

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 705 760**

51 Int. Cl.:

H04N 19/593 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **30.06.2015 PCT/US2015/038629**

87 Fecha y número de publicación internacional: **07.01.2016 WO16004086**

96 Fecha de presentación y número de la solicitud europea: **30.06.2015 E 15736769 (9)**

97 Fecha y número de publicación de la concesión europea: **17.10.2018 EP 3164993**

54 Título: **Procedimiento para la codificación en modo de paleta**

30 Prioridad:

02.07.2014 US 201462020340 P

23.07.2014 US 201462028039 P

29.06.2015 US 201514754577

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

26.03.2019

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

ATTN: International IP Administration, 5775

Morehouse Drive

San Diego, CA 92121-1714, US

72 Inventor/es:

PU, WEI;

JOSHI, RAJAN LAXMAN;

CHEN, JIANLE;

KARCZEWICZ, MARTA;

HSIEH, CHENG-TEH;

ZOU, FENG y

SOLE ROJALS, JOEL

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 705 760 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

Procedimiento para la codificación en modo de paleta

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere al campo de la codificación y compresión de vídeo y, en particular, a la codificación de contenido de pantalla.

10 **ANTECEDENTES**

[0002] Las capacidades de vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de sobremesa, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por el Grupo de expertos en imágenes en movimiento 2 (MPEG-2), MPEG-4, el Sector de normalización de las telecomunicaciones de la unión internacional de telégrafos (ITU-T) H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC) y las extensiones de tales normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando tales técnicas de codificación de vídeo.

[0003] Con la prevalencia del acceso a Internet de alta velocidad, las aplicaciones de vídeo emergentes como el uso compartido de escritorio remoto, la infraestructura de escritorio virtual y la pantalla inalámbrica requieren una alta eficiencia de compresión de los contenidos de la pantalla. Sin embargo, las herramientas adicionales de codificación intra e intervdeo fueron diseñadas principalmente para contenidos naturales. Los contenidos de la pantalla tienen características significativamente diferentes en comparación con los contenidos naturales (por ejemplo, bordes afilados y menos o ningún ruido), lo que hace que esas herramientas de codificación tradicionales sean insuficientes. XP030115927 XIU X y otros: "Description of screen content coding technology proposal by InterDigital [Descripción de la tecnología de codificación de contenido de pantalla propuesta por InterDigital]", 17. REUNIÓN JCT-VC; 27-3-2014 - 4-4-2014; VALENCIA; (EQUIPO COLABORATIVO CONJUNTO SOBRE LA CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 y ITU-T SG.16); URL: HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/, número JCTVC-Q0037, 18 de marzo de 2014 (2014-03-18), describe el desarrollo de tecnologías de codificación dedicadas a las características del contenido de la pantalla para futuras extensiones de HEVC para la codificación del contenido de la pantalla. Describe dos tecnologías principales, a saber, la codificación de paleta mejorada y la conversión del espacio de color de residuos adaptable.

40 **SUMARIO**

[0004] Cada uno de los sistemas, procedimientos y dispositivos de esta divulgación tiene varios aspectos innovadores, ninguno de los cuales es el único responsable de los atributos deseables divulgados en el presente documento.

[0005] En un aspecto, un procedimiento para descodificar datos de vídeo en un flujo de bits, donde el flujo de bits incluye una unidad de codificación (CU) codificada en el modo de paleta, incluye: analizar una paleta asociada con la CU proporcionada en el flujo de bits, incluyendo la paleta una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice; analizar una o más longitudes de ejecución proporcionadas en el flujo de bits que están asociadas con la CU, indicando cada longitud de ejecución un número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba; analizar uno o más valores de índice proporcionados en el flujo de bits asociado con la CU, indicando cada valor de índice un valor de píxel en la paleta que está asociado con la posición actual en la CU; analizar uno o más valores de píxel de escape proporcionados en el flujo de bits que están asociados con la CU, indicando cada valor de píxel de escape un valor de píxel que no está en la paleta, en el que los valores de píxel de escape se analizan desde posiciones consecutivas en el flujo de bits, estando las posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU; y descodificar la CU en base a la paleta analizada, las longitudes de ejecución analizadas, los valores de índice analizados y los valores de escape analizados.

[0006] En otro aspecto, un aparato para descodificar datos de vídeo proporcionados en un flujo de bits incluye una memoria y un procesador en comunicación con la memoria. La memoria está configurada para almacenar datos de vídeo asociados con el flujo de bits, incluyendo el flujo de bits una unidad de codificación (CU) codificada en el modo de paleta. El procesador está configurado para: analizar una paleta asociada con la CU proporcionada en el flujo de bits, incluyendo la paleta una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice; analizar una o más longitudes de ejecución

proporcionadas en el flujo de bits que están asociadas con la CU, indicando cada longitud de ejecución un número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba; analizar uno o más valores de índice proporcionados en el flujo de bits que están asociados con la CU, indicando cada valor de índice un valor de píxel en la paleta que está asociado con la posición actual en la CU; analizar uno o más valores de píxel de escape proporcionados en el flujo de bits que están asociados con la CU, indicando cada valor de píxel de escape un valor de píxel que no está en la paleta, en el que los valores de píxel de escape se analizan desde posiciones consecutivas en el flujo de bits, estando las posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU; y descodificar la CU en función de la paleta analizada, las longitudes de ejecución analizadas, los valores de índice analizados y los valores de escape analizados.

[0007] En un aspecto, un procedimiento para codificar datos de vídeo en un flujo de bits incluye: analizar una pluralidad de píxeles en una unidad de codificación (CU), teniendo cada píxel un valor de píxel asociado con el mismo; generar una paleta basada en la pluralidad de píxeles en la CU, incluyendo la paleta una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice; determinar una o más longitudes de ejecución asociadas con la CU en el flujo de bits, indicando cada longitud de ejecución un número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba; determinar uno o más valores de índice asociados con la CU en el flujo de bits, indicando cada valor de índice un valor de píxel en la paleta que está asociado con la posición actual en la CU; determinar uno o más valores de píxel de escape asociados con la CU en el flujo de bits, indicando cada valor de píxel de escape un valor de píxel que no está en la paleta; y codificar la CU en base a la paleta generada, las longitudes de ejecución determinadas, los valores de índice determinados y los valores de píxeles de escape determinados, en donde los valores de píxeles de escape se codifican en posiciones consecutivas en el flujo de bits, estando las posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU.

