceso tendrán el mismo valor de nal_unit_type. También se hace referencia a una imagen o una unidad de acceso que tiene un nal_unit_type igual al nal_unit_type de las unidades NAL de segmento codificado de la imagen o unidad de acceso.

25 **[0044]** Si una imagen tiene nal_unit_type igual a ELDISC_N, TRAIL_N, TSA_N, STSA_N, RADL N, RASL N, RSV_VCL_N10, RSV_VCL_N12 o RSV_VCL_N14, la imagen es una imagen de subcapa de no referencia. De lo contrario, la imagen es una imagen de referencia de subcapa.

[0045] Es un requisito de conformidad con el flujo de bits que una imagen ELDISC que pertenece a una AU IRAP sea una imagen de referencia de subcapa.

30 **[0046]** Cada imagen, aparte de la primera imagen en el flujo de bits en orden de descodificación, se considera que está asociada a la imagen IRAP anterior en orden de descodificación.

[0047] Cuando una imagen es una imagen inicial, debe ser una imagen RADL o RASL.

35 **[0048]** Cuando una imagen es una imagen final, no debe ser una imagen RADL o RASL.

[0049] Cuando una imagen es una imagen inicial, precederá, en orden de descodificación, a todas las imágenes finales que están asociadas a la misma imagen IRAP.

[0050] Ninguna imagen RASL estará presente en el flujo de bits que esté asociado a una imagen BLA que tenga `nal_unit_type` igual a `BLA_W_RADL` o `BLA_N_LP`.

[0051] Ninguna imagen RASL estará presente en el flujo de bits que esté asociado a una imagen IDR.

[0052] Ninguna imagen RADL estará presente en el flujo de bits que esté asociado a una imagen BLA que tenga `nal_unit_type` igual a `BLA_N_LP` o que esté asociado a una imagen IDR que tenga `nal_unit_type` igual a `IDR_N_LP`.

NOTA 7 - Es posible realizar un acceso aleatorio en la posición de una unidad de acceso IRAP descartando todas las unidades de acceso antes de la unidad de acceso IRAP (y descodificar correctamente la imagen IRAP y todas las imágenes posteriores no RASL en el orden de descodificación), siempre cada conjunto de parámetros esté disponible (ya sea en el flujo de bits o por medios externos no especificados en esta memoria descriptiva) cuando sea necesario activarlo.

[0053] Cualquier imagen que tenga `PicOutputFlag` igual a 1 que precede a una imagen IRAP en orden de descodificación deberá preceder a la imagen IRAP en orden de salida y deberá preceder a cualquier imagen RADL asociada a la imagen IRAP en orden de salida.

[0054] Cualquier imagen RASL asociada a una imagen CRA o BLA precederá a cualquier imagen RADL asociada a la imagen CRA o BLA en orden de salida.

[0055] Cualquier imagen RASL asociada a una imagen CRA deberá seguir, en orden de salida, a cualquier imagen IRAP que preceda a la imagen CRA en orden de descodificación.

[0056] Cuando `sps_temporal_id_nesting_flag` es igual a 1 y `TemporalId` es mayor que 0, el `nal_unit_type` será igual a `TSA_R`, `TSA_N`, `RADL_R`, `RADL_N`, `RASL_R` o `RASL_N`.

[0057] `nuh_layer_id` será igual a 0. Otros valores de `id` de capa `nuh` pueden especificarse en el futuro por la ITU-T | ISO/IEC. Para fines distintos a la determinación de la cantidad de datos en las unidades de descodificación del flujo de bits (como se especifica en el Anexo C), los descodificadores ignorarán (es decir, eliminarán del flujo de bits y descartarán) todas las unidades NAL con valores de `id` de capa `nuh` distintos de 0.

NOTA 8 - Se anticipa que en futuras ampliaciones de la codificación de vídeo, escalables o en 3D, este elemento sintáctico se usará para identificar capas adicionales que pueden estar presentes en la CVS, donde una capa puede ser, por ejemplo, una capa espacial escalable, una capa de calidad escalable, una vista de textura o una vista en profundidad.

[0058] Si `nal_unit_type` es igual a `ELDISC_R` o `ELDISC_N`, es decir, el segmento codificado pertenece a una imagen `ELDISC`, el valor de `nuh_layer_id` será mayor que cero.

[0059] De forma alternativa, si `nal_unit_type` es igual a `ELDISC_R` o `ELDISC_N`, es decir, el segmento codificado pertenece a una imagen `ELDISC`, el valor de `id` de capa `nuh` se restringirá como sigue:

- si la AU actual es una AU IRAP, `id` de capa `nuh` será mayor que el ID de capa de acceso de la AU actual.

- de lo contrario, el valor de `id` de capa `nuh` será mayor que el ID de capa de acceso de la AU IRAP anterior.

[0060] `nuh_temporal_id_plus1` menos 1 especifica un identificador temporal para la unidad NAL. El valor de `nuh_temporal_id_plus1` no será igual a 0.

[0061] La variable `TemporalId` se especifica de la siguiente forma:

$$\text{TemporalId} = \text{nuh_temporal_id_plus1} - 1 \quad (7-1)$$

[0062] Si `nal_unit_type` está en el intervalo de `BLA_W_LP` a `RSV_IRAP_VCL23`, inclusive, es decir, el segmento codificado pertenece a una imagen IRAP, `TemporalId` será igual a 0. De lo contrario, cuando `nal_unit_type` es igual a `TSA_R`, `TSA_N`, `STSA_R` o `STSA_N`, `TemporalId` no será igual a 0.

[0063] El valor de `TemporalId` será el mismo para todas las unidades NAL VCL de una unidad de acceso. El valor de `TemporalId` de una unidad de acceso es el valor de `TemporalId` de las unidades NAL VCL de la unidad de acceso.

[0064] El valor de `TemporalId` para unidades NAL que no son VCL está restringido como sigue:

- Si `nal_unit_type` es igual a `VPS_NUT` o `SPS_NUT`, `TemporalId` será igual a 0 y `TemporalId` de la unidad de acceso que contiene la unidad NAL será igual a 0.

- De lo contrario, si `nal_unit_type` es igual a `EOS_NUT` o `EOB_NUT`, `TemporalId` será igual a 0.

- De lo contrario, si `nal_unit_type` es igual a `AUD_NUT` o `FD_NUT`, `TemporalId` será igual a `TemporalId` de la unidad de acceso que contiene la unidad NAL.

- De lo contrario, `TemporalId` será mayor que o igual a `TemporalId` de la unidad de acceso que contiene la unidad NAL.

NOTA 9 - Cuando la unidad NAL es una unidad NAL no VCL, el valor de `TemporalId` es igual al valor mínimo de los valores de `TemporalId` de todas las unidades de acceso a las que se aplica la unidad NAL no VCL. Cuando `nal_unit_type` es igual a `PPS_NUT`, `TemporalId` puede ser mayor que o igual a `TemporalId` de la unidad de acceso contenedora, ya que todos los PPS pueden incluirse al comienzo de un flujo de bits, en el que la primera imagen codificada tiene `TemporalId` igual a 0. Cuando `nal_unit_type` es igual a `PREFIX_SEI_NUT` o `SUFFIX_SEI_NUT`, `TemporalId` puede ser mayor que o igual al `TemporalId` de la unidad de acceso contenedora, ya que una unidad NAL SEI puede contener información, por ejemplo, en un mensaje SEI de período de almacenamiento temporal o en un mensaje SEI de temporización de imagen, que se aplica a un subconjunto de flujo de bits que incluye unidades de acceso para las cuales los valores de `TemporalId` son mayores que el `TemporalId` de la unidad de acceso que contiene la unidad NAL SEI.

[0065] De forma alternativa, no hay nuevos tipos de unidades NAL para las imágenes ELDISC. Aquí se debe tener en cuenta que las imágenes ELDISC pueden ser una imagen final, una imagen TSA o una imagen STSA.

[0066] A continuación se describe el proceso de decodificación general.

[0067] El proceso de decodificación es el mismo que el definido en la subcláusula F.8 en SHVC WD1, con algunos cambios, y los cambios en cada subcláusula modificada en el proceso de decodificación para SHVC WD1 y HEVC WD10 se dan a continuación.

[0068] A continuación se describen los cambios en la subcláusula F.8.1 en SHVC WD1.

[0069] Las especificaciones en la subcláusula 8.1 se aplican con las siguientes adiciones. Cuando la imagen actual tiene `nuh_layer_id` mayor que 0, se aplica lo siguiente.

- Dependiendo del valor de `separate_colour_plane_flag`, el proceso de decodificación se estructura como sigue:

- Si `separate_colour_plane_flag` es igual a 0, el siguiente proceso de decodificación se invoca una sola vez siendo la imagen actual la salida.

- De lo contrario (`separate_colour_plane_flag` es igual a 1), el siguiente proceso de decodificación se invoca tres veces. Las entradas al proceso de decodificación son todas las unidades NAL de la imagen codificada con el valor idéntico de `colour_plane_id`. El proceso de decodificación de unidades NAL con un valor particular de `colour_plane_id` se especifica como si solo un CVS con formato de color monocromo con ese valor particular de `colour_plane_id` estuviera presente en el flujo de bits. La salida de cada uno de los tres procesos de decodificación se asigna a una de las 3 matrices de muestra de la imagen actual, con las unidades NAL con `colour_plane_id` igual a 0, 1 y 2 asignadas a `SL`, `SCb` y `SCr`, respectivamente.

NOTA - La variable `ChromaArrayType` toma el valor 0 cuando `separate_colour_plane_flag` es igual a 1 y `chroma_format_idc` es igual a 3. En el proceso de decodificación, el valor de esta variable se evalúa, dando como resultado operaciones idénticas a las de las imágenes monocromo (cuando `chroma_format_idc` es igual a 0).

- El proceso de decodificación funciona como sigue para la imagen actual `CurrPic`.

- La variable `NoELDiscPicDecodeFlag` para cada imagen se obtiene como se especifica en la sección titulada *Obtención de NoELDiscPicDecodeFlag*

- Si `CurrPic` tiene `nuh_layer_id` igual a cero y pertenece a una AU IRAP, la variable `NoRasOutputFlag` para la AU se obtiene como sigue

◦ Si la imagen con `nuh_layer_id` igual a 0 en la AU IRAP tiene `NoRasOutputFlag` igual a 1, entonces la variable `NoRasOutputFlag` para la AU IRAP actual se establece igual a 1.

◦ De lo contrario, `NoRasOutputFlag` para la AU IRAP actual se establece igual a 0.

- Para la decodificación de la cabecera de segmento del primer fragmento, en orden de decodificación, de la imagen actual, se invoca el proceso de decodificación para iniciar la decodificación de una imagen codificada con un id de capa `nuh` mayor que 0 especificado en la subcláusula F.8.1.1.

- Si ViewId[nuh_layer_id] es mayor que 0, se invoca el proceso de decodificación para una imagen codificada con un id de capa nuh mayor que 0 especificado en la subcláusula G.8.1.
- 5 - De lo contrario, cuando DependencyId[nuh_layer_id] es mayor que 0, se invoca el proceso de decodificación para una imagen codificada con un id de capa nuh mayor que 0 especificado en la subcláusula F.8.1.1.
- Después de que todos los fragmentos de la imagen actual se hayan decodificado, se invoca el proceso de decodificación para finalizar la decodificación de una imagen codificada con un id de capa nuh mayor que 0 especificado en la subcláusula F.8.1.2.

Obtención de NoELDiscPicDecodeFlag

[0070] La variable NoELDiscPicDecodeFlag se obtiene de la siguiente forma:

- Cuando la AU actual es una AU IRAP, se aplica lo siguiente:

◦ Si NoRaslOutputFlag es igual a 1 para la imagen con nuh_layer_id igual a 0 en la AU actual, se aplica lo siguiente:

▪ Si nuh_layer_id de la imagen actual es mayor que el ID de capa de acceso de la AU actual, NoELDiscPicDecodeFlag se establece igual a 1.

▪ De lo contrario, NoELDiscPicDecodeFlag se establece igual a 0.

◦ De lo contrario (la imagen con nuh_layer_id igual a 0 en la AU actual es una imagen CRA con NoRaslOutputFlag igual a 0), se aplica lo siguiente:

▪ Si la imagen actual es una imagen IRAP,

NoELDiscPicDecodeFlag se establece igual a 0.

▪ De lo contrario, NoELDiscPicDecodeFlag se establece igual al valor de NoELDiscPicDecodeFlag de la imagen anterior, en orden de decodificación, en la misma capa.

- De lo contrario, se aplica lo siguiente:

◦ NoELDiscPicDecodeFlag se establece igual al valor de NoELDiscPicDecodeFlag de la imagen anterior, en orden de decodificación, en la misma capa.

[0071] De forma alternativa, NoELDiscPicDecodeFlag no se infiere para imágenes ELDISC, sino que se señala en la cabecera de fragmento o se especifica por medios externos.

[0072] De forma alternativa, para cada AU IRAP se indica un conjunto de capas que indica qué imágenes son IRAP en esa AU particular; las imágenes en la AU IRAP cuyas ID de capa no están incluidas en el conjunto de capas se considerarán no decodificables. Se puede inferir que el valor de NoELDiscPicDecodeFlag es igual a uno para todas las imágenes en esa CVS que no estén incluidas en el conjunto de capas indicado. Este conjunto de capas puede señalarse mediante un mensaje SEI o por medios externos.

[0073] A continuación se describen los cambios en la subcláusula F.8.1.1 en SHVC WD1.

Cada imagen a la que se hace referencia en esta subcláusula es una imagen completa codificada.

El proceso de decodificación funciona como sigue para la imagen actual CurrPic:

1. La decodificación de las unidades NAL se especifica en la subcláusula 8.2.

2. Los procesos en la subcláusula 8.3 especifican los siguientes procesos de decodificación utilizando elementos sintácticos en la capa de segmento y superiores:

- Las variables y funciones relacionadas con el recuento de orden de imágenes se obtienen en la subcláusula 8.3.1. Esto debe invocarse solo para el primer segmento de una imagen. [eliminado: "Es un requisito de conformidad con el flujo de bits que PicOrderCntVal permanezca sin cambios dentro de una unidad de acceso"].