[0008] En otro aspecto, un aparato para codificar datos de vídeo en un flujo de bits incluye una memoria y un procesador en comunicación con la memoria. La memoria está configurada para almacenar datos de vídeo asociados con el flujo de bits, incluyendo el flujo de bits una unidad de codificación (CU) codificada en el modo de paleta. El procesador está configurado para: analizar una pluralidad de píxeles en una unidad de codificación (CU), teniendo cada píxel un valor de píxel asociado con el mismo; generar una paleta basada en la pluralidad de píxeles en la CU, incluyendo la paleta una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice; determinar una o más longitudes de ejecución asociadas con la CU en el flujo de bits, indicando cada longitud de ejecución un número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba; determinar uno o más valores de índice asociados con la CU en el flujo de bits; indicando cada valor de índice un valor de píxel en la paleta que está asociada con la posición actual en la CU; determinar uno o más valores de píxel de escape asociados con la CU en el flujo de bits; indicando cada valor de píxel de escape un valor de píxel que no está en la paleta; y codificar la CU en base a la paleta generada, las longitudes de ejecución determinadas, los valores de índice determinados y los valores de píxel de escape determinados, en donde los valores de píxel de escape se codifican en posiciones consecutivas en el flujo de bits, estando las posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0009]

La figura 1A es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede utilizar técnicas según los aspectos descritos en esta divulgación.

La figura 1B es un diagrama de bloques que ilustra otro sistema de codificación y descodificación de vídeo de ejemplo que puede llevar a cabo técnicas según los aspectos descritos en esta divulgación.

La figura 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas según aspectos descritos en esta divulgación.

La figura 3 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas según aspectos descritos en esta divulgación.

La figura 4 es un diagrama de bloques que ilustra una CU de entrada, un bloque de índice, un píxel de escape y una paleta asociada con la CU.

La figura 5 es un diagrama de flujo que ilustra un procedimiento para la codificación de datos de vídeo en un flujo de bits de acuerdo con aspectos descritos en esta divulgación.

La figura 6 es un diagrama de flujo que ilustra un procedimiento para la descodificación de datos de vídeo en un flujo de bits de acuerdo con aspectos descritos en esta divulgación.

5 La figura 7 es un diagrama de flujo que ilustra otro procedimiento para descodificar datos de vídeo en un flujo de bits de acuerdo con aspectos descritos en esta divulgación.

La figura 8 es un diagrama de flujo que ilustra otro procedimiento para codificar datos de vídeo en un flujo de bits de acuerdo con aspectos descritos en esta divulgación.

10 La figura 9 es un diagrama de flujo que ilustra un procedimiento para la codificación de vídeo en un flujo de bits de acuerdo con aspectos descritos en esta divulgación.

DESCRIPCIÓN DETALLADA

15 **[0010]** En implementaciones existentes de codificación de contenido de pantalla, puede haber algunas redundancias en el flujo de bits. Estas redundancias se pueden eliminar al omitir cierta señalización del elemento sintáctico cuando se cumplen ciertas condiciones. Además, algunos elementos sintácticos pueden introducir la dependencia del análisis sintáctico. Por ejemplo, un elemento sintáctico para indicar el modo de ejecución puede no necesitar señalización si el píxel actual está en la primera línea del bloque, ya que el descodificador puede inferir que el modo de ejecución es el modo de copia de índice (por ejemplo, modo de copia de la izquierda). Además, en un caso donde el descodificador descodifica el valor del índice primero, y dependiendo del valor del índice descodificado, el descodificador decide si el modo es el modo de copia del índice o el modo de escape (por ejemplo, basado en si el valor del índice representa o no un valor de índice de escape). Si el descodificador determina que el modo es el modo de copia de índice, el analizador del descodificador continúa analizando la longitud de ejecución. Si el descodificador determina que el modo es el modo de escape, el analizador del descodificador puede continuar analizando los valores de escape y/o la longitud de ejecución. Como los analizadores generalmente operan a una velocidad mucho mayor que los descodificadores, tal dependencia entre el motor de descodificación y el motor de análisis puede afectar el rendimiento del analizador (por ejemplo, dado que el motor de análisis puede necesitar esperar a que el motor de descodificación descodifique los bits analizados). Por lo tanto, se desea un procedimiento mejorado de procesamiento de bloques codificados en el modo de codificación de paleta. En esta solicitud, se describen varios procedimientos novedosos para organizar los elementos de la paleta en el flujo de bits para evitar o reducir la dependencia del análisis en el modo de paleta.

35 **[0011]** En la descripción a continuación, se describen técnicas de codificación avanzada de vídeo (AVC)/H.264 relacionadas con ciertos modos de realización; también se analizan la norma HEVC y las técnicas relacionadas. Aunque ciertos modos de realización se describen en el presente documento en el contexto de las normas HEVC y/o H.264, un experto en la materia puede apreciar que los sistemas y procedimientos divulgados en el presente documento pueden aplicarse a cualquier norma de codificación de vídeo adecuada. Por ejemplo, los modos de realización divulgados en el presente documento pueden ser aplicables a una o más de las siguientes normas: H.261 del Sector de normalización de las telecomunicaciones (ITU-T) de la Unión internacional de telecomunicaciones (ITU), MPEG-1 Visual de la Organización Internacional de Normalización/Comisión Electrotécnica Internacional (ISO/IEC), 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), incluidas las ampliaciones escalables.

45 **[0012]** HEVC, en general, sigue el marco de las normas de codificación de vídeo anteriores en muchos aspectos. La unidad de predicción en HEVC es diferente de las unidades de predicción (por ejemplo, macrobloques) en ciertas normas anteriores de codificación de vídeo. De hecho, el concepto de un macrobloque no existe en HEVC como se entiende en ciertas normas anteriores de codificación de vídeo. Un macrobloque se reemplaza por una estructura jerárquica basada en un esquema de árbol cuaternario, que puede proporcionar una gran flexibilidad, entre otros beneficios posibles. Por ejemplo, dentro del esquema HEVC, se definen tres tipos de bloques, unidad de codificación (CU), unidad de predicción (PU) y unidad de transformación (TU). La CU puede referirse a la unidad básica de división de la región. La CU puede considerarse análoga al concepto de macrobloque, pero HEVC no limita el tamaño máximo de la CU y puede permitir la división recursiva en cuatro CU de igual tamaño para mejorar la adaptabilidad del contenido. La PU puede considerarse la unidad básica de inter/intrapredicción, y una única PU puede contener múltiples particiones de forma arbitraria para codificar de manera efectiva patrones de imagen irregulares. La TU se puede considerar como la unidad básica de transformación. La PU se puede definir independientemente de la PU; sin embargo, el tamaño de una TU puede estar limitado al tamaño de la CU a la que pertenece la TU. Esta separación de la estructura de bloques en tres conceptos diferentes puede permitir optimizar cada unidad de acuerdo con su función respectiva de la unidad, lo que puede resultar en una mejora de la eficacia de codificación.