- El proceso de decodificación para RPS en la subcláusula 8.3.2 se invoca para imágenes con nuh_layer_id igual al de CurrPic, donde las imágenes de referencia pueden marcarse como "no utilizadas como referencia" o "usadas como referencia a largo plazo". Esto debe invocarse solo para el primer segmento de una imagen.

- Cuando CurrPic es una imagen BLA o es una imagen CRA con NoRaslOutputFlag igual a 1 o una imagen ELDISC con NoELDiscPicDecodeFlag igual a 1 y pertenece a una AU IRAP, se invoca el proceso de descodificación para generar imágenes de referencia no disponibles especificadas en la subcláusula 8.3.3, que debe invocarse solo para el primer segmento de una imagen.

[0074] A continuación se describen los cambios en la subcláusula F.8.1.2 en SHVC WD1. PicOutputFlag se establece como sigue:

- Si la imagen actual es una imagen RASL y NoRaslOutputFlag de la imagen IRAP asociada es igual a 1, o una imagen ELDISC con NoELDiscPicDecodeFlag igual a 1, PicOutputFlag se establece igual a 0.

- De lo contrario, PicOutputFlag se establece igual a pic_output_flag.

[0075] Se aplica lo siguiente:

- La imagen descodificada se marca como "utilizada como referencia a corto plazo".

- Cuando TemporalId es igual a HighestTid se invoca el proceso de marcado para imágenes de subcapa de no referencia no necesarias para la predicción intercapa especificada en la subcláusula F.8.1.2.1 con latestDecLayerId igual a id de capa nuh como entrada.

[0076] A continuación se describen cambios en la subcláusula 8.3.1 en HEVC WD10.

[0077] La salida de este proceso es PicOrderCntVal, el recuento de orden de imágenes de la imagen actual.

[0078] Los recuentos de orden de imágenes se usan para identificar imágenes, para obtener parámetros de movimiento en modo de combinación y predicción de vectores de movimiento, y para la verificación de conformidad del descodificador (véase la subcláusula C.5).

[0079] Cada imagen codificada está asociada a una variable de recuento de orden de imágenes, denotada como PicOrderCntVal.

[0080] Cuando la imagen actual *no pertenece a* [eliminado: "no es"] una *unidad de acceso* IRAP [eliminado: "imagen"] con NoRaslOutputFlag igual a 1, las variables prevPicOrderCntLsb y prevPicOrderCntMsb se obtienen como sigue:

- Sea prevTid0Pic la imagen anterior en el orden de descodificación que tiene TemporalId igual a 0 y que no es una imagen RASL, una imagen RADL o una imagen de subcapa de no referencia.

- La variable prevPicOrderCntLsb se establece igual a slice_pic_order_cnt_lsb de prevTid0Pic.

- La variable prevPicOrderCntMsb se establece igual a PicOrderCntMsb de prevTid0Pic.

[0081] La variable PicOrderCntMsb de la imagen actual se obtiene como sigue:

- Si la imagen actual *pertenece a* [eliminado: "es"] una *unidad de acceso* IRAP [eliminado: "imagen"] con NoRaslOutputFlag igual a 1, PicOrderCntMsb se establece igual a 0.

- De lo contrario, PicOrderCntMsb se obtiene como sigue:

```

        if((slice_pic_order_cnt_lsb < prevPicOrderCntLsb) &&
           ((prevPicOrderCntLsb - slice_pic_order_cnt_lsb) >= (
55         MaxPicOrderCntLsb / 2)))
            PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb (8-1)
        else if((slice_pic_order_cnt_lsb > prevPicOrderCntLsb) &&
           ((slice_pic_order_cnt_lsb - prevPicOrderCntLsb) > (MaxPicOrderCntLsb
60         /2)))
            PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
        else
            PicOrderCntMsb = prevPicOrderCntMsb
            PicOrderCntVal se obtiene como sigue:
            PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb (8-2)

```

NOTA 1 - Todas las imágenes IDR tendrán PicOrderCntVal igual a 0, ya que se infiere que slice_pic_order_cnt_lsb es 0 para las imágenes IDR y prevPicOrderCntLsb y prevPicOrderCntMsb se han establecido ambos a 0.

[0082] El valor de PicOrderCntVal estará en el intervalo de -2^{31} a $2^{31} - 1$, inclusivos. En una CVS, los valores de PicOrderCntVal para dos imágenes codificadas cualesquiera no serán iguales.

5 **[0083]** La función PicOrderCnt(picX) se especifica como sigue:

$$\text{PicOrderCnt}(\text{picX}) = \text{PicOrderCntVal de la imagen picX} \quad (8-3)$$

10 **[0084]** La función DiffPicOrderCnt(picA, picB) se especifica como sigue:

$$\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = \text{PicOrderCnt}(\text{picA}) - \text{PicOrderCnt}(\text{picB}) \quad (8-4)$$

[0085] El flujo de bits no contendrá datos que den como resultado valores de

15 DiffPicOrderCnt (picA, picB) usados en el proceso de descodificación que no estén en el intervalo de -2^{15} a $2^{15} - 1$, inclusivos.

NOTA 2 - Sea X la imagen actual e Y y Z otras dos imágenes en la misma secuencia, se considera que Y y Z están en la misma dirección de orden de salida desde X cuando tanto DiffPicOrderCnt(X, Y) como DiffPicOrderCnt(X, Z) son positivos o ambos son negativos.

[0086] A continuación se explican los cambios en la subcláusula 8.3.2 en HEVC WD10.

25 **[0087]** Este proceso se invoca una vez por imagen, después de la descodificación de una cabecera de fragmento, pero antes de la descodificación de cualquier unidad de codificación y antes del proceso de descodificación para la creación de la lista de imágenes de referencia del fragmento, como se especifica en la subcláusula 8.3.3. Este proceso puede dar como resultado que una o más imágenes de referencia en el DPB se marquen como "no usadas como referencia" o "usadas como referencia a largo plazo".

30 **NOTA 1** - el RPS es una descripción absoluta de las imágenes de referencia usadas en el proceso de descodificación de las imágenes codificadas actuales y futuras. La señalización de RPS es explícita en el sentido de que todas las imágenes de referencia incluidas en el RPS se enumeran explícitamente.

35 **[0088]** Una imagen descodificada en el DPB puede marcarse como "no usada como referencia", "usada como referencia a corto plazo" o "usada como referencia a largo plazo", pero solo una de estas tres en un momento dado durante el funcionamiento del proceso de descodificación. La asignación de una de estas marcas a una imagen elimina de manera implícita otra de estas marcas cuando corresponda. Cuando se hace referencia a una imagen que está marcada como "usada como referencia", esto se refiere conjuntamente a la imagen marcada como "usada como referencia a corto plazo" o "usada como referencia a largo plazo" (pero no ambas).

40 **[0089]** Cuando la imagen actual *pertenece* a una AU IRAP [eliminado: "imagen"] con NoRaslOutputFlag igual a 1, todas las imágenes de referencia que se encuentran actualmente en el DPB (si lo hubiera) están marcadas como "no usadas como referencia".

45 **[0090]** Las imágenes de referencia a corto plazo se identifican mediante sus valores PicOrderCntVal. Las imágenes de referencia a largo plazo se identifican mediante sus valores PicOrderCntVal o sus valores slice_pic_order_cnt_lsb.

50 - Cuando la imagen actual es una imagen que sigue, en orden de descodificación, a una imagen STSA que tiene TemporalId igual al de la imagen actual, no habrá ninguna imagen que tenga el TemporalId igual al de la imagen actual incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetLtCurr que preceda a la imagen STSA en orden de descodificación.

55 - Cuando la imagen actual es una imagen CRA, no habrá ninguna imagen incluida en el RPS que preceda, en orden de descodificación, a ninguna imagen IRAP precedente en orden de descodificación (cuando esté presente).

- Cuando la imagen actual es una imagen final *que no es una imagen ELDISC con NoELDiscPicDecodeFlag igual a 1*, no habrá ninguna imagen en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetLtCurr que fue generada por el proceso de descodificación para generar imágenes de referencia no disponibles, como se especifica en la subcláusula 8.3.3.

60 - Cuando la imagen actual es una imagen final, no habrá ninguna imagen en el RPS que preceda a la imagen IRAP asociada en orden de salida o en orden de descodificación.

[0091] A continuación se describe un segundo ejemplo. Este ejemplo es similar al primer ejemplo, pero no se define ningún tipo nuevo de unidad NAL. Se usa una bandera reservada en la cabecera de fragmento para indicar si el fragmento pertenece a una imagen que pertenece a una AU IRAP.

5 Sintaxis de segmento de cabecera de fragmento

[0092]

slice segment header() {	Descriptor
first_slice_segment_in_pic_flag	u(1)
if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23)	
no_output_of_prior_pics_flag	u(1)
slice_pic_parameter_set_id	ue(v)
if(!first_slice_segment_in_pic_flag) {	
if(dependent_slice_segments_enabled_flag)	
dependent_slice_segment_flag	u(1)
slice_segment_address	u(v)
}	
if(!dependent_slice_segment_flag) {	
slice_in_irap_au_flag	u(1)
for(i = 0; i < num_extra_slice_header_bits - 1; i++)	
slice_reserved_flag[i]	u(1)
slice_type	ue(v)
...	

10 **[0093]** Los elementos sintácticos de num_extra_slice_header_bits se cambian como se muestra a continuación, donde las *palabras encursivas* representan adiciones y [eliminado: ""] representa eliminaciones.

15 **num_extra_slice_header_bits** [eliminado: "igual a 0 especifica que no"] *indica el número de bits de cabecera de fragmento adicionales que están presentes en el RBSP de cabecera de fragmento para imágenes codificadas que se refieren al PPS. num_extra_slice_header_bits será igual a 0 o mayor que 1* [eliminado: "0 en flujos de bits conforme a esta versión de esta memoria descriptiva"]. *Otros valores para num_extra_slice_header_bits está reservado para su uso futuro por ITU-T | ISO/IEC. Sin embargo, los decodificadores permitirán que num_extra_slice_header_bits tenga cualquier valor.*

20 **[0094]** La sintaxis de slice_in_irap_au_flag se especifica como sigue:

slice_in_irap_au_flag *igual a 1 especifica que el fragmento está presente en una imagen que pertenece a una AU IRAP. slice_in_irap_au_flag igual a 0 especifica que el fragmento está presente en una imagen que no pertenece a una AU IRAP.*

25 **[0095]** A continuación se describe un tercer ejemplo. En este ejemplo, las AU IRAP incluyen cualquier AU que incluya una imagen IRAP con nuh_layer_id igual a 0. Se usan dos tipos de unidades IRAP NAL adicionales para indicar la primera imagen IDR en cada capa en el flujo de bits, cuando la descodificación comienza desde una AU IRAP que es la AU actual o la AU IRAP anterior en orden de descodificación. Además, se usarían dos tipos más de unidades NAL para indicar aquellas imágenes que no se podrían descodificar cuando se empalma el flujo de bits. Se supone que las imágenes BLA y las imágenes IDR pueden no estar alineadas en varias capas.

[0096] Este ejemplo supone las siguientes definiciones:

35 **Unidad de acceso (AU) IRAP:** Una unidad de acceso que contiene una o más imágenes IRAP donde la imagen en la AU con id de capa nuh igual a 0 es una imagen IRAP.

AU IRAP **alineada:** Una AU IRAP en la que todas las imágenes de la AU son imágenes IRAP.

40 AU IRAP **parcialmente alineada:** Una AU IRAP que no es una AU IRAP alineada y para cada imagen IRAP en la AU con nuh_layer_id igual a layerId, cada imagen en la AU con un valor nuh_layer_id menor que layerId, es también una imagen IRAP.

AU IRAP **AU no alineada:** Una AU IRAP que no es una AU IRAP alineada ni una AU IRAP parcialmente alineada.

45

AU IRAP de conmutación de capa: Una AU que no es una AU IRAP y contiene al menos una imagen IRAP.

ID de capa de acceso: El ID de capa de acceso de una AU IRAP es el mayor valor de layerId de id de capa nuh para el que todas las imágenes en la AU con id de capa nuh menor que o igual a layerId son imágenes IRAP.

Imagen de omisión de acceso aleatorio de varias capas: Una imagen con id de capa nuh igual a layerId que pertenece a una AU IRAP con ID de capa de acceso menor que layerId, o que pertenece a una AU que sigue, en orden de descodificación, a una AU IRAP con ID de capa de acceso menor que layerId y que precede, en orden de descodificación, a la siguiente AU IRAP y que también precede, en orden de descodificación, a una AU IRAP de conmutación de capa que contiene una imagen IRAP con una id de capa nuh igual a layerId.

CL-RASP: una imagen con nal_unit_type igual a CL_RAS_N o CL_RAS_R.

[0097] Es un requisito de conformidad con el flujo de bits que una imagen de omisión de acceso aleatorio de varias capas que pertenezca a una AU IRAP sea una imagen de referencia de subcapa.

imagen de actualización instantánea de descodificación (IDR): Una imagen IRAP para la que cada unidad NAL VCL tiene nal_unit_type igual a BL_IDR_W_RADL, BL_IDR_N_LP, IDR_W_RADL o IDR_N_LP.

NOTA 7 - Una imagen IDR solo contiene fragmentos I, y puede ser la primera imagen en el flujo de bits en orden de descodificación, o puede aparecer después en el flujo de bits. Cada imagen IDR con nuh_layer_id igual a 0 es la primera imagen de una CVS en orden de descodificación. Cuando una imagen IDR para la cual cada unidad NAL VCL tiene nal_unit_type igual a IDR_W_RADL, puede tener imágenes RADL asociadas. Cuando una imagen IDR para la cual cada unidad NAL VCL tiene nal_unit_type igual a IDR_N_LP, no tiene ninguna imagen inicial asociada. Una imagen IDR no tiene imágenes RASL asociadas.

[0098] De forma alternativa, se añade la siguiente restricción:

Es un requisito de conformidad con el flujo de bits que para cada AU IRAP de conmutación de capa actual que tenga una imagen IRAP con un id de capa nuh igual a layerId y cuya AU IRAP precedente en orden de descodificación tenga un valor de ID de capa de acceso menor que nuh_layer_id, deberá haber al menos una AU de conmutación de capa que tenga una imagen IRAP con un id de capa nuh igual a layer_id_in_nuh [LayerIdInVps[layerId] - 1] y que preceda, en orden de descodificación, a la AU IRAP de conmutación de capa actual y que suceda, en orden de descodificación, a la AU IRAP anterior.

[0099] A continuación se describe la semántica de cabecera de unidad NAL.

[0100] Se presentan cuatro tipos de unidades NAL que se muestran en la tabla ofrecida a continuación. Los cambios en la semántica del tipo de unidades NAL se representan mediante *texto en cursiva* para representar adiciones y [eliminado: ""] para representar eliminaciones.

forbidden_zero_bit será igual a 0.

nal_unit_type especifica el tipo de estructura de datos RBSP contenida en la unidad NAL como se especifica en la Tabla 7-1.

[0101] Las unidades NAL que tienen nal_unit_type en el intervalo de UNSPEC48..UNSPEC63, inclusive, para las cuales no se especifica la semántica, no afectarán al proceso de descodificación especificado en esta memoria descriptiva.

NOTA 1 - Los tipos de unidades NAL en el intervalo de UNSPEC48..UNSPEC63 se pueden usar según lo determine la aplicación. En esta memoria descriptiva no se especifica ningún proceso de descodificación para estos valores de nal_unit_type. Dado que diferentes aplicaciones podrían usar estos tipos de unidades NAL para diferentes propósitos, se debe tener especial cuidado en el diseño de los codificadores que generan unidades NAL con estos valores nal_unit_type, y en el diseño de descodificadores que interpretan el contenido de las unidades NAL con estos valores nal_unit_type.

[0102] Para propósitos distintos a la determinación de la cantidad de datos en las unidades de descodificación del flujo de bits (como se especifica en el Anexo C), los descodificadores ignorarán (eliminarán del flujo de bits y descartarán) el contenido de todas las unidades NAL que usan valores reservados de nal_unit_type.

NOTA 2 - Este requisito permite la definición futura de ampliaciones compatibles con esta memoria descriptiva.

Tabla 7-1 - Códigos de tipos de unidad NAL y clases de tipos de unidad NAL

nal_unit_type	Nombre de nal_unit_type	Contenido de la unidad NAL y estructura sintáctica RBSP	Clase de tipo de unidad de NAL
0 1	TRAIL_N TRAIL_R	Segmento codificado de una slice_segment_layer_rbsp() de imagen final no TSA ni STSA	VCL
2 3	TSA_N TSA_R	Segmento codificado de una slice_segment_layer_rbsp() de imagen TSA	VCL
4 5	STSA_N STSA_R	Segmento codificado de una slice_segment_layer_rbsp() de imagen STSA	VCL
6 7	RADL_N RADL_R	Segmento codificado de una slice_segment_layer_rbsp() de imagen RADL	VCL
8 9	RASL_N RASL_R	Segmento codificado de una slice_segment_layer_rbsp() de imagen RASL	VCL
10 12 14	RSV_VCL_N10 RS V_VCL_N12 RSV_VCL_N14	Tipos reservados de unidades NAL VCL de no referencia de subcapa no IRAP	VCL
11 13 15	RSV_VCL_R11 RSV_VCL_R13 RSV_VCL_R15	Tipos reservados de unidades NAL VCL de referencia de subcapa no IRAP	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_N_LP	Segmento codificado de una slice_segment_layer_rbs() de imagen BLA	VCL
19 20	IDR_W_RADL IDR_N_LP	Segmento codificado de una slice_segment_layer_rbsp() de imagen IDR	VCL
21	CRA_NUT	Segmento codificado de una slice_segment_layer_rbsp() de imagen CRA	VCL
22 23	BL_IDR_W_RADL BL_IDR_N_LP	Segmentos codificados de una slice_segment_layer_rbsp() de imagen BL-IDR	VCL
24 25	CL_RAS_N CL_RAS_R	Segmento codificado de una imagen de omisión de acceso aleatorio limpio	VCL
[eliminado: "24"]26..31	RSV_VCL24.. RSV_VCL31	Tipos reservados de unidades NAL VCL no IRAP	VCL
32	VPS_NUT	Conjunto de parámetros de vídeo video_parameter_set_rbsp()	no VCL
33	SPS_NUT	Conjunto de parámetros de secuencia seq_parameter_set_rbsp()	no VCL
34	PPS_NUT	Conjunto de parámetros de imagen pic_parameter_set_rbsp()	no VCL
35	AUD_NUT	Delimitador de unidad de acceso access_unit_delimiter_rbsp()	no VCL
36	EOS_NUT	Fin de secuencia end_of_seq_rbsp()	no VCL
37	EOB_NUT	Fin de flujo de bits end_of_bitstream_rbsp()	no VCL
38	FD_NUT	Datos de relleno filler_data_rbsp()	no VCL
39 40	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Información complementaria de mejora sei_rbsp()	no VCL
41..47	RSV_NVCL41.. RSV_NVCL47	Reservado	no VCL
48..63	UNSPEC48.. UNSPEC63	Sin especificar	no VCL

NOTA 3 - Una imagen CRA puede tener imágenes RASL o RADL asociadas presentes en el flujo de bits.

NOTA 4 - Una imagen BLA que tenga `nal_unit_type` igual a `BLA_W_LP` puede tener imágenes RASL o RADL asociadas presentes en el flujo de bits. Una imagen BLA que tiene `nal_unit_type` igual a `BLA_W_DLP` no tiene imágenes RASL asociadas presentes en el flujo de bits, pero puede tener imágenes RADL asociadas en el flujo de bits. Una imagen BLA que tiene `nal_unit_type` igual a `BLA_N_LP` no tiene imágenes iniciales asociadas presentes en el flujo de bits.

NOTA 5 - Una imagen IDR que tiene `nal_unit_type` igual a `IDR_N_LP` no tiene imágenes iniciales asociadas presentes en el flujo de bits. Una imagen IDR que tiene `nal_unit_type` igual a `IDR_W_RADL` no tiene imágenes RASL asociadas presentes en el flujo de bits, pero puede tener imágenes RADL asociadas en el flujo de bits.

NOTA 6 - No se incluye una imagen de subcapa de no referencia en `RefPicSetStCurrBefore`, `RefPicSetStCurrAfter` ni `RefPicSetLtCurr` de ninguna imagen con el mismo valor de `TemporalId`, y se puede descartar sin afectar a la descodificación de otras imágenes con el mismo valor de `TemporalId`.

[0103] Todas las unidades NAL de segmento codificado de una unidad de acceso tendrán el mismo valor de `nal_unit_type`. También se hace referencia a una imagen o una unidad de acceso que tiene un `nal_unit_type` igual al `nal_unit_type` de las unidades NAL de segmento codificado de la imagen o unidad de acceso.

[0104] Si una imagen tiene `nal_unit_type` igual a `CL_RAS_N`, `TRAIL_N`, `TSA_N`, `STSA_N`, `RADL_N`, `RASL_N`, `RSV_VCL_N10`, `RSV_VCL_N12`, o `RSV_VCL_N14`, la imagen es una imagen de subcapa de no referencia. De lo contrario, la imagen es una imagen de referencia de subcapa.

[0105] *Es un requisito de conformidad con el flujo de bits que una imagen CL-RAS que pertenece a una AU IRAP sea una imagen de referencia de subcapa.*

[0106] Cada imagen, aparte de la primera imagen en el flujo de bits en orden de descodificación, se considera que está asociada a la imagen IRAP anterior en orden de descodificación.

[0107] Cuando una imagen es una imagen inicial, debe ser una imagen RADL o RASL.

[0108] Cuando una imagen es una imagen final, no debe ser una imagen RADL o RASL.

[0109] Cuando una imagen es una imagen inicial, precederá, en orden de descodificación, a todas las imágenes finales que están asociadas a la misma imagen IRAP.

[0110] Ninguna imagen RASL estará presente en el flujo de bits que esté asociado a una imagen BLA que tenga `nal_unit_type` igual a `BLA_W_RADL` o `BLA_N_LP`.

[0111] Ninguna imagen RASL estará presente en el flujo de bits que esté asociado a una imagen IDR.

[0112] Ninguna imagen RADL estará presente en el flujo de bits que esté asociado a una imagen BLA que tenga `nal_unit_type` igual a `BLA_N_LP` o que esté asociado a una imagen IDR que tenga `nal_unit_type` igual a `IDR_N_LP`.

NOTA 7 - Es posible realizar un acceso aleatorio en la posición de una unidad de acceso IRAP descartando todas las unidades de acceso antes de la unidad de acceso IRAP (y descodificar correctamente la imagen IRAP y todas las imágenes posteriores no RASL en el orden de descodificación), siempre cada conjunto de parámetros esté disponible (ya sea en el flujo de bits o por medios externos no especificados en esta memoria descriptiva) cuando sea necesario activarlo.

[0113] Cualquier imagen que tenga `PicOutputFlag` igual a 1 que precede a una imagen IRAP en orden de descodificación deberá preceder a la imagen IRAP en orden de salida y deberá preceder a cualquier imagen RADL asociada a la imagen IRAP en orden de salida.

[0114] Cualquier imagen RASL asociada a una imagen CRA o BLA precederá a cualquier imagen RADL asociada a la imagen CRA o BLA en orden de salida.

[0115] Cualquier imagen RASL asociada a una imagen CRA deberá seguir, en orden de salida, a cualquier imagen IRAP que preceda a la imagen CRA en orden de descodificación.

[0116] Cuando `sps_temporal_id_nesting_flag` es igual a 1 y `TemporalId` es mayor que 0, el `nal_unit_type` será igual a `TSA_R`, `TSA_N`, `RADL_R`, `RADL_N`, `RASL_R` o `RASL_N`.

[0117] `nuh_layer_id` será igual a 0. Otros valores de `id` de capa `nuh` pueden especificarse en el futuro por la ITU-T | ISO/IEC. Para propósitos distintos a la determinación de la cantidad de datos en las unidades de descodificación del flujo de bits (como se especifica en el Anexo C), los descodificadores ignorarán (es decir, eliminarán del flujo de bits y descartarán) todas las unidades NAL con valores de `id` de capa `nuh` distintos de 0.

NOTA 8 - Se anticipa que en futuras ampliaciones de la codificación de vídeo, escalables o en 3D, este elemento sintáctico se usará para identificar capas adicionales que pueden estar presentes en la CVS, donde una capa puede ser, por ejemplo, una capa espacial escalable, una capa de calidad escalable, una vista de textura o una vista en profundidad.

[0118] Si *nal_unit_type* es igual a *CL_RAS_R* o *CL_RAS_N*, es decir, el segmento codificado pertenece a una imagen *CL_RAS*, el valor de *nuh_layer_id* será mayor que cero.

[0119] De forma alternativa, si *nal_unit_type* es igual a *CL_RAS_R* o *CL_RAS_N*, es decir, el segmento codificado pertenece a una imagen de omisión de acceso aleatorio limpio, el valor de *nuh_layer_id* se restringirá como sigue:

- Si la AU actual es una AU IRAP, *nuh_layer_id* será mayor que el ID de capa de acceso de la AU actual.

- de lo contrario, el valor de *nuh_layer_id* será mayor que el ID de la capa de acceso de la AU IRAP precedente en orden de descodificación.

[0120] *nuh_temporal_id_plus1* menos 1 especifica un identificador temporal para la unidad NAL. El valor de *nuh_temporal_id_plus1* no será igual a 0.

[0121] La variable *TemporalId* se especifica como sigue:

$$\text{TemporalId} = \text{nuh_temporal_id_plus1} - 1 \quad (7-1)$$

[0122] Si *nal_unit_type* está en el intervalo de *BLA_W_LP* a *RSV_IRAP_VCL23*, inclusive, es decir, el segmento codificado pertenece a una imagen IRAP, *TemporalId* será igual a 0. De lo contrario, cuando *nal_unit_type* es igual a *TSA_R*, *TSA_N*, *STSA_R* o *STSA_N*, *TemporalId* no será igual a 0.

[0123] El valor de *TemporalId* será el mismo para todas las unidades NAL VCL de una unidad de acceso. El valor de *TemporalId* de una unidad de acceso es el valor de *TemporalId* de las unidades NAL VCL de la unidad de acceso.

[0124] El valor de *TemporalId* para unidades NAL que no son VCL está restringido como sigue:

- Si *nal_unit_type* es igual a *VPS_NUT* o *SPS_NUT*, *TemporalId* será igual a 0 y *TemporalId* de la unidad de acceso que contiene la unidad NAL será igual a 0.

- De lo contrario, si *nal_unit_type* es igual a *EOS_NUT* o *EOB_NUT*, *TemporalId* será igual a 0.

- De lo contrario, si *nal_unit_type* es igual a *AUD_NUT* o *FD_NUT*, *TemporalId* será igual a *TemporalId* de la unidad de acceso que contiene la unidad NAL.

- De lo contrario, *TemporalId* será mayor que o igual a *TemporalId* de la unidad de acceso que contiene la unidad NAL.

NOTA 9 - Cuando la unidad NAL es una unidad NAL no VCL, el valor de *TemporalId* es igual al valor mínimo de los valores de *TemporalId* de todas las unidades de acceso a las que se aplica la unidad NAL no VCL. Cuando *nal_unit_type* es igual a *PPS_NUT*, *TemporalId* puede ser mayor que o igual a *TemporalId* de la unidad de acceso contenedora, ya que todos los PPS pueden incluirse al comienzo de un flujo de bits, en el que la primera imagen codificada tiene *TemporalId* igual a 0. Cuando *nal_unit_type* es igual a *PREFIX_SEI_NUT* o *SUFFIX_SEI_NUT*, *TemporalId* puede ser mayor que o igual al *TemporalId* de la unidad de acceso contenedora, ya que una unidad NAL SEI puede contener información, por ejemplo, en un mensaje SEI de período de almacenamiento temporal o en un mensaje SEI de temporización de imagen, que se aplica a un subconjunto de flujo de bits que incluye unidades de acceso para las cuales los valores de *TemporalId* son mayores que el *TemporalId* de la unidad de acceso que contiene la unidad NAL SEI.

[0125] A continuación se describe el proceso de descodificación general.