Normas de codificación de vídeo

65 **[0013]** Una imagen digital, tal como una imagen de vídeo, una imagen de televisión, una imagen fija o una imagen generada por una grabadora de vídeo o un ordenador, puede consistir en píxeles o muestras dispuestas en líneas

horizontales y verticales. La cantidad de píxeles en una sola imagen suele ser de decenas de miles. Cada píxel contiene habitualmente información de luminancia y crominancia. Sin compresión, la mera cantidad de información que se transmitirá desde un codificador de imágenes a un descodificador de imágenes haría nada práctica la transmisión de imágenes en tiempo real. Para reducir la cantidad de información a transmitir, se han desarrollado una serie de procedimientos de compresión diferentes, tales como las normas JPEG, MPEG y H.263.

[0014] Las normas de codificación de vídeo incluyen ITU-T H.261, MPEG-1 Visual de la ISO/IEC, ITU-T H.262 o MPEG-2 Visual de la ISO/IEC, ITU-T H.263, MPEG-4 Visual de la ISO/IEC e ITU-T H.264 (también conocida como AVC del MPEG-4 de la ISO/IEC) y HEVC, incluidas las ampliaciones escalables.

[0015] Además, se ha desarrollado una norma de codificación de vídeo, denominada HEVC, por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y MPEG de ISO/IEC. La cita completa del borrador 10 de HEVC es el documento JCTVC-L1003, de Bross y otros, titulado "High Efficiency Video Coding (HEVC) Text Specification Draft 10 [Especificación textual de la Codificación de Vídeo de Alta Eficacia (HEVC), Borrador 10]", Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª reunión: Ginebra, Suiza, del 14 de enero de 2013 al 23 de enero de 2013. La ampliación escalable a HEVC también está siendo desarrollada por el JCT-VC.

Sistema de codificación de vídeo

[0016] En lo sucesivo se describen de forma más detallada diversos aspectos de los sistemas, aparatos y procedimientos novedosos, con referencia a los dibujos adjuntos. Sin embargo, esta divulgación se puede integrar de muchas formas diferentes y no se debería interpretar que está limitada a cualquier estructura o función específica presentada a lo largo de esta divulgación. En cambio, estos aspectos se proporcionan para que esta divulgación sea exhaustiva y completa, y transmita por completo el alcance de la divulgación a los expertos en la materia. Basándose en las enseñanzas del presente documento, un experto en la materia debería apreciar que el alcance de la divulgación pretende abarcar cualquier aspecto de los sistemas, aparatos y procedimientos novedosos divulgados en el presente documento, ya sean implementados de forma independiente de, o en combinación con, cualquier otro aspecto de la presente divulgación. Por ejemplo, un aparato se puede implementar o un procedimiento se puede llevar a la práctica usando cualquier número de los aspectos expuestos en el presente documento. Además, el alcance de la presente divulgación pretende abarcar dicho aparato o procedimiento que se lleva a la práctica usando otra estructura, funcionalidad o estructura y funcionalidad, además, o aparte, de los diversos aspectos de la presente divulgación expuestos en el presente documento. Debería entenderse que cualquier aspecto divulgado en el presente documento puede realizarse mediante uno o más elementos de una reivindicación.

[0017] Aunque en el presente documento se describen aspectos particulares, muchas variantes y permutaciones de estos aspectos están dentro del alcance de la divulgación. Aunque se mencionan algunos beneficios y ventajas de los aspectos preferentes, el alcance de la divulgación no está concebido para limitarse a beneficios, usos u objetivos particulares. En cambio, los aspectos de la divulgación pretenden ser ampliamente aplicables a diferentes tecnologías inalámbricas, configuraciones de sistema, redes y protocolos de transmisión, algunos de los cuales se ilustran a modo de ejemplo en las figuras y en la siguiente descripción de los aspectos preferentes. La descripción detallada y los dibujos son meramente ilustrativos de la divulgación en lugar de ser limitativos, definiéndose el alcance de la divulgación mediante las reivindicaciones adjuntas y equivalentes de las mismas.

[0018] Los dibujos adjuntos ilustran ejemplos. Los elementos indicados mediante números de referencia en los dibujos adjuntos corresponden a elementos indicados mediante números de referencia similares en la siguiente descripción. En esta divulgación, los elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero", etc.) no necesariamente implican que los elementos tienen un orden particular. Más bien, dichas palabras ordinales se usan simplemente para referirse a diferentes elementos de un mismo tipo o un tipo similar.

[0019] La **figura 1A** es un diagrama de bloques que ilustra un sistema de codificación de vídeo 10 de ejemplo que puede utilizar técnicas de acuerdo con aspectos descritos en esta divulgación. Como se usa en el presente documento, el término "codificador de vídeo" o "codificador" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse genéricamente a la codificación de vídeo y a la descodificación de vídeo. Además de los codificadores de vídeo y los descodificadores de vídeo, los aspectos descritos en la presente solicitud pueden extenderse a otros dispositivos relacionados, tales como los transcodificadores (por ejemplo, dispositivos que pueden descodificar un flujo de bits y recodificar otro flujo de bits) y cajas intermedias (por ejemplo, dispositivos que pueden modificar, transformar y/o manipular de otro modo un flujo de bits).

[0020] Como se muestra en la **figura 1A**, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados para ser descodificados posteriormente por un dispositivo de destino 14. En el ejemplo de la **figura 1A**, el dispositivo de origen 12 y el dispositivo de destino 14 constituyen dispositivos

independientes. Sin embargo, se observa que el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar en el mismo dispositivo, o ser parte del mismo, como se muestra en el ejemplo de la **figura 1B**.