[0126] El proceso de descodificación es similar al descrito en el ejemplo 1, donde los cambios realizados en el proceso de descodificación para imágenes ELDISC con *NoELDiscPicDecodeFlag* igual a 1 se aplican a las imágenes CL-RAS.

[0127] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo 10 de ejemplo que puede utilizar las técnicas descritas en esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados que un dispositivo de destino 14 va a descodificar en un momento posterior. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluidos 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 continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0128] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de transferir los datos de vídeo codificados desde el dispositivo de origen 12 hasta el 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 a un dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación, inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. 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 hasta el dispositivo de destino 14.

[0129] De forma alternativa, los datos codificados pueden emitirse desde la interfaz de salida 22 hasta un dispositivo de almacenamiento 34. De forma similar, puede accederse a los datos codificados del dispositivo de almacenamiento 34 mediante una interfaz de entrada. El dispositivo de almacenamiento 34 puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco duro, unos discos Blu-ray, unos DVD, unos 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 otro ejemplo, el dispositivo de almacenamiento 34 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados en el dispositivo de almacenamiento 34 a través de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor del FTP, dispositivos de almacenamiento conectados en 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, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 34 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0130] Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo a cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, a través de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para prestar soporte a una transmisión de vídeo unidireccional o bidireccional, a fin de prestar soporte a aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0131] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir un origen tal como un dispositivo de captación de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo captado previamente, una interfaz de entrada de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo de origen, o una combinación de dichas fuentes. Como un ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación se pueden aplicar a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas.

[0132] El codificador de vídeo 12 puede codificar el vídeo captado, captado previamente o generado por ordenador. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados se pueden almacenar de forma adicional (o alternativa) en el dispositivo de almacenamiento 34 para un posterior acceso por el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción.

[0133] 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 a través del enlace 16. Los datos de

vídeo codificados comunicados a través del 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 descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados transmitidos en un medio de comunicación, almacenarse en un medio de almacenamiento o almacenarse en un servidor de archivos.

[0134] El dispositivo de visualización 32 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse 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 de una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0135] Aunque no se muestra en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos independientes. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexión ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0136] Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden implementarse como cualquiera de entre una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de puertas programables in situ (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 adecuado no transitorio 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 se pueden incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

[0137] El JCT-VC ha desarrollado la norma HEVC. Los trabajos de normalización de HEVC se basan en un modelo de dispositivo de codificación de vídeo de modelo de prueba de HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a los dispositivos existentes, de acuerdo con, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación mediante intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación mediante intrapredicción.

[0138] En general, el modelo de explotación del HM describe que una trama o imagen de vídeo puede dividirse en una secuencia de bloques de árbol o unidades de codificación más grandes (LCU), que incluyen tanto muestras de luminancia como de crominancia. Un bloque de árbol tiene un propósito similar al de un macrobloque de la norma H.264. Un fragmento incluye una pluralidad de bloques de árbol consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque de árbol puede dividirse en unidades de codificación (CU) de acuerdo con un árbol cuaternario. Por ejemplo, un bloque de árbol, tal como un nodo raíz del árbol cuaternario, puede dividirse en cuatro nodos hijo, y cada nodo hijo puede, a su vez, ser un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final, no dividido, tal como un nodo hoja del árbol cuaternario, 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 que puede dividirse un bloque de árbol, y también pueden definir un tamaño mínimo de los nodos de codificación.

[0139] Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol, con un máximo de 64x64 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, la división de la CU en una o más PU. Los modos de división pueden diferir entre si la CU está codificada en modo de omisión o en modo directo, codificada en modo de intrapredicción o codificada en modo de interpredicción. Las PU pueden dividirse para tener forma no cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener forma cuadrada o no cuadrada.

[0140] La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU se basa típicamente en el tamaño de las PU dentro de una CU dada definida para una LCU dividida, aunque puede que no sea siempre así. Las TU son típicamente del mismo tamaño o de un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas utilizando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformada (TU). Los valores de diferencias de

píxeles asociados a las TU pueden transformarse para generar coeficientes de transformada, que pueden cuantificarse.

[0141] En general, una PU incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU se codifica mediante intramodo, la PU puede incluir datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU se codifica mediante intermodo, 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, una componente horizontal del vector de movimiento, una componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o 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.

[0142] En general, se usa una TU para los procesos de transformada y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Tras la predicción, el codificador de vídeo 20 puede calcular valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencias de píxeles que pueden transformarse en coeficientes de transformada, cuantificarse y explorarse usando las TU para generar coeficientes de transformada serializados para la codificación por entropía. Esta divulgación usa típicamente el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término "bloque de vídeo" para referirse a un bloque de árbol, es decir, una LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

[0143] Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una pluralidad de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen un modo de codificación para el fragmento respectivo. El codificador de vídeo 20 actúa típicamente sobre bloques de vídeo dentro de fragmentos 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 diferir en tamaño de acuerdo con una norma de codificación especificada.

[0144] En un ejemplo, el HM admite la predicción en diversos tamaños de PU. Si se supone que el tamaño de una CU particular es $2N \times 2N$, el HM admite la intrapredicción en tamaños de PU de $2N \times 2N$ o $N \times N$, y la interpredicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la interpredicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida entre un 25 % y un 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una "n" seguida de una indicación de "arriba", "abajo", "izquierda" o "derecha". Así pues, por ejemplo, " $2N \times nU$ " se refiere a una CU de tamaño $2N \times 2N$ que está dividida horizontalmente, con una PU de tamaño $2N \times 0,5N$ encima y una PU de tamaño $2N \times 1,5N$ debajo.

[0145] En esta divulgación, " $N \times N$ " y " N por N " pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en cuanto a dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

[0146] Tras la codificación intrapredictiva o interpredictiva, usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de píxeles en el dominio espacial (también denominado dominio de píxel) y las TU pueden comprender coeficientes en el dominio de la transformada tras la aplicación de una transformada, por ejemplo, una transformada discreta de coseno (DCT), una transformada entera, una transformada de ondículas o una transformada conceptualmente similar, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU, incluidos los datos residuales para la CU y, a continuación, transformar las TU para generar coeficientes de transformada para la CU.

[0147] Tras cualquier transformada para generar coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. Por ejemplo, un valor de n bits puede redondearse a la baja hasta un valor de m bits durante la cuantificación, donde n es mayor que m .

[0148] En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados, para producir un vector serializado que se pueda codificar por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía de división en intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

[0149] Para realizar la codificación CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo a transmitir. El contexto se puede referir, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la codificación CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la codificación VLC se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta forma, el uso de la codificación VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

[0150] El codificador de vídeo 20 puede formar un flujo de bits que incluye múltiples capas de datos de vídeo. Por ejemplo, el flujo de bits puede incluir una capa base y una o más capas de mejora de acuerdo con una ampliación escalable de una norma de codificación de vídeo, tal como la ampliación escalable de HEVC (SHVC). Adicionalmente o de forma alternativa, el flujo de bits puede incluir múltiples vistas, por ejemplo, una vista base y una o más vistas dependientes. Por ejemplo, el flujo de bits puede ajustarse a una ampliación de múltiples vistas de HEVC (MV-HEVC) o una ampliación tridimensional de HEVC (3D-HEVC). En esta divulgación, una vista se considera un tipo de capa. Por lo tanto, las referencias a datos de vídeo de múltiples capas incluyen datos de vídeo de múltiples vistas.

[0151] De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede formar datos para el flujo de bits de manera que diferentes capas tengan diferentes frecuencias de imágenes IRAP. Por ejemplo, el codificador de vídeo 20 puede proporcionar una imagen IRAP para una capa base por cada dos segundos de tiempo de reproducción, para una primera capa de mejora cada cinco segundos de tiempo de reproducción, y para una segunda capa de mejora cada diez segundos de tiempo de reproducción. En este ejemplo, el dispositivo de destino 14 puede comenzar el acceso aleatorio para recuperar datos al menos de la capa base en intervalos de reproducción de dos segundos. En otras palabras, en este ejemplo, el dispositivo de destino 14 puede experimentar un retardo de ajuste máximo de dos segundos. Típicamente, los usuarios consideran que al menos algunos datos de vídeo, aunque sean de una calidad relativamente baja, son más deseables que una pantalla en blanco. Al proporcionar un flujo de bits con frecuencias variables para puntos de acceso aleatorios en diferentes capas, el codificador de vídeo 20 puede permitir que el dispositivo de destino 14 reduzca el retardo de ajuste, al tiempo que reduce la velocidad de bits para el flujo de bits a través de las diversas capas de codificación de vídeo.

[0152] Como se analiza anteriormente, una unidad de acceso incluye datos para que todas las imágenes se reproduzcan en un tiempo de salida particular. Por ejemplo, de acuerdo con el ejemplo descrito anteriormente, una unidad de acceso puede incluir una imagen para la capa base, una imagen para la primera capa de mejora y una imagen para la segunda capa de mejora. El descodificador de vídeo 30 puede usar las imágenes de la capa de mejora para mejorar la calidad de la imagen de capa base. Por ejemplo, las imágenes en las capas de mejora pueden incluir datos para mejorar uno o más de una resolución espacial, la calidad de relación señal/ruido (SNR), profundidad de bits de crominancia, diferentes perspectivas de cámara (por ejemplo, para datos de vídeo de múltiples vistas), o similar.

[0153] El codificador de vídeo 20 puede codificar imágenes en la(s) capa(s) de mejora usando intrapredicción, predicción entre capas y/o interpredicción temporal. Para que una imagen que se predice usando la interpredicción temporal se pueda descodificar, las imágenes de referencia de las que depende la imagen deben estar disponibles (por ejemplo, previamente decodificadas). En algunos casos, tales imágenes de referencia no están disponibles, por ejemplo, debido a errores de transmisión. Por consiguiente, el descodificador de vídeo 30 puede configurarse para realizar un proceso de imagen no existente cuando se intenta descodificar una imagen para la cual una o más imágenes de referencia no están disponibles.

[0154] Esta divulgación reconoce que, cuando se realiza un acceso aleatorio comenzando con una unidad de acceso IRAP parcialmente alineada, las imágenes no IRAP de la unidad de acceso IRAP parcialmente alineada y/o las imágenes subsiguientes de las capas de codificación de vídeo, incluidas las imágenes no IRAP de la unidad de acceso, pueden no ser descodificables. Por lo tanto, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede señalar datos que indican imágenes de una secuencia de vídeo codificada, que comienza con una unidad de acceso IRAP parcialmente alineada, que no se pueden descodificar correctamente. De esta manera, el descodificador de vídeo 30 puede usar los datos señalizados que indican que una imagen no se puede descodificar correctamente para omitir la descodificación de la imagen (cuando se realiza un acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada correspondiente). Por lo tanto, el descodificador de vídeo 30 no necesita

intentar descodificar la imagen y realizar el proceso de imagen no existente, sino que puede omitir cualquier intento de descodificar la imagen y simplemente analizar sintácticamente los datos de la imagen. De forma alternativa, el descodificador de vídeo 30 puede determinar realizar el proceso de imagen no existente, por ejemplo, basándose en estos datos señalizados.

[0155] De esta manera, las técnicas de esta divulgación admiten unidades de acceso IRAP parcialmente alineadas, que pueden proporcionar beneficios en cuanto a un retardo de ajuste reducido para el acceso aleatorio y una velocidad de bits reducida para capas de mejora (que pueden incluir vistas dependientes). Del mismo modo, las técnicas de esta divulgación también pueden proporcionar beneficios para los descodificadores de vídeo, tal como el descodificador de vídeo 30, ya que el descodificador de vídeo 30 puede omitir fácilmente las imágenes no descodificables cuando se realiza un acceso aleatorio desde una unidad de acceso IRAP parcialmente alineada.

[0156] La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas descritas en esta divulgación. El codificador de vídeo 20 puede estar configurado para proporcionar vídeo a una entidad de posprocesamiento 27. La entidad de posprocesamiento 27 pretende representar un ejemplo de una entidad de vídeo, tal como un elemento de red con capacidad de medios (MANE) o un dispositivo de empalme/edición, que puede procesar datos de vídeo codificados del codificador de vídeo 20. En algunos casos, la entidad de posprocesamiento puede ser un ejemplo de entidad de red. En algunos sistemas de codificación de vídeo, la entidad de posprocesamiento 27 y el codificador de vídeo 20 pueden formar parte de dispositivos separados, mientras que, en otros casos, la funcionalidad descrita con respecto a la entidad de posprocesamiento 27 se puede realizar por el mismo dispositivo que comprende el codificador de vídeo 20.

[0157] El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) puede referirse a cualquiera de varios modos de compresión espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de varios modos de compresión temporal.

[0158] En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de división 35, una unidad de procesamiento de predicción 41, una unidad de filtro 63, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformada 52, una unidad de cuantificación 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 41 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44 y una unidad de intrapredicción 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantificación inversa 58, una unidad de transformada inversa 60 y un sumador 62. La unidad de filtro 63 está destinada a representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF) y un filtro de desplazamiento adaptativo de muestras (SAO). Aunque la unidad de filtro 63 que se muestra en la FIG. 2 es un filtro de bucle, en otras configuraciones, la unidad de filtro 63 puede implementarse como un filtro postbucle.

[0159] Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe datos de vídeo, y la unidad de división 35 divide los datos en bloques de vídeo. Esta división también puede incluir la división en fragmentos, mosaicos u otras unidades mayores, así como la división de bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de LCU y CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo dentro de un fragmento de vídeo que se va a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 41 puede seleccionar uno entre una pluralidad de posibles modos de codificación, tal como uno entre una pluralidad de modos de intracodificación, o uno entre una pluralidad de modos de intercodificación, para el bloque de vídeo actual basándose en resultados de errores (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

[0160] La unidad de intrapredicción 46, dentro de la unidad de procesamiento de predicción 41, puede realizar la codificación intrapredictiva del bloque de vídeo actual con respecto a uno o más bloques próximos en la misma trama o fragmento que el bloque actual a codificar, para proporcionar compresión espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 de la unidad de procesamiento de predicción 41 realizan la codificación interpretativa del bloque de vídeo actual con respecto a uno o más bloques predictivos de una o más imágenes de referencia para proporcionar una compresión temporal.

[0161] La unidad de estimación de movimiento 42 puede estar configurada para determinar el modo de interpredicción para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar fragmentos de vídeo en la secuencia como fragmentos P, fragmentos B o fragmentos GPB. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento,

realizada por la unidad de estimación de movimiento 42, es el proceso de generación de 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 de una trama o imagen de vídeo actual en relación con un bloque predictivo de una imagen de referencia.

[0162] Un bloque predictivo es un bloque que resulta corresponder estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxel, que puede determinarse mediante una suma de una diferencia absoluta (SAD), una suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de imágenes de referencia 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 píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento relativa a las posiciones de píxel completo y las posiciones fraccionarias de píxel, y proporcionar un vector de movimiento con una precisión de píxel fraccionaria.

[0163] La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, 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 (lista 0) de imágenes de referencia o una segunda lista (lista 1) de imágenes de referencia, cada una de las cuales identifica una o más imágenes de referencia almacenadas en una memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.

[0164] La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar extraer o generar el bloque predictivo en función del vector de movimiento determinado mediante estimación de movimiento, realizando posiblemente interpolaciones hasta una precisión de subpíxel. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia tanto de luminancia como de crominancia. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento 44 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.

[0165] La unidad de intrapredicción 46 puede intrapredecir un bloque actual, como alternativa a la interpredicción llevada a cabo por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de intrapredicción 46 puede determinar un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pases de codificación independientes, y la unidad de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción adecuada a usar, entre los modos probados. Por ejemplo, la unidad de intrapredicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tenga las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que fue codificado para generar el bloque codificado, así como una velocidad de bits (es decir, un número de bits) usada para generar el bloque codificado. La unidad de intrapredicción 46 puede calcular razones a partir de las distorsiones y tasas para los diversos bloques codificados, para determinar qué modo de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0166] En cualquier caso, tras seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 46 puede proporcionar información que indica el modo de intrapredicción seleccionado para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica el modo de intrapredicción} seleccionado de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices de modo de intrapredicción y una pluralidad de tablas modificadas de índices de modo de intrapredicción (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices de modo de intrapredicción y una tabla modificada de índices de modo de intrapredicción a usar para cada uno de los contextos.

[0167] El codificador de vídeo 20 puede determinar codificar una imagen de acceso intraaleatorio (IRAP) de una capa de codificación de vídeo base (por ejemplo, una vista base). Por consiguiente, la unidad de procesamiento de predicción 41 puede configurarse para hacer que la unidad de intrapredicción 46 prediga los bloques de la imagen IRAP usando la intrapredicción. La imagen IRAP puede ser, por ejemplo, una imagen IDR, una imagen CRA o una

imagen BLA. El codificador de vídeo 20 puede determinar además codificar una imagen para una capa de mejora (por ejemplo, una vista no básica) de una unidad de acceso que incluye la imagen IRAP de la capa base como una imagen no IRAP. Por lo tanto, la unidad de procesamiento de predicción 41 puede determinar si predecir los bloques de la imagen no IRAP usando interpredicción (por ejemplo, interpredicción temporal y/o predicción intercapa) usando la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, y/o usando la intrapredicción, realizada por la unidad de intrapredicción 46.

[0168] Cuando el codificador de vídeo 20 determina codificar una imagen de una unidad de acceso como una imagen no IRAP, donde la unidad de acceso también incluye al menos una imagen IRAP, el codificador de vídeo 20 puede codificar además un valor para un elemento sintáctico que indica que la imagen codificada como la imagen no IRAP no se puede descodificar correctamente cuando se realiza un acceso aleatorio comenzando con la unidad de acceso. En esta situación, la unidad de acceso representa una unidad de acceso IRAP parcialmente alineada. Por ejemplo, el codificador de vídeo 20 puede asignar un valor particular para las unidades de capa de abstracción de red (NAL) que encapsulan segmentos de la imagen no IRAP, por ejemplo, valores de 24 o 25, como se muestra en el ejemplo de la Tabla 7-1 anterior. Adicionalmente o de forma alternativa, el codificador de vídeo 20 puede codificar un valor para una bandera que indica que, cuando el acceso aleatorio comienza con la unidad de acceso IRAP parcialmente alineada, las imágenes no IRAP de la unidad de acceso no se pueden descodificar correctamente. Además, el codificador de vídeo 20 puede codificar, adicionalmente o de forma alternativa, datos para otras imágenes de otras unidades de acceso en la misma secuencia de vídeo codificada que indica si estas imágenes pueden descodificarse cuando se realiza el acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada.

[0169] Después de que la unidad de procesamiento de predicción 41 genere el bloque predictivo para el bloque de vídeo actual, ya sea por medio de la interpredicción o la intrapredicción, el codificador de vídeo 20 forma un bloque de vídeo residual sustrayendo el bloque predictivo del bloque de vídeo actual. Los datos de vídeo residuales del bloque residual pueden estar incluidos en una o más TU y aplicarse a la unidad de procesamiento de transformada 52. La unidad de procesamiento de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales usando una transformada, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar. La unidad de procesamiento de transformada 52 puede convertir los datos de vídeo residuales desde un dominio de píxel a un dominio de transformada, tal como un dominio de frecuencia.

[0170] La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar, a continuación, una exploración de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

[0171] Tras la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación de entropía por división en intervalos de probabilidad (PIPE) u otra metodología o técnicas de codificación por entropía. Tras la codificación por entropía mediante la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir 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 el fragmento de vídeo actual que se esté codificando.

[0172] La unidad de cuantificación inversa 58 y la unidad de transformada inversa 60 aplican una cuantificación inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel para su posterior uso como bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores fraccionarios de píxeles, para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción con compensación de movimiento generado por la unidad de compensación de movimiento 44 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden usar el bloque de referencia como bloque de referencia para realizar la interpredicción en un bloque en una trama o imagen de vídeo subsiguiente.

[0173] De esta manera, el codificador de vídeo 20 representa un ejemplo de un codificador de vídeo configurado para codificar una imagen de punto de acceso intraaleatorio (IRAP) de una unidad de acceso IRAP parcialmente alineada de los datos de vídeo, y codificar datos que indican, cuando se realiza el acceso aleatorio desde la unidad de acceso IRAP parcialmente alineada, al menos una imagen de una capa de codificación de vídeo que no se puede descodificar correctamente.

[0174] La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que puede implementar las técnicas descritas en esta divulgación. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 80, una unidad de procesamiento de predicción 81, una unidad de cuantificación inversa 86, una unidad de transformación inversa 88, un sumador 90, una unidad de filtro 91 y una memoria de imágenes de referencia 92. La unidad de procesamiento de predicción 81 incluye una unidad de compensación de movimiento 82 y una unidad de intrapredicción 84. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación que, en general, es recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 2.

[0175] Durante el proceso de descodificación, el descodificador de vídeo 30 recibe, desde el codificador de vídeo 20, datos de un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados. El descodificador de vídeo 30 puede recibir el flujo de bits de vídeo codificado desde una entidad de red 29. La entidad de red 29 puede ser, por ejemplo, un servidor, un MANE, un editor/empalmador de vídeo u otro dispositivo similar configurado para implementar una o más de las técnicas descritas anteriormente. La entidad de red 29 puede incluir o no un codificador de vídeo, tal como el codificador de vídeo 20. La entidad de red 29 puede implementar algunas de las técnicas descritas en esta divulgación, antes de que la entidad de red 29 transmita el flujo de bits de vídeo codificado al descodificador de vídeo 30. En algunos sistemas de descodificación de vídeo, la entidad de red 29 y el descodificador de vídeo 30 pueden formar parte de dispositivos independientes, mientras que, en otros casos, el mismo dispositivo que comprende el descodificador de vídeo 30 puede desempeñar la funcionalidad descrita con respecto a la entidad de red 29.

[0176] De acuerdo con las técnicas de esta divulgación, el descodificador de vídeo 30 puede recibir datos sintácticos (por ejemplo, una bandera o un valor de unidad NAL) que indica si los datos de una imagen se pueden descodificar cuando se realiza un acceso aleatorio desde una unidad de acceso IRAP parcialmente alineada de una secuencia de vídeo codificada que incluye la imagen. El descodificador de vídeo 30 puede determinar además si el acceso aleatorio se realizó desde la unidad de acceso IRAP parcialmente alineada correspondiente. Cuando el descodificador de vídeo 30 determina que se realizó un acceso aleatorio desde la unidad de acceso IRAP parcialmente alineada, el descodificador de vídeo 30 puede omitir la descodificación de imágenes indicadas como no descodificables. El descodificador de vídeo 30 puede determinar que el acceso aleatorio se realiza desde la unidad de acceso IRAP parcialmente alineada cuando la unidad de acceso IRAP parcialmente alineada es la primera unidad de acceso ordinal del flujo de bits a descodificar, y/o cuando una o más imágenes de referencia para las imágenes no IRAP de la unidad de acceso IRAP parcialmente alineada no están presentes en una memoria intermedia de imágenes descodificadas, por ejemplo, de la memoria de imágenes de referencia 92.

[0177] En algunos ejemplos, el descodificador de vídeo 30 puede determinar si una imagen se puede descodificar en función de un tipo de unidad NAL para unidades NAL que encapsulan fragmentos de la imagen. Tal como se analiza anteriormente con respecto a la Tabla 7-1, un valor de tipo de unidad NAL de 24 o 25 puede indicar que una imagen es una imagen de capa de mejora descartable (por ejemplo, una imagen ELDISC), que no se puede descodificar correctamente cuando se realiza un acceso aleatorio desde una unidad de acceso IRAP parcialmente alineada de una secuencia de vídeo codificada que incluye la imagen.

[0178] El descodificador de vídeo 30 puede simplemente analizar sintácticamente los datos de las imágenes indicadas como no descodificables, en lugar de intentar descodificar los datos de dichas imágenes. Cuando se omite la descodificación de dicha imagen, el descodificador de vídeo 30 puede evitar hacer pasar elementos sintácticos para las PU y/o las TU de los bloques de la imagen (por ejemplo, sintaxis de intrapredicción, elementos sintácticos de interpredicción, elementos sintácticos de coeficiente de transformada y similares) a la unidad de procesamiento de predicción 81, la unidad de cuantificación inversa 86 y la unidad de transformada inversa 88. En su lugar, el descodificador de vídeo 30 puede simplemente analizar sintácticamente y descartar los datos, hasta alcanzar una imagen que se pueda descodificar.

[0179] Una norma o ampliación de codificación de vídeo aplicable, tal como SHVC, MV-HEVC o 3D-HEVC, puede especificar que un flujo de bits que comience con una unidad de acceso IRAP parcialmente alineada sea un flujo de bits válido. Por lo tanto, el descodificador de vídeo 30 puede configurarse para analizar sintácticamente dicho flujo de bits de acuerdo con la norma o la ampliación de la norma. Por ejemplo, el descodificador de vídeo 30 puede analizar sintácticamente el flujo de bits de acuerdo con una gramática libre de contexto u otra gramática que corresponda a la norma o ampliación de codificación de vídeo aplicable.

[0180] En lugar de omitir la descodificación de estas imágenes, el descodificador de vídeo 30 puede configurarse para realizar un proceso de imagen inexistente para imágenes señalizadas como que no pueden descodificarse correctamente cuando se realiza un acceso aleatorio desde una unidad de acceso IRAP parcialmente alineada. Por ejemplo, el descodificador de vídeo 30 puede identificar un primer conjunto de imágenes en la unidad de acceso IRAP que no se pueden descodificar correctamente (por ejemplo, en base a los datos señalizados). Para cada imagen del primer conjunto de imágenes, en función de los conjuntos de imágenes de referencia para las imágenes, el descodificador de vídeo 30 puede generar segundos conjuntos respectivos de imágenes de referencia no disponibles. Es decir, es posible que las imágenes de la unidad de acceso IRAP parcialmente alineada no se puedan descodificar correctamente porque no se recibieron las imágenes de referencia de las que dependen estas imágenes. Por lo tanto,

el descodificador de vídeo 30 puede generar las imágenes de referencia de acuerdo con el proceso de imágenes no existentes. El descodificador de vídeo 30 puede marcar además las imágenes del segundo conjunto de imágenes (es decir, las imágenes de referencia generadas) como imágenes de referencia a corto o largo plazo, como se indica en el conjunto de imágenes de referencia respectivo. El descodificador de vídeo 30 puede, a continuación, descodificar cada imagen del primer conjunto de imágenes usando las imágenes de referencia generadas. Debe entenderse que esta descodificación de estas imágenes puede no reproducir con precisión las imágenes.

[0181] La unidad de descodificación por entropía 80 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de descodificación por entropía 80 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 81. El descodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

[0182] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en un modo de intrapredicción señalado y en datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 82 de la unidad de procesamiento de predicción 81 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 80. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede crear las listas de tramas de referencia, lista 0 y lista 1, mediante técnicas de construcción predeterminadas, basándose en imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

[0183] La unidad de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 82 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intrapredicción o interpredicción) usado} para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, un fragmento B, un fragmento P o un fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, el estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

[0184] La unidad de compensación de movimiento 82 también puede realizar la interpolación basándose en 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 píxeles fraccionarios de los 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 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para generar bloques predictivos.

[0185] La unidad de cuantificación inversa 86 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del fragmento de vídeo con el fin de determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debería aplicar. La unidad de transformada inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada entera inversa, o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada con el fin de generar bloques residuales en el dominio de píxel.

[0186] Después de que la unidad de compensación de movimiento 82 genere el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales de la unidad de transformada inversa 88 a los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también pueden usarse filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. La unidad de filtro 91 está destinada a representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF) y un filtro de desplazamiento adaptativo de muestras (SAO). Aunque la unidad de filtro 91 que se muestra en la FIG. 3 es un filtro de bucle, en otras configuraciones, la unidad de filtro 91 puede implementarse como un filtro postbucle. Los bloques de vídeo descodificados en una trama o imagen dada se almacenan a continuación en la memoria de imágenes de referencia 92, que almacena imágenes de referencia usadas para una compensación de movimiento subsiguiente. La memoria de imágenes de referencia 92 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

[0187] De esta manera, el descodificador de vídeo 30 representa un ejemplo de un descodificador de vídeo configurado para descodificar una imagen de punto de acceso intraaleatorio (IRAP) de una unidad de acceso IRAP parcialmente alineada de los datos de vídeo, y para descodificar datos que indican, cuando se realiza un acceso aleatorio desde la unidad de acceso IRAP parcialmente alineada, al menos una imagen de una capa de codificación de vídeo que no se puede descodificar correctamente.