[0021] Con referencia una vez más a la **figura 1A**, el dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender, respectivamente, cualquiera entre una amplia gama de dispositivos, incluidos los ordenadores de escritorio, los ordenadores plegables (por ejemplo, los ordenadores portátiles), los ordenadores de tableta, los descodificadores, los equipos telefónicos tales como los llamados "teléfonos inteligentes", los llamados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de flujo de vídeo o similares. En algunos modos de realización, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0022] El dispositivo de destino 14 puede recibir, a través del enlace 16, los datos de vídeo codificados a descodificar. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En el ejemplo de la **figura 1A**, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse 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, inalámbrica o cableada, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0023] De forma alternativa, los datos codificados pueden ser emitidos desde una interfaz de salida 22 a un dispositivo de almacenamiento 31 (opcionalmente presente). De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 31 mediante la interfaz de entrada 28, por ejemplo, del dispositivo de destino 14. El dispositivo de almacenamiento 31 puede incluir cualquiera de una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco duro, memoria flash, memoria volátil o no volátil u otros medios cualesquiera de almacenamiento digital adecuados para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 31 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que pueda retener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento 31, mediante transmisión o descarga. El servidor de ficheros 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 ejemplos de servidores de archivos incluyen un servidor de la red (por ejemplo, para una página de la red), un servidor de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento anexos a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados mediante cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión inalámbrica de red de área local (WLAN)), una conexión cableada (por ejemplo, una línea de abonado digital (DSL), un módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 31 puede ser una transmisión en continuo, una transmisión de descarga o una combinación de ambas.

[0024] Las técnicas de esta divulgación no están limitadas a aplicaciones o a configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, para admitir cualquiera de una diversidad de aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, a través de Internet (por ejemplo, transmisión adaptativa dinámica sobre protocolo de transferencia de hipertexto (HTTP), etc.), 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 de codificación de vídeo 10 puede configurarse para admitir la transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0025] En el ejemplo de la **figura 1A**, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20 y la 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, el origen de vídeo 18 puede incluir un origen tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación 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 dichos orígenes. Como un ejemplo, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados "teléfonos de cámara" o "videoteléfonos", tal como se ilustra en el ejemplo de la **figura 1B**. Sin embargo, las

técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas.

5 [0026] El vídeo capturado, precapturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. Los datos de vídeo codificados pueden ser transmitidos al dispositivo de destino 14 mediante la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados pueden almacenarse también (o de forma alternativa) en el dispositivo de almacenamiento 31 para un acceso posterior por parte del dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción. El codificador de vídeo 20 ilustrado en la **figura 1A** y la **figura 1B** puede comprender el codificador de vídeo 20 ilustrado en la **figura 2** o cualquier otro codificador de vídeo descrito en el presente documento.

15 [0027] En el ejemplo de la **figura 1A**, el dispositivo de destino 14 incluye la 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 puede recibir los datos de vídeo codificados por el enlace 16 y/o desde el dispositivo de almacenamiento 31. Los datos de vídeo codificados, comunicados por el enlace 16, o proporcionados en el dispositivo de almacenamiento 31, 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 pueden incluirse con los datos de vídeo codificados, transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de ficheros. El codificador de vídeo 30 ilustrado en la **figura 1A** y la **figura 1B** puede comprender el codificador de vídeo 30 ilustrado en la **figura 3** o cualquier otro codificador de vídeo descrito en el presente documento.

25 [0028] El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para 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 entre varios 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.

35 [0029] En aspectos relacionados, la **figura 1B** muestra un sistema de codificación de vídeo 10' de ejemplo en el que el dispositivo de origen 12 y el dispositivo de destino 14 están en, o forman parte de, un dispositivo 11. El dispositivo 11 puede ser un equipo telefónico de mano, tal como un teléfono "inteligente" o similar. El dispositivo 11 puede incluir un dispositivo controlador/procesador 13 (optativamente presente) en comunicación operativa con el dispositivo de origen 12 y el dispositivo de destino 14. El sistema de codificación de vídeo 10' de la **figura 1B**, y sus componentes, son similares de otro modo al sistema de codificación de vídeo 10 de la **figura 1A** y sus componentes.

40 [0030] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como HEVC, y pueden ajustarse a un modelo de prueba HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o de la industria, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, Parte 10, AVC, o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263.

50 [0031] Aunque no se muestra en los ejemplos de la **figura 1A** y la **figura 1B**, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados, cada uno, con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o flujos de datos por separado. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ser conformes al protocolo de multiplexador ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

55 [0032] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse, cada uno, como cualquiera 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), matrices 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 mediante uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

Proceso de codificación de vídeo

- 5 **[0033]** Como se ha mencionado brevemente en lo que antecede, el codificador de vídeo 20 codifica datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes es una imagen fija que forma parte de un vídeo. En algunos casos, una imagen puede denominarse una "trama" de vídeo. Cuando el codificador de vídeo 20 codifica los datos de vídeo, el codificador de vídeo 20 puede generar un flujo de bits. El flujo de bits puede incluir una secuencia de bits que forman una representación codificada de los datos de vídeo. El flujo de bits puede incluir imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen.
- 10 **[0034]** Para generar el flujo de bits, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada imagen en los datos de vídeo. Cuando el codificador de vídeo 20 realiza operaciones de codificación en las imágenes, el codificador de vídeo 20 puede generar una serie de imágenes codificadas y datos asociados. Los datos asociados pueden incluir conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS), conjuntos de parámetros de adaptación (APS) y otras estructuras sintácticas. Un SPS puede contener parámetros aplicables a cero o más secuencias de imágenes. Un PPS puede contener parámetros aplicables a cero o más imágenes. Un APS puede contener parámetros aplicables a cero o más imágenes. Los parámetros en un APS pueden ser parámetros que tienen más probabilidades de cambiar que los parámetros en el PPS.
- 15 **[0035]** Para generar una imagen codificada, el codificador de vídeo 20 puede dividir una imagen en bloques de vídeo de igual tamaño. Un bloque de vídeo puede ser una formación bidimensional de muestras. Cada uno de los bloques de vídeo está asociado con un bloque de árbol. En algunos casos, un bloque de árbol también puede denominarse una unidad de codificación más grande (LCU). Los bloques de árbol de HEVC pueden ser aproximadamente análogos a los macrobloques de normas anteriores, tales como la H.264/AVC. Sin embargo, un bloque de árbol no está limitado necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). El codificador de vídeo 20 puede usar la división en árbol cuaternario para dividir los bloques de vídeo de bloques de árbol en bloques de vídeo asociados con las CU, de ahí el nombre "bloques de árbol".
- 20 **[0036]** En algunos ejemplos, el codificador de vídeo 20 puede dividir una imagen en una pluralidad de fragmentos. Cada uno de los fragmentos puede incluir un número entero de CU. En algunos casos, un fragmento comprende un número entero de bloques de árbol. En otros casos, un límite de un fragmento puede estar dentro de un bloque de árbol.
- 25 **[0037]** Como parte de la realización de una operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada fragmento de la imagen. Cuando el codificador de vídeo 20 realiza una operación de codificación en un fragmento, el codificador de vídeo 20 puede generar datos codificados asociados con el fragmento. Los datos codificados asociados con el fragmento pueden denominarse un "fragmento codificado".
- 30 **[0038]** Para generar un fragmento codificado, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada bloque de árbol en un fragmento. Cuando el codificador de vídeo 20 realiza una operación de codificación en un bloque de árbol, el codificador de vídeo 20 puede generar un bloque de árbol codificado. El bloque de árbol codificado puede comprender datos que representan una versión codificada del bloque de árbol.
- 35 **[0039]** Cuando el codificador de vídeo 20 genera un fragmento codificado, el codificador de vídeo 20 puede realizar operaciones de codificación (por ejemplo, codificar) en los bloques de árbol en el fragmento de acuerdo con una orden de escaneo de trama. Por ejemplo, el codificador de vídeo 20 puede codificar los bloques de árbol del fragmento en un orden que avanza de izquierda a derecha en la fila más alta de los bloques de árbol en el fragmento, luego de izquierda a derecha en la siguiente fila inferior de los bloques de árbol, y así sucesivamente hasta que el codificador de vídeo 20 ha codificado cada uno de los bloques de árbol del fragmento.
- 40 **[0040]** Como resultado de codificar los bloques de árbol de acuerdo con el orden de escaneo de trama, los bloques de árbol de arriba y a la izquierda de un bloque de árbol determinado pueden haber sido codificados, pero los bloques de árbol de abajo y a la derecha del bloque de árbol determinado aún no han sido codificados. En consecuencia, el codificador de vídeo 20 puede tener acceso a la información generada por la codificación de los bloques de árbol de arriba y a la izquierda del bloque de árbol determinado cuando codifica el bloque de árbol determinado. Sin embargo, el codificador de vídeo 20 puede no ser capaz de acceder a la información generada por la codificación de los bloques de árbol de abajo y a la derecha del bloque de árbol determinado al codificar el bloque de árbol determinado.
- 45 **[0041]** Para generar un bloque de árbol codificado, el codificador de vídeo 20 puede llevar a cabo de forma recursiva la división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado con una CU diferente. Por ejemplo, el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, etc. Una CU dividida puede ser una CU cuyo bloque de vídeo está dividido en
- 50
- 55
- 60
- 65