[0188] La FIG. 4 es un diagrama de bloques que ilustra un conjunto de ejemplo de dispositivos que forman parte de la red 100. En este ejemplo, la red 100 incluye dispositivos de encaminamiento 104A, 104B (dispositivos de encaminamiento 104) y un dispositivo de transcodificación 106. Los dispositivos de encaminamiento 104 y el dispositivo de transcodificación 106 están concebidos para representar un pequeño número de dispositivos que pueden formar parte de la red 100. Otros dispositivos de red, tales como conmutadores, concentradores, pasarelas, cortafuegos, puentes y otros dispositivos de este tipo también pueden estar incluidos dentro de la red 100. Además, pueden proporcionarse dispositivos de red adicionales a lo largo de una trayectoria de red entre un dispositivo servidor 102 y un dispositivo cliente 108. El dispositivo servidor 102 puede corresponder a un dispositivo de origen 12 (FIG. 1), mientras que el dispositivo cliente 108 puede corresponder a un dispositivo de destino 14 (FIG. 1), en algunos ejemplos.

[0189] En general, los dispositivos de encaminamiento 104 implementan uno o más protocolos de encaminamiento para intercambiar datos de red a través de la red 100. En algunos ejemplos, los dispositivos de encaminamiento 104 pueden estar configurados para realizar operaciones de apoderado (*proxy*) o de memoria caché. Por lo tanto, en algunos ejemplos, los dispositivos de encaminamiento 104 pueden denominarse dispositivos de apoderado. En general, los dispositivos de encaminamiento 104 ejecutan protocolos de encaminamiento para descubrir rutas a través de la red 100. Al ejecutar dichos protocolos de encaminamiento, un dispositivo de encaminamiento 104B puede descubrir una ruta de red desde sí mismo hasta el dispositivo servidor 102 por medio del dispositivo de encaminamiento 104A.

[0190] Dispositivos de red tales como dichos dispositivos de encaminamiento 104 y dicho dispositivo de transcodificación 106 pueden implementar las técnicas de esta divulgación, aunque un dispositivo cliente 108 también las puede implementar. De esta manera, los dispositivos de encaminamiento 104, el dispositivo de transcodificación 106 y el dispositivo cliente 108 representan ejemplos de dispositivos configurados para realizar las técnicas de esta divulgación. Por otro lado, los dispositivos de la FIG. 1 y el codificador 20 ilustrado en la FIG. 2 y el descodificador 30 ilustrado en la FIG. 3 son también dispositivos a modo de ejemplo que pueden estar configurados para realizar las técnicas de esta divulgación.

[0191] Por ejemplo, el dispositivo servidor 102 puede incluir un codificador de vídeo para codificar una imagen o unas imágenes que se encuentran después de un punto de acceso aleatorio o un punto de adaptación de flujo, tal como un punto de conmutación de capa temporal, u otro punto de adaptación de flujo. Por ejemplo, este punto podría ser un punto de conmutación para la adaptación de la velocidad de bits, la velocidad de trama (es decir, un punto de conmutación de capa temporal) o la resolución espacial. De manera similar, el dispositivo cliente 108 puede descodificar una imagen o imágenes que se encuentran después de un punto de acceso aleatorio o punto de adaptación de flujo, tal como un punto de conmutación de capa temporal.

[0192] El dispositivo de transcodificación 106 puede configurarse para realizar el empalme del flujo de bits. Es decir, el dispositivo de transcodificación 106 puede recibir datos para dos flujos de bits diferentes, y para unir datos de uno de los flujos de bits con datos de otro de los flujos de bits, por ejemplo, en un punto de acceso aleatorio. La FIG. 6, como se explica a continuación, ilustra un ejemplo de un flujo de bits empalmado. El punto en el que los dos flujos de bits se empalman puede comprender una unidad de acceso IRAP parcialmente alineada.

[0193] Además, las técnicas de esta divulgación pueden aplicarse a dicho flujo de bits empalmado. Es decir, el dispositivo servidor 102, el dispositivo de transcodificación 106 u otro dispositivo pueden señalar datos que indican imágenes que no se pueden descodificar correctamente cuando se produce un acceso aleatorio en una unidad de acceso IRAP parcialmente alineada. Como resultado del empalme, los datos del flujo de bits que siguen al punto de empalme pueden considerarse como accedidos de forma aleatoria.

[0194] En algunos ejemplos, el dispositivo de transcodificación 106 puede eliminar las imágenes no descodificables del flujo de bits empalmado. En otros ejemplos, el dispositivo cliente 108 (es decir, un descodificador de vídeo del dispositivo cliente 108) puede analizar sintácticamente, sin descodificar, las imágenes que siguen a la unidad de acceso IRAP parcialmente alineada (que actúa como el punto de empalme) indicadas como no descodificables cuando se produce el acceso aleatorio a partir de la unidad de acceso IRAP parcialmente alineada. Si bien el flujo de bits empalmado podría incluir datos que preceden al punto de empalme, la unidad de acceso IRAP parcialmente alineada que actúa como el punto de empalme puede considerarse como accedido de forma aleatoria para los propósitos de las técnicas de esta divulgación.

[0195] La FIG. 5 es un diagrama conceptual que ilustra varios ejemplos de unidades de acceso. En el ejemplo de la FIG. 5, una unidad de acceso IRAP alineada se considera una unidad de acceso IRAP que incluye solo imágenes

IRAP. En este ejemplo, se considera que una unidad de acceso IRAP parcialmente alineada incluye imágenes IRAP hasta una capa particular y unidades de acceso no IRAP después de esa capa (en orden de descodificación de capa). Otras unidades de acceso, por ejemplo, aquellas en las que hay una imagen no IRAP en una capa debajo de una capa que incluye una imagen IRAP, no se consideran unidades de acceso IRAP en este ejemplo.

[0196] La FIG. 6 es un diagrama conceptual que ilustra un empalme de flujos de bits. En este ejemplo, un flujo de bits empalmado incluye imágenes de un primer flujo de bits 110 e imágenes de un segundo flujo de bits 114, separados por el punto de empalme 112. Dentro de este flujo de bits empalmado, después del punto de empalme 112, las imágenes del segundo flujo de bits 114 comienzan con la unidad de acceso IRAP parcialmente alineada 116. En este ejemplo, la unidad de acceso IRAP parcialmente alineada 116 incluye tres imágenes IRAP y dos imágenes que no son IRAP. Las dos imágenes que no son IRAP se tratan como imágenes de capa de mejora descartables, porque no se pueden descodificar correctamente. Por consiguiente, un codificador de vídeo (por ejemplo, un codificador de vídeo 20) puede codificar datos que indiquen que, cuando se realiza un acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada 116, estas imágenes no se pueden descodificar correctamente. De nuevo, debe entenderse que aunque el descodificador de vídeo puede recibir las imágenes del primer flujo de bits 110, se considera que las imágenes del segundo flujo de bits 114 son accedidas de manera aleatoria, ya que las imágenes del primer flujo de bits 110 no incluyen imágenes de referencia para las imágenes del segundo flujo de bits 114.

[0197] En el ejemplo de la FIG. 6, las imágenes no descodificables (por ejemplo, imágenes de la capa de mejora descartable) están sombreadas en gris. Así, cada una de las imágenes, sombreadas en gris en la FIG. 6, en las capas de codificación de vídeo que contienen imágenes no IRAP de la unidad de acceso IRAP parcialmente alineada 116 que suceden, en orden de descodificación, a la unidad de acceso 116 y que preceden, en orden de descodificación, a la siguiente imagen IRAP en la capa que contiene la imagen no se pueden descodificar correctamente cuando se realiza el acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada. De acuerdo con las técnicas de esta divulgación, un codificador de vídeo (por ejemplo, un codificador de vídeo 20 o un descodificador de vídeo 30) puede codificar (codificar o descodificar) datos que indican que estas imágenes no pueden descodificarse correctamente cuando se realiza un acceso aleatorio comenzando con una unidad de acceso IRAP parcialmente alineada 116.

[0198] La FIG. 6 ilustra además una imagen IRAP 118 y una imagen IRAP 120 en las capas de codificación de vídeo, incluidas imágenes no IRAP de la unidad de acceso IRAP parcialmente alineada 116. Por lo tanto, los datos para las imágenes IRAP 118, 120 y las imágenes que siguen a las imágenes IRAP 118, 120, en las respectivas capas de codificación de vídeo, pueden indicar que estas imágenes son descodificables.

[0199] La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de procedimiento para la codificación de datos de vídeo de acuerdo con las técnicas de esta divulgación. El ejemplo de la FIG. 7 se describe con respecto al codificador de vídeo 20. Sin embargo, debe entenderse que otros dispositivos de codificación de vídeo pueden configurarse para realizar el procedimiento de la FIG. 7 o procedimientos similares.

[0200] En este ejemplo, el codificador de vídeo 20 codifica una imagen IRAP de una unidad de acceso IRAP parcialmente alineada (P.A.) (150). La imagen IRAP puede corresponder a una imagen IRAP de capa base. Aunque no se muestra en la FIG. 7, el codificador de vídeo 20 también puede codificar todas las demás imágenes de la capa, incluida la imagen IRAP, por ejemplo, la capa base (o una imagen de capa inferior). El codificador de vídeo 20 puede codificar a continuación una imagen no IRAP de la misma unidad de acceso IRAP parcialmente alineada (152). La imagen no IRAP puede corresponder a una imagen de una capa de mejora (por ejemplo, una capa encima de la capa que incluye la imagen IRAP).

[0201] El codificador de vídeo 20 puede determinar imágenes no descodificables para un flujo de bits que incluye la unidad de acceso parcialmente alineada (154). Dichas imágenes no descodificables pueden incluir la imagen no IRAP de la unidad de acceso IRAP parcialmente alineada y las imágenes subsiguientes (en orden de codificación) de la misma capa de codificación de vídeo que la que incluye la imagen no IRAP, dentro de la misma secuencia de vídeo codificada. El codificador de vídeo 20 puede codificar a continuación datos que señalizan que estas imágenes son imágenes no descodificables cuando se realiza un acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada (156). Por ejemplo, el codificador de vídeo 20 puede codificar valores para los tipos de unidades NAL de unidades NAL que encapsulan fragmentos de las imágenes que indican que estas imágenes no se pueden descodificar correctamente cuando se realiza un acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada.

[0202] De esta manera, el procedimiento de la FIG. 7 representa un ejemplo de un procedimiento que incluye la codificación de una imagen de punto de acceso intraaleatorio (IRAP) de una unidad de acceso IRAP parcialmente alineada y la codificación de datos que indican, cuando se realiza el acceso aleatorio desde una unidad de acceso de punto de acceso intraaleatorio parcialmente alineado (IRAP), al menos una imagen de una capa de codificación de vídeo que no se puede descodificar correctamente.

[0203] La FIG. 8 es un diagrama de flujo que ilustra un ejemplo de procedimiento de descodificación de datos de vídeo de acuerdo con las técnicas de esta divulgación. En este ejemplo, el procedimiento de la FIG. 8 se describe

como realizado por el descodificador de vídeo 30. Sin embargo, debe entenderse que otros dispositivos de descodificación de vídeo también pueden realizar este procedimiento u otro similar.

[0204] Inicialmente, el descodificador de vídeo 30 puede determinar que el acceso aleatorio se ha realizado comenzando con una unidad de acceso IRAP parcialmente alineada (P.A.) (160). Por ejemplo, el descodificador de vídeo 30 puede determinar las imágenes de referencia que se incluirán en las listas de imágenes de referencia para una o más imágenes no IRAP en la unidad de acceso IRAP parcialmente alineada, y determinar si estas imágenes están presentes en una memoria intermedia de imágenes descodificadas. Un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o una cabecera de segmento para las imágenes que no son IRAP de la unidad de acceso IRAP parcialmente alineada pueden indicar estas imágenes de referencia. Si las imágenes de referencia no están presentes en la memoria intermedia de imágenes descodificadas, el descodificador de vídeo 30 puede determinar que se ha realizado un acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada. De forma alternativa, un dispositivo, unidad o módulo externo al descodificador de vídeo 30 puede indicar que se ha realizado un acceso aleatorio, comenzando con la unidad de acceso IRAP parcialmente alineada.

[0205] En cualquier caso, el descodificador de vídeo 30 puede descodificar una imagen IRAP de la unidad de acceso IRAP parcialmente alineada (162). A continuación, el descodificador de vídeo 30 puede determinar imágenes en y posteriores a la unidad de acceso IRAP parcialmente alineada que se señalizan como no descodificables cuando se realiza un acceso aleatorio comenzando con la unidad de acceso IRAP parcialmente alineada (164). Por ejemplo, el descodificador de vídeo 30 puede determinar que una imagen no se puede descodificar correctamente en función de los tipos de unidades NAL para las unidades NAL que encapsulan fragmentos de la imagen. Basándose en estos datos, el descodificador de vídeo 30 puede descodificar las imágenes descodificables (166) y omitir la descodificación de las imágenes no descodificables (168).

[0206] De esta manera, el procedimiento de la FIG. 8 representa un ejemplo de un procedimiento que incluye descodificar datos que indican, cuando se realiza un acceso aleatorio desde una unidad de acceso de punto de acceso intraaleatorio parcialmente alineada (IRAP), al menos una imagen de una capa de codificación de vídeo que no se puede descodificar correctamente, decodificar una imagen IRAP de la unidad de acceso IRAP, y en función de los datos, omitir la descodificación de la imagen que los datos indican que no se puede descodificar correctamente.