bloques de vídeo asociados con otras CU. Una CU no dividida puede ser una CU cuyo bloque de vídeo no esté dividido en bloques de vídeo asociados con otras CU.

5 **[0042]** Uno o más elementos sintácticos en el flujo de bits pueden indicar un número máximo de veces que el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol. Un bloque de vídeo de una CU puede tener forma cuadrada. El tamaño del bloque de vídeo de una CU (es decir, el tamaño de la CU) puede variar desde 8x8 píxeles hasta el tamaño de un bloque de vídeo de un bloque de árbol (es decir, el tamaño del bloque de árbol) con un máximo de 64x64 píxeles o mayor.

10 **[0043]** El codificador de vídeo 20 puede realizar operaciones de codificación (por ejemplo, codificar) en cada CU de un bloque de árbol de acuerdo con un orden de escaneo z. En otras palabras, el codificador de vídeo 20 puede codificar una CU superior izquierda, una CU superior derecha, una CU inferior izquierda y luego una CU inferior derecha, en ese orden. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU dividida, el codificador de vídeo 20 puede codificar las CU asociadas con subbloques del bloque de vídeo de la CU dividida de acuerdo con el orden de escaneo z. En otras palabras, el codificador de vídeo 20 puede codificar una CU asociada con un subbloque superior izquierdo, una CU asociada con un subbloque superior derecho, una CU asociada con un subbloque inferior izquierdo, y luego una CU asociada con un subbloque inferior derecho, en ese orden.

20 **[0044]** Como resultado de la codificación de las CU de un bloque de árbol de acuerdo con un orden de escaneo z, las CU arriba, arriba y a la izquierda, arriba y a la derecha, izquierda y abajo y a la izquierda de una CU determinada pueden haber sido codificadas. Las CU de abajo y a la derecha de la CU determinada todavía no han sido codificadas. En consecuencia, el codificador de vídeo 20 puede ser capaz de acceder a la información generada codificando algunas CU que se encuentran próximas a la CU determinada cuando codifica la CU determinada. Sin embargo, el codificador de vídeo 20 puede ser incapaz de acceder a la información generada por la codificación de otras CU que se encuentran junto a la CU determinada cuando codifica la CU determinada.

25 **[0045]** Cuando el codificador de vídeo 20 codifica una CU no dividida, el codificador de vídeo 20 puede generar una o más unidades de predicción (PU) para la CU. Cada una de las PU de la CU puede estar asociada con un bloque de vídeo diferente dentro del bloque de vídeo de la CU. El codificador de vídeo 20 puede generar un bloque de vídeo predicho para cada PU de la CU. El bloque de vídeo predicho de una PU puede ser un bloque de muestras. El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar el bloque de vídeo predicho para una PU.

30 **[0046]** Cuando el codificador de vídeo 20 utiliza la intrapredicción para generar el bloque de vídeo predicho de una PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basándose en muestras descodificadas de la imagen asociada con la PU. Si el codificador de vídeo 20 utiliza la intrapredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU intrapredicha. Si el codificador de vídeo 20 utiliza la interpredicción para generar el bloque de vídeo predicho de la PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basándose en muestras descodificadas de una o más imágenes distintas a la imagen asociada con la PU. Si el codificador de vídeo 20 utiliza la interpredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU interpredicha.

35 **[0047]** Además, cuando el codificador de vídeo 20 utiliza la interpredicción para generar un bloque de vídeo predicho para una PU, el codificador de vídeo 20 puede generar información de movimiento para la PU. La información de movimiento para una PU puede indicar uno o más bloques de referencia de la PU. Cada bloque de referencia de la PU puede ser un bloque de vídeo dentro de una imagen de referencia. La imagen de referencia puede ser una imagen distinta de la imagen asociada con la PU. En algunos casos, un bloque de referencia de una PU también puede denominarse la "muestra de referencia" de la PU. El codificador de vídeo 20 puede generar el bloque de vídeo predicho para la PU en base a los bloques de referencia de la PU.

40 **[0048]** Después de que el codificador de vídeo 20 genere bloques de vídeo predichos para una o más PU de una CU, el codificador de vídeo 20 puede generar datos residuales para la CU basándose en los bloques de vídeo predichos para las PU de la CU. Los datos residuales para la CU pueden indicar diferencias entre muestras en los bloques de vídeo predichos para las PU de la CU y el bloque de vídeo original de la CU.

45 **[0049]** Además, como parte de la realización de una operación de codificación en una CU no dividida, el codificador de vídeo 20 puede realizar la división recursiva en árbol cuaternario en los datos residuales de la CU para dividir los datos residuales de la CU en uno o más bloques de datos residuales (es decir, bloques de vídeo residuales) asociados con las unidades de transformación (TU) de la CU. Cada TU de una CU puede estar asociada con un bloque de vídeo residual diferente.

50 **[0050]** El codificador de vídeo 20 puede aplicar una o más transformaciones a bloques de vídeo residuales asociados con las TU para generar bloques de coeficientes de transformación (por ejemplo, bloques de coeficientes de transformación) asociados con las TU. Conceptualmente, un bloque de coeficientes de transformación puede ser una matriz bidimensional (2D) de coeficientes de transformación.

[0051] Después de generar un bloque de coeficientes de transformación, el codificador de vídeo 20 puede realizar un proceso de cuantificación en el bloque de coeficientes de transformación. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformación se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes de transformación, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o a la totalidad de los coeficientes de transformación. Por ejemplo, un coeficiente de transformación de n bits puede redondearse a la baja hasta un coeficiente de transformación de m bits durante la cuantificación, donde n es mayor que m .

[0052] El codificador de vídeo 20 puede asociar cada CU con un valor de parámetro de cuantificación (QP). El valor QP asociado con una CU puede determinar cómo el codificador de vídeo 20 cuantifica los bloques de coeficientes de transformación asociados con la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados con una CU ajustando el valor QP asociado con la CU.

[0053] Después de que el codificador de vídeo 20 cuantifica un bloque de coeficientes de transformación, el codificador de vídeo 20 puede generar conjuntos de elementos sintácticos que representan los coeficientes de transformación en el bloque de coeficientes de transformación cuantificado. El codificador de vídeo 20 puede aplicar operaciones de codificación por entropía, tales como operaciones de codificación aritmética binaria adaptativa al contexto (CABAC), a algunos de estos elementos sintácticos. También podrían usarse otras técnicas de codificación por entropía, tales como la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación por entropía de división de intervalo de probabilidad (PIPE) u otra codificación aritmética binaria.

[0054] El flujo de bits generado por el codificador de vídeo 20 puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede ser una estructura sintáctica que contiene una indicación de un tipo de datos en la unidad NAL y los bytes que contienen los datos. Por ejemplo, una unidad NAL puede contener datos que representan un conjunto de parámetros de vídeo, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un fragmento codificado, SEI, un delimitador de unidad de acceso, datos de relleno u otro tipo de datos. Los datos en una unidad NAL pueden incluir varias estructuras sintácticas.

[0055] El descodificador de vídeo 30 puede recibir el flujo de bits generado por el codificador de vídeo 20. El flujo de bits puede incluir una representación codificada de los datos de vídeo codificados por el codificador de vídeo 20. Cuando el descodificador de vídeo 30 recibe el flujo de bits, el descodificador de vídeo 30 puede realizar una operación de análisis sintáctico en el flujo de bits. Cuando el descodificador de vídeo 30 realiza la operación de análisis sintáctico, el descodificador de vídeo 30 puede extraer elementos sintácticos del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose en los elementos sintácticos extraídos del flujo de bits. El proceso para reconstruir los datos de vídeo basados en los elementos sintácticos puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20 para generar los elementos sintácticos.

[0056] Después de que el descodificador de vídeo 30 extraiga los elementos sintácticos asociados con una CU, el descodificador de vídeo 30 puede generar bloques de vídeo predichos para las PU de la CU basándose en los elementos sintácticos. Además, el descodificador de vídeo 30 puede cuantificar de forma inversa bloques de coeficientes de transformación asociados con las TU de la CU. El descodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes de transformación para reconstruir los bloques de vídeo residuales asociados con las TU de la CU. Después de generar los bloques de vídeo predichos y reconstruir los bloques de vídeo residuales, el descodificador de vídeo 30 puede reconstruir el bloque de vídeo de la CU basándose en los bloques de vídeo predichos y los bloques de vídeo residuales. De esta manera, el descodificador de vídeo 30 puede reconstruir los bloques de vídeo de las CU basándose en los elementos sintácticos del flujo de bits.

Codificador de vídeo

[0057] La **figura 2** es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación. El codificador de vídeo 20 puede configurarse para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el codificador de vídeo 20 puede configurarse para realizar cualquiera o todas las técnicas de esta divulgación. En algunos ejemplos, las técnicas descritas en esta divulgación pueden compartirse entre los diversos componentes del codificador de vídeo 20. En algunos ejemplos, adicionalmente o de forma alternativa, un procesador (no mostrado) puede configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación.

[0058] Con fines de explicación, esta divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la **figura 2** es para un códec de una sola capa. Sin

embargo, en ciertos modos de realización, algunos o todos los codificadores de vídeo 20 pueden duplicarse para procesar un códec multicapa.

[0059] El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se apoya en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo determinada. 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. La intramodalidad (modalidad I) puede referirse a cualquiera de varias modalidades de codificación de base espacial. Las intermodalidades, tales como la predicción unidireccional (modalidad P) o la predicción bidireccional (modalidad B), pueden referirse a cualquiera de varias modalidades de codificación de base temporal.

[0060] En el ejemplo de la **figura 2**, el codificador de vídeo 20 incluye una pluralidad de componentes funcionales. Los componentes funcionales del codificador de vídeo 20 incluyen una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 113, una memoria intermedia de imágenes descodificadas 114 y una unidad de codificación por entropía 116. La unidad de procesamiento de predicción 100 incluye una unidad de interpredicción 121, una unidad de estimación de movimiento 122, una unidad de compensación de movimiento 124, una unidad de intrapredicción 126 y una unidad de predicción entre capas 128. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales. Además, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden estar altamente integradas, pero están representadas en el ejemplo de la **figura 2** de forma separada con fines explicativos.

[0061] El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede recibir los datos de vídeo de diversas fuentes. Por ejemplo, el codificador de vídeo 20 puede recibir los datos de vídeo de la fuente de vídeo 18 (por ejemplo, mostrada en la **figura 1A** o **1B**) u otra fuente. Los datos de vídeo pueden representar una serie de imágenes. Para codificar los datos de vídeo, el codificador de vídeo 20 puede realizar una operación de codificación en cada una de las imágenes. Como parte de realizar la operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada fragmento de la imagen. Como parte de realizar una operación de codificación en un fragmento, el codificador de vídeo 20 puede realizar operaciones de codificación en bloques de árbol en el fragmento.

[0062] Como parte de realizar una operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede realizar la división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado con una CU diferente. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, etc.

[0063] Los tamaños de los bloques de vídeo asociados con las CU pueden variar desde muestras de 8x8 hasta el tamaño del bloque de árbol, con un máximo de muestras de 64x64 o mayores. En esta divulgación, "NxN" y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de muestras de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, muestras de 16x16 o muestras de 16 por 16. En general, un bloque de vídeo de 16x16 tiene dieciséis muestras en una dirección vertical ($y = 16$) y dieciséis muestras en una dirección horizontal ($x = 16$). Asimismo, un bloque NxN presenta, en general, N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo.

[0064] Además, como parte de la realización de la operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede generar una estructura de datos jerárquica en árbol cuaternario para el bloque de árbol. Por ejemplo, un bloque de árbol puede corresponder a un nodo raíz de la estructura de datos en árbol cuaternario. Si la unidad de procesamiento de predicción 100 divide el bloque de vídeo del bloque de árbol en cuatro subbloques, el nodo raíz tiene cuatro nodos secundarios en la estructura de datos en árbol cuaternario. Cada uno de los nodos secundarios corresponde a una CU asociada con uno de los subbloques. Si la unidad de procesamiento de predicción 100 divide uno de los subbloques en cuatro subsubbloques, el nodo correspondiente a la CU asociada con el subbloque puede tener cuatro nodos secundarios, cada uno de los cuales corresponde a una CU asociada con uno de los subsubbloques.

[0065] Cada nodo de la estructura de datos en árbol cuaternario puede contener datos sintácticos (por ejemplo, elementos sintácticos) para el bloque de árbol o CU correspondiente. Por ejemplo, un nodo en el árbol cuaternario puede incluir un indicador de división, que indica si el bloque de vídeo de la CU correspondiente al nodo está dividido (es decir, partido) en cuatro subbloques. Los elementos sintácticos para una CU pueden definirse de manera recursiva y pueden depender de si el bloque de vídeo de la CU está dividido en subbloques. Una CU cuyo bloque de vídeo no está dividido puede corresponder a un nodo hoja en la estructura de datos en árbol cuaternario. Un bloque de árbol codificado puede incluir datos basados en la estructura de datos en árbol cuaternario para un bloque de árbol correspondiente.

[0066] El codificador de vídeo 20 puede realizar operaciones de codificación en cada CU no dividida de un bloque de árbol. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU no dividida, el codificador de vídeo 20 genera datos que representan una representación codificada de la CU no dividida.

[0067] Como parte de la realización de una operación de codificación en una CU, la unidad de procesamiento de predicción 100 puede dividir el bloque de vídeo de la CU entre una o más PU de la CU. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden admitir varios tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden admitir tamaños de PU de $2N \times 2N$ o $N \times N$ e interpredicción en tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $nL \times 2N$, $nR \times 2N$ o similares. El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden admitir divisiones asimétricas para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En algunos ejemplos, la unidad de procesamiento de predicción 100 puede realizar una división geométrica para dividir el bloque de vídeo de una CU entre las PU de la CU a lo largo de un límite que no coincide con los lados del bloque de vídeo de la CU en ángulos rectos.

[0068] La unidad de interpredicción 121 puede realizar una interpredicción en cada PU de la CU. La interpredicción puede proporcionar compresión temporal. Para realizar la interpredicción en una PU, la unidad de estimación de movimiento 122 puede generar información de movimiento para la PU. La unidad de compensación de movimiento 124 puede generar un bloque de vídeo predicho para la PU basado en la información de movimiento y las muestras descodificadas de imágenes distintas a la imagen asociada con la CU (por ejemplo, imágenes de referencia). En esta divulgación, un bloque de vídeo predicho generado por la unidad de compensación de movimiento 124 se puede denominar un bloque de vídeo interpredicho.

[0069] Los fragmentos pueden ser fragmentos I, fragmentos P o fragmentos B. El módulo de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento en I, todas las PU se intrapredicen. Por lo tanto, si la PU está en un fragmento I, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 no realizan la interpredicción en la PU.

[0070] Si la PU está en un fragmento P, la imagen que contiene la PU está asociada con una lista de imágenes de referencia denominada "lista 0". Cada una de las imágenes de referencia en la lista 0 contiene muestras que pueden usarse para la interpredicción de otras imágenes. Cuando la unidad de estimación de movimiento 122 realiza la operación de estimación de movimiento con respecto a una PU en un fragmento P, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en la lista 0 para un bloque de referencia para la PU. El bloque de referencia de la PU puede ser un conjunto de muestras, por ejemplo, un bloque de muestras que se corresponde más estrechamente con las muestras del bloque de vídeo de la PU. La unidad de estimación de movimiento 122 puede usar una variedad de métricas para determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras del bloque de vídeo de una PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras en el bloque de vídeo de una PU por la suma de la diferencia absoluta (SAD), la suma de diferencias al cuadrado (SSD) u otras métricas de diferencia.

[0071] Después de identificar un bloque de referencia de una PU en un fragmento P, la unidad de estimación de movimiento 122 puede generar un índice de referencia que indica la imagen de referencia de la lista 0 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. En varios ejemplos, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con diferentes grados de precisión. Por ejemplo, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con una precisión de un cuarto de muestra, una precisión de un octavo de muestra o una precisión de otra fracción de muestra. En el caso de la precisión de la fracción de muestra, los valores del bloque de referencia pueden interpolarse a partir de valores de muestra de posición de entero en la imagen de referencia. La unidad de estimación de movimiento 122 puede emitir el índice de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU basándose en el bloque de referencia identificado por la información de movimiento de la PU.

[0072] Si la PU está en un fragmento B, la imagen que contiene la PU puede estar asociada con dos listas de imágenes de referencia, denominadas "lista 0" y "lista 1." En algunos ejemplos, una imagen que contiene un fragmento B puede estar asociada con una combinación de listas que es una combinación de la lista 0 y la lista 1.

[0073] Además, si la PU está en un fragmento B, la unidad de estimación de movimiento 122 puede llevar a cabo la predicción unidireccional o la predicción bidireccional para la PU. Cuando la unidad de estimación de movimiento 122 realiza la predicción unidireccional para la PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 o la lista 1 para un bloque de referencia para la PU. La unidad de estimación de movimiento 122 puede entonces generar un índice de referencia que indica la imagen de referencia en la lista 0 o la lista 1 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial

entre la PU y el bloque de referencia. La unidad de estimación de movimiento 122 puede emitir el índice de referencia, un indicador de la dirección de predicción y el vector de movimiento como la información de movimiento de la PU. El indicador de dirección de predicción puede indicar si el índice de referencia indica una imagen de referencia en la lista 0 o la lista 1. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU basándose en el bloque de referencia indicado por la información de movimiento de la PU.