[0207] El procedimiento de la FIG. 8 representa un procedimiento de ejemplo. En otro ejemplo, un procedimiento puede incluir descodificar datos que indican, al realizar un acceso aleatorio desde una unidad de acceso de punto de acceso intraaleatorio parcialmente alineada (IRAP), al menos una imagen de una capa de codificación de vídeo que no se puede descodificar correctamente, descodificar una imagen IRAP de la unidad de acceso IRAP, en función de los datos que indican la al menos una imagen que no se puede descodificar correctamente, identificar un primer conjunto de imágenes en la unidad de acceso IRAP que no se pueden descodificar correctamente, para cada imagen del primer conjunto de imágenes, en función de los conjuntos de imágenes de referencia para las imágenes, generar segundos conjuntos respectivos de imágenes de referencia no disponibles, marcar las imágenes del segundo conjunto de imágenes como imágenes de referencia a corto o largo plazo como se indica en el conjunto de imágenes de referencia respectivo, y descodificar cada imagen del primer conjunto de imágenes.

[0208] 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 se pueden almacenar en o transmitirse por, como una o más instrucciones o código, 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 desde un lugar a otro, por ejemplo, de acuerdo a un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0209] A modo de ejemplo, y no de limitación, 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 recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o 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 ni otros medios

transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), el disco 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 datos ópticamente con láseres. Las combinaciones de lo anterior se deberían incluir también dentro del alcance de los medios legibles por ordenador.

[0210] 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 in situ (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o software dedicados configurados para la codificación y la decodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

[0211] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionarse mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

[0212] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para descodificar datos de vídeo de codificación de vídeo de alta eficiencia, HEVC, donde los datos de vídeo comprenden una pluralidad de capas que incluyen una capa base y una o más capas de mejora, comprendiendo el procedimiento:

descodificar datos que indican, cuando se realiza un acceso aleatorio desde una unidad de acceso de punto de acceso intraaleatorio, IRAP, parcialmente alineada, al menos una imagen de una capa de mejora que no se puede descodificar correctamente, en el que la unidad de acceso IRAP parcialmente alineada incluye una imagen de acceso de enlace roto, BLA, en una capa base y una imagen no IRAP en una capa de mejora; y

descodificar (162) la imagen BLA de la capa base de la unidad de acceso IRAP parcialmente alineada, en el que, para la al menos una imagen que no se puede descodificar correctamente, se invoca un proceso para generar imágenes de referencia no disponibles y la al menos una imagen no se proporciona,

en el que la al menos una imagen de la capa de mejora que no se puede descodificar correctamente comprende la imagen no IRAP en la capa de mejora en la unidad de acceso IRAP parcialmente alineada.

2. Un dispositivo para descodificar datos de vídeo de codificación de vídeo de alta eficiencia, HEVC, donde los datos de vídeo comprenden una pluralidad de capas que incluyen una capa base y una o más capas de mejora, comprendiendo el dispositivo:

medios para descodificar datos que indican, cuando se realiza un acceso aleatorio desde una unidad de acceso IRAP parcialmente alineada, al menos una imagen de una capa de mejora que no se puede descodificar correctamente, en el que la unidad de acceso IRAP parcialmente alineada incluye una imagen de acceso de enlace roto, BLA, en una capa base y una imagen no IRAP en una capa de mejora; y

medios para descodificar (162) la imagen BLA de la capa base de la unidad de acceso IRAP parcialmente alineada, en el que, para la al menos una imagen que no se puede descodificar correctamente, se invoca un proceso para generar imágenes de referencia no disponibles y la al menos una imagen no se proporciona,

en el que la al menos una imagen de la capa de mejora que no se puede descodificar correctamente comprende la imagen no IRAP en la capa de mejora en la unidad de acceso IRAP parcialmente alineada.

3. Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, al ejecutarse, hacen que un procesador de un dispositivo de descodificación de datos de vídeo lleve a cabo el procedimiento según la reivindicación 1.

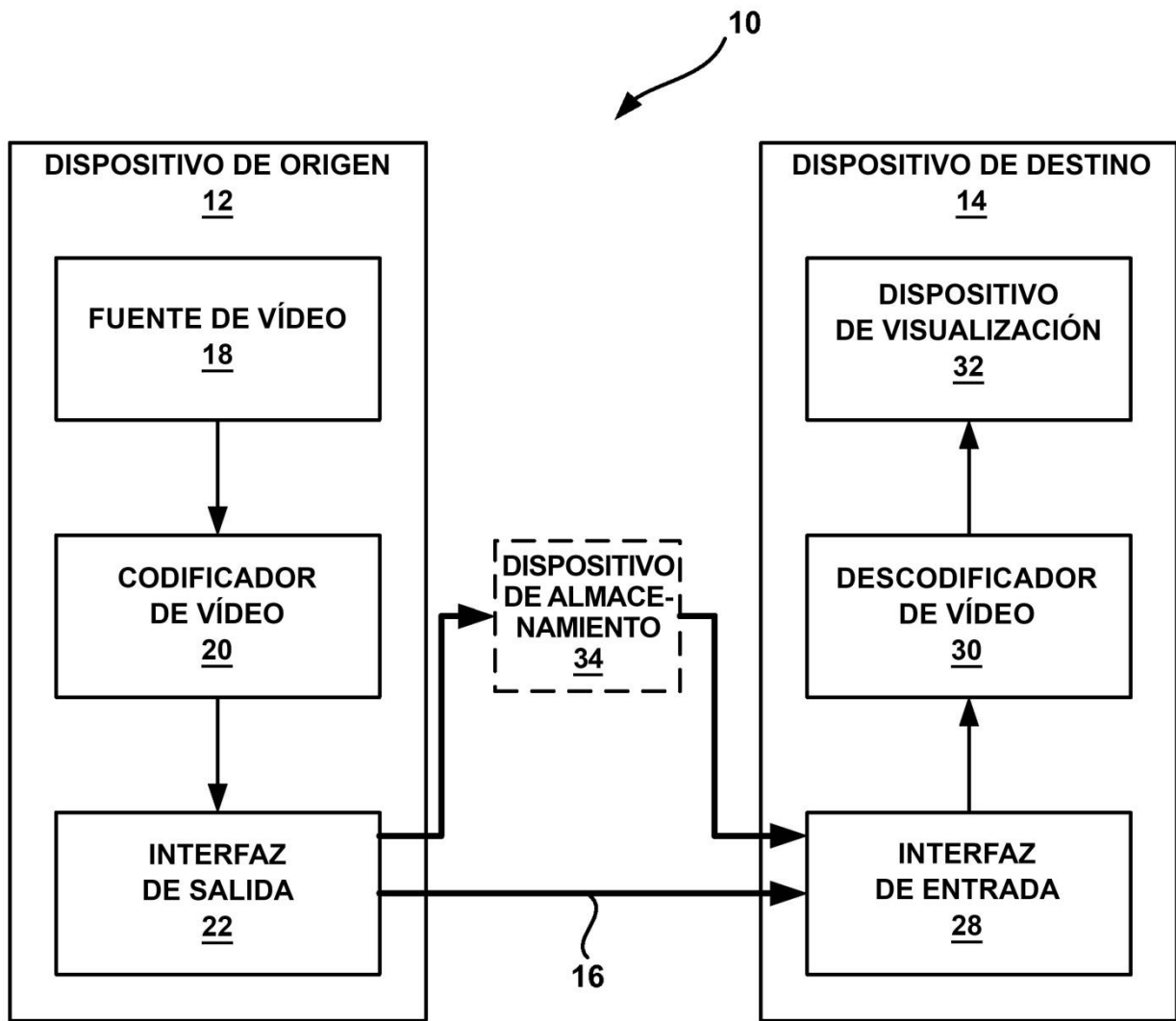
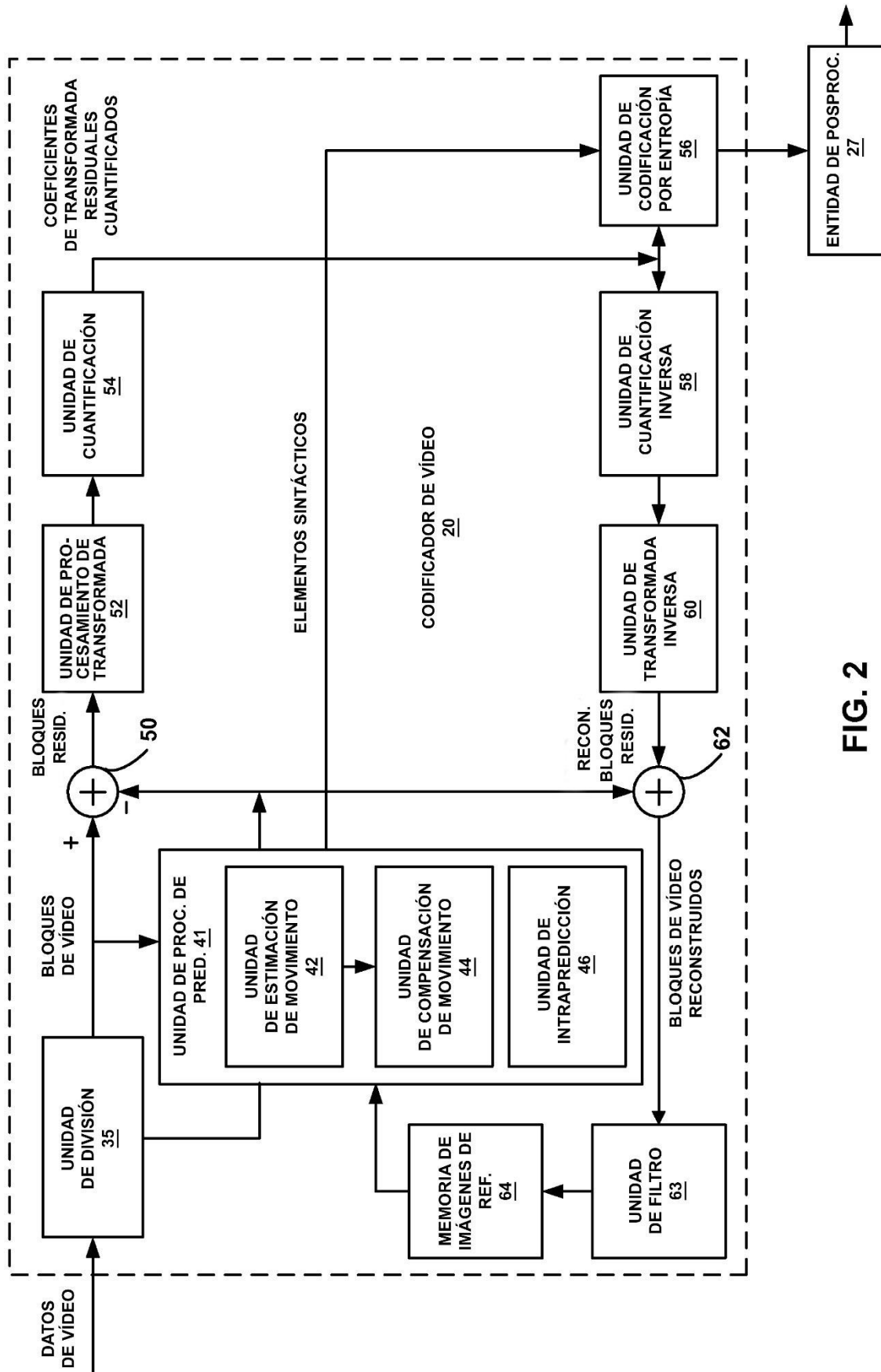