[0074] Cuando la unidad de estimación de movimiento 122 realiza la predicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 para un bloque de referencia para la PU y también puede buscar las imágenes de referencia de la lista 1 para otro bloque de referencia para la PU. La unidad de estimación de movimiento 122 puede entonces generar índices de referencia que indican las imágenes de referencia en la lista 0 y la lista 1 que contienen los bloques de referencia y los vectores de movimiento que indican desplazamientos espaciales entre los bloques de referencia y la PU. La unidad de estimación de movimiento 122 puede emitir los índices de referencia y los vectores de movimiento de la PU como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU basándose en los bloques de referencia indicados por la información de movimiento de la PU.

[0075] En algunos casos, la unidad de estimación de movimiento 122 no emite un conjunto completo de información de movimiento para una PU a la unidad de codificación por entropía 116. En lugar de eso, la unidad de estimación de movimiento 122 puede señalar la información de movimiento de una PU con referencia a la información de movimiento de otra PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar que la información de movimiento de la PU es suficientemente similar a la información de movimiento de una PU próxima. En este ejemplo, la unidad de estimación de movimiento 122 puede indicar, en una estructura sintáctica asociada con la PU, un valor que indica al descodificador de vídeo 30 que la PU tiene la misma información de movimiento que la PU próxima. En otro ejemplo, la unidad de estimación de movimiento 122 puede identificar, en una estructura sintáctica asociada con la PU, una PU próxima y una diferencia de vectores de movimiento (MVD). La diferencia de vectores de movimiento indica una diferencia entre el vector de movimiento de la PU y el vector de movimiento de la PU próxima indicada. El descodificador de vídeo 30 puede utilizar el vector de movimiento de la PU próxima indicada y la diferencia de vectores de movimiento para determinar el vector de movimiento de la PU. Haciendo referencia a la información de movimiento de una primera PU cuando se señala la información de movimiento de una segunda PU, el codificador de vídeo 20 puede ser capaz de señalar la información de movimiento de la segunda PU usando menos bits.

[0076] Como parte de la realización de una operación de codificación en una CU, la unidad de intrapredicción 126 puede llevar a cabo la intrapredicción en las PU de la CU. La intrapredicción puede proporcionar compresión espacial. Cuando la unidad de intrapredicción 126 realiza la intrapredicción en una PU, la unidad de intrapredicción 126 puede generar datos de predicción para la PU basados en muestras descodificadas de otras PU en la misma imagen. Los datos de predicción para la PU pueden incluir un bloque de vídeo predicho y varios elementos sintácticos. La unidad de intrapredicción 126 puede realizar la intrapredicción en PU en fragmentos I, fragmentos P y fragmentos B.

[0077] Para realizar la intrapredicción en una PU, la unidad de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos de predicción para la PU. Cuando la unidad de intrapredicción 126 utiliza un modo de intrapredicción para generar un conjunto de datos de predicción para la PU, la unidad de intrapredicción 126 puede extender muestras de bloques de vídeo de PU próximas a través del bloque de vídeo de la PU en una dirección y/o gradiente asociada con el modo de intrapredicción. Las PU contiguas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha, de arriba a abajo para las PU, CU y bloques de árbol. La unidad de intrapredicción 126 puede usar varios números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional, dependiendo del tamaño de la PU.

[0078] La unidad de procesamiento de predicción 100 puede seleccionar los datos de predicción para una PU entre los datos de predicción generados por la unidad de compensación de movimiento 124 para la PU o los datos de predicción generados por la unidad de intrapredicción 126 para la PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos de predicción para la PU basándose en las métricas de velocidad/distorsión de los conjuntos de datos de predicción.

[0079] Si la unidad de procesamiento de predicción 100 selecciona datos de predicción generados por la unidad de intrapredicción 126, la unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción que se utilizó para generar los datos de predicción para las PU, es decir, el modo de intrapredicción seleccionado. La unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción seleccionado de varias maneras. Por ejemplo, es probable que el modo de intrapredicción seleccionado sea el mismo que el modo de intrapredicción de una PU próxima. En otras palabras, el modo de intrapredicción de la PU próxima puede ser el modo más probable para la PU actual. De este modo, la unidad de procesamiento de predicción 100 puede generar

un elemento sintáctico para indicar que el modo de intrapredicción seleccionado es el mismo que el modo de intrapredicción de la PU próxima.

5 [0080] Como se analizó anteriormente, el codificador de vídeo 20 puede incluir la unidad de predicción entre capas 128. La unidad de predicción entre capas 128 está configurada para predecir un bloque actual (por ejemplo, un bloque actual en la EL) usando una o más capas diferentes que están disponibles en SHVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 128 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando así la eficacia de codificación y reduciendo los requisitos de recursos informáticos. Algunos ejemplos de predicción entre capas incluyen la intrapredicción entre capas, la predicción de movimiento entre capas y la predicción residual entre capas. La intrapredicción entre capas utiliza la reconstrucción de bloques ubicados conjuntamente en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa la información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción residual entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora.

15 [0081] Después de que la unidad de procesamiento de predicción 100 seleccione los datos de predicción para las PU de una CU, la unidad de generación residual 102 puede generar datos residuales para la CU restando (por ejemplo, indicado por el signo menos) los bloques de vídeo predichos de las PU de la CU del bloque de vídeo de la CU. Los datos residuales de una CU pueden incluir bloques de vídeo residuales en 2D que corresponden a diferentes componentes de muestra de las muestras en el bloque de vídeo de la CU. Por ejemplo, los datos residuales pueden incluir un bloque de vídeo residual que corresponde a diferencias entre componentes de luminancia de muestras en los bloques de vídeo predichos de las PU de la CU y componentes de luminancia de muestras en el bloque de vídeo original de la CU. Además, los datos residuales de la CU pueden incluir bloques de vídeo residuales que corresponden a las diferencias entre componentes de crominancia de muestras en los bloques de vídeo predichos de las PU de la CU y los componentes de crominancia de las muestras en el bloque de vídeo original de la CU.

20 [0082] La unidad de procesamiento de predicción 100 puede realizar la división en árbol cuaternario para dividir los bloques de vídeo residuales de una CU en subbloques. Cada bloque de vídeo residual no dividido puede estar asociado con una TU diferente de la CU. Los tamaños y posiciones de los bloques de vídeo residuales asociados con las TU de una CU pueden o no basarse en los tamaños y posiciones de los bloques de vídeo asociados con las PU de la CU. Una estructura en árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados con cada uno de los bloques de vídeo residuales. Las TU de una CU pueden corresponder a nodos