FIG. 1



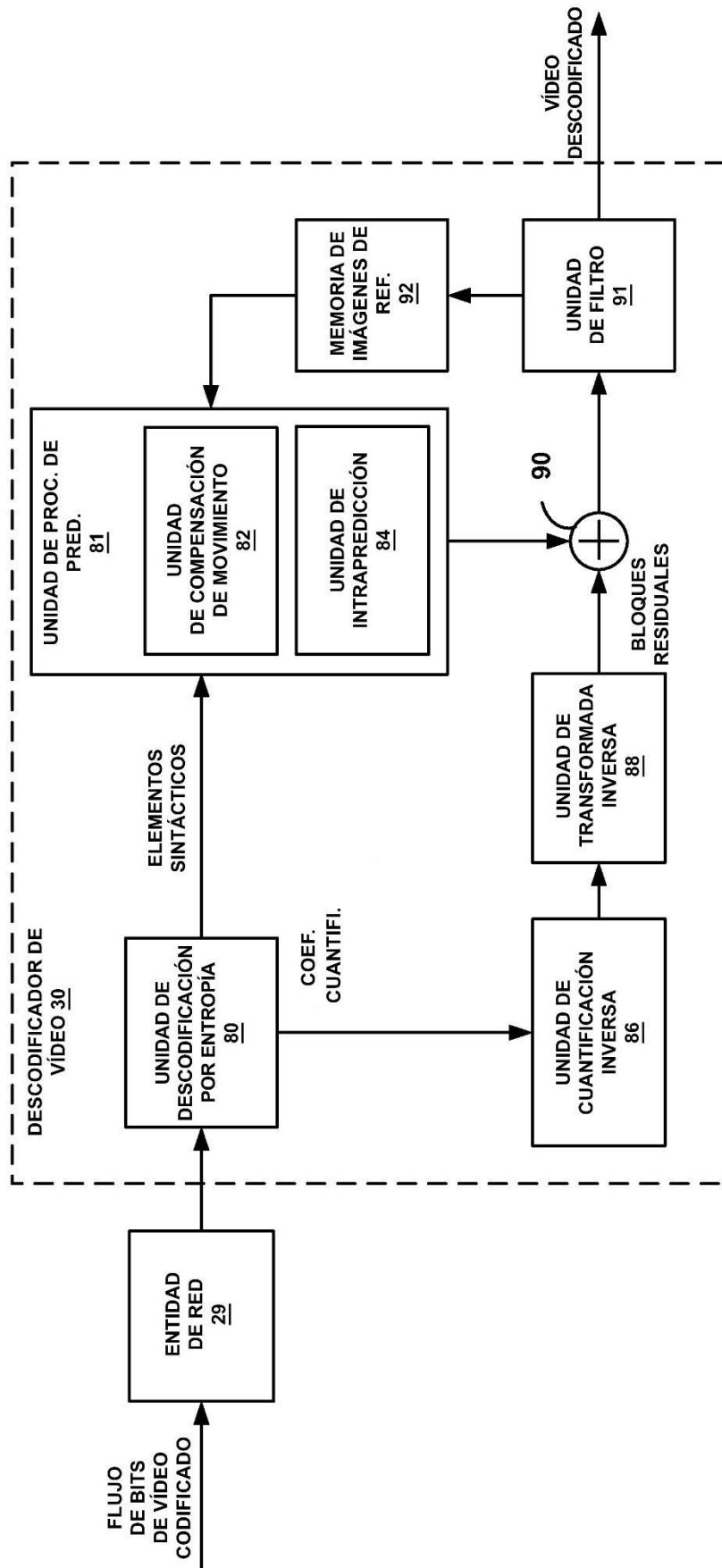


FIG. 3

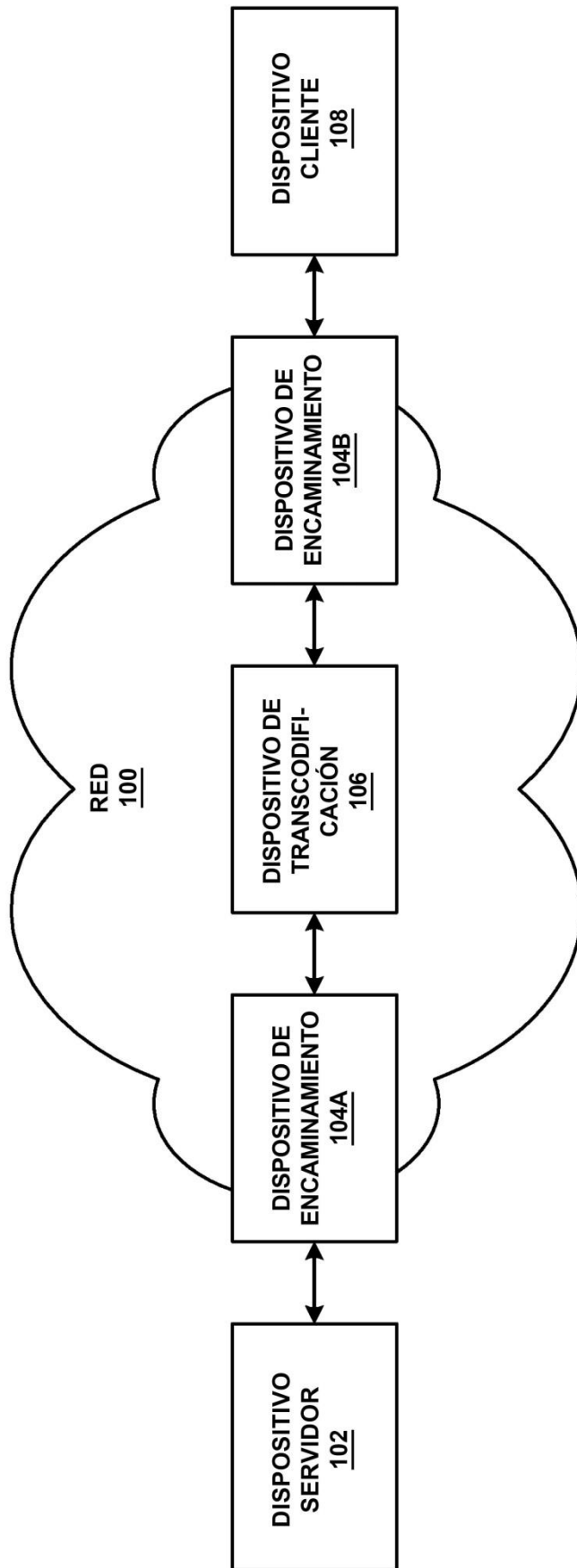


FIG. 4

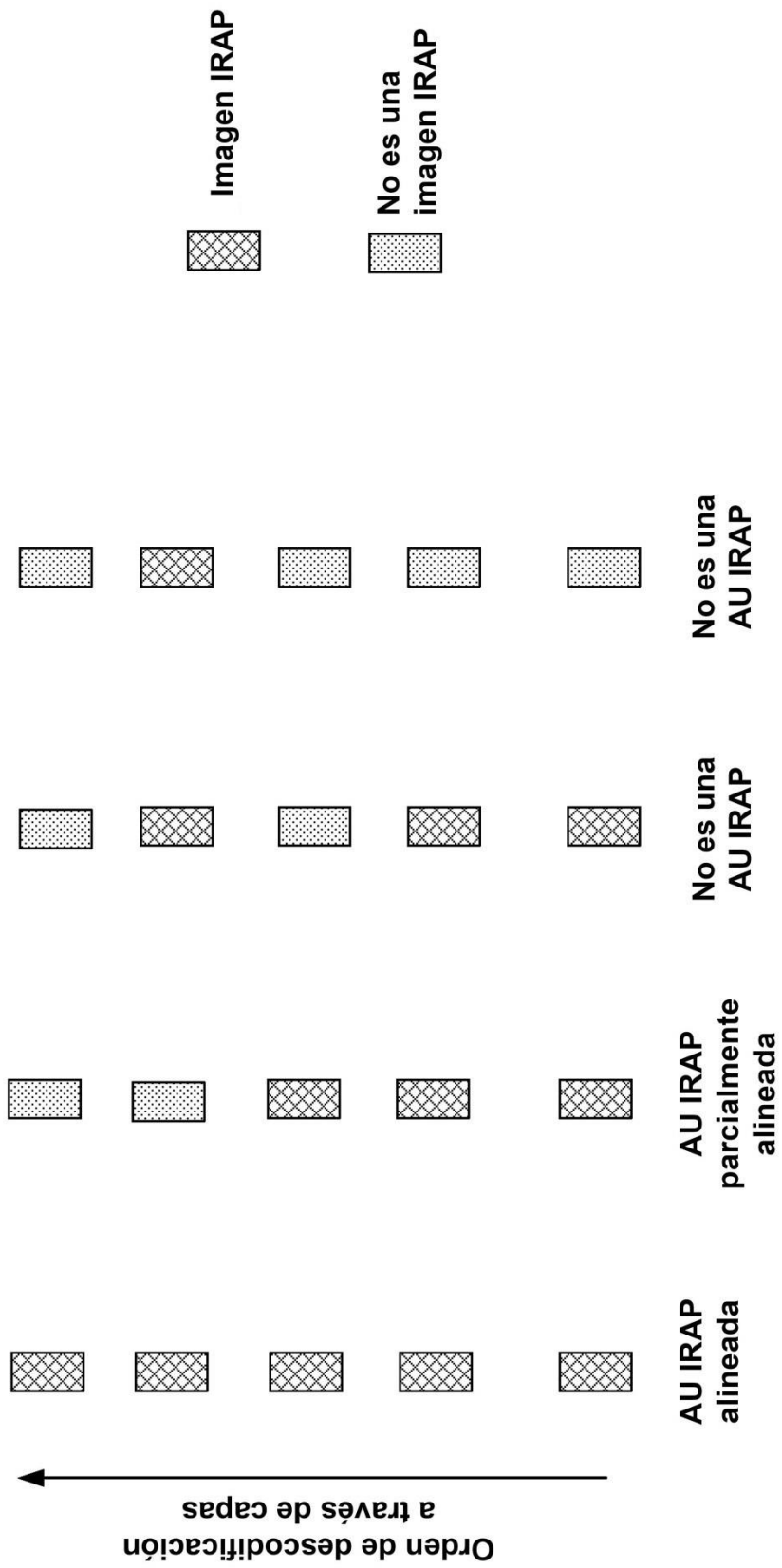


FIG. 5

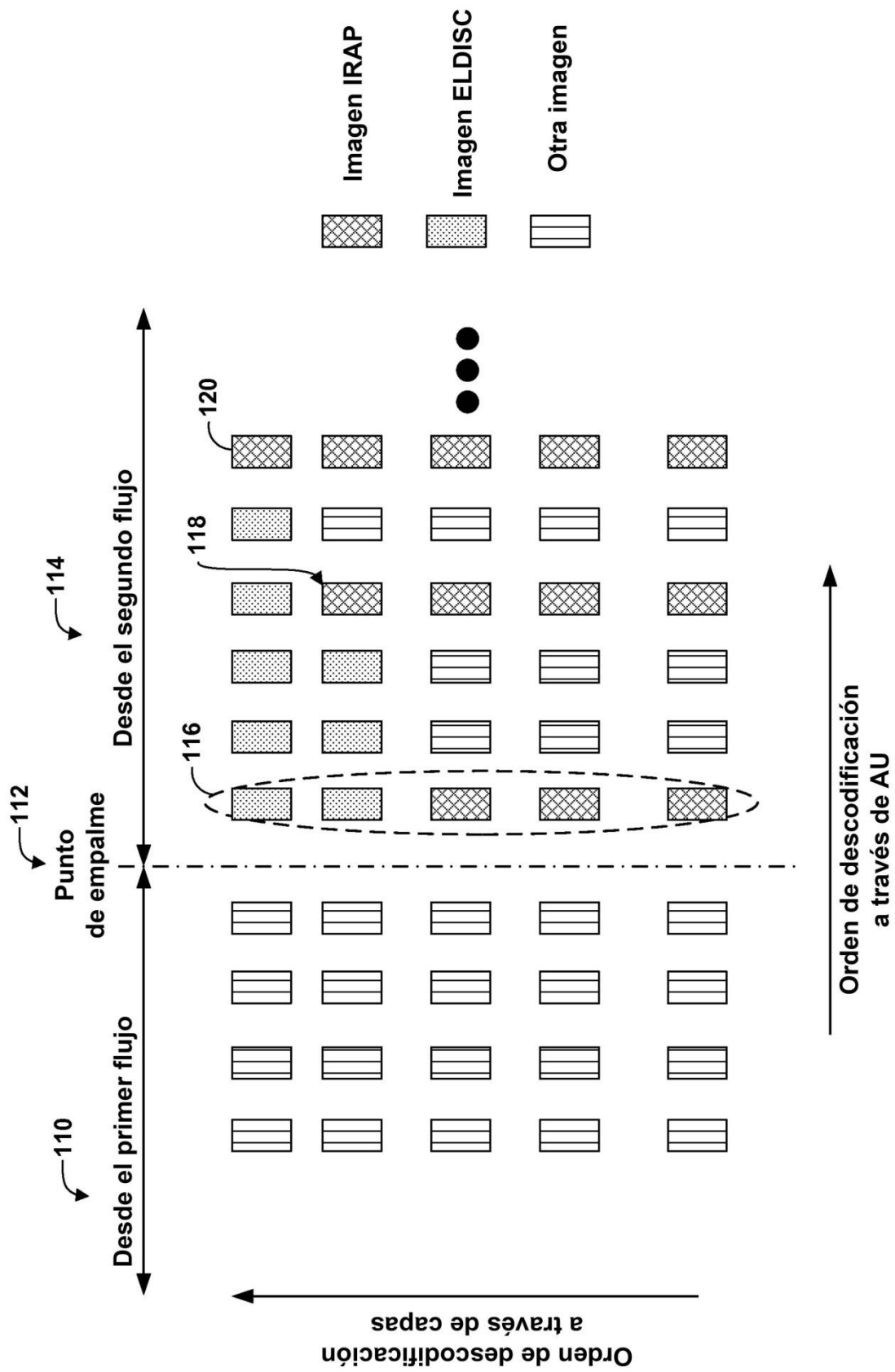


FIG. 6

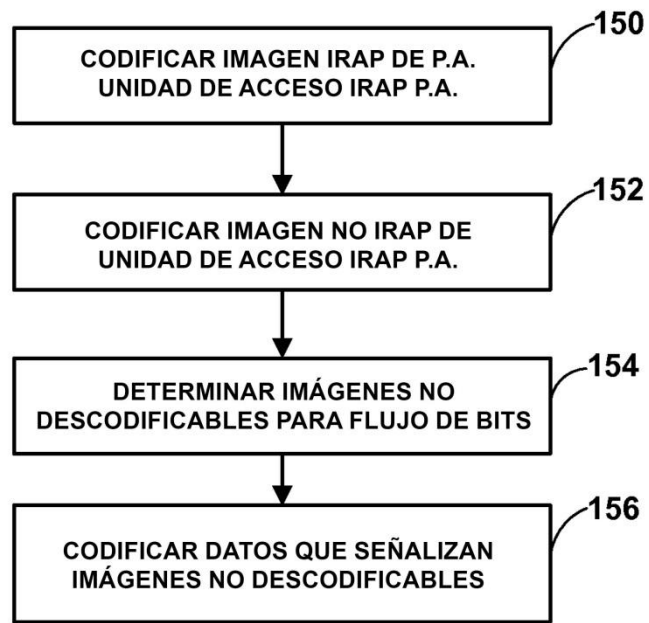


FIG. 7

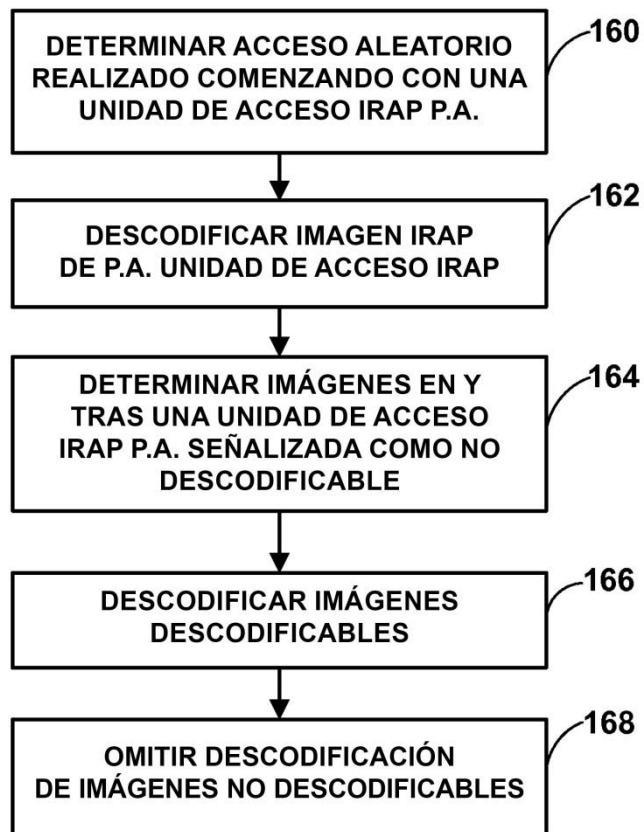


FIG. 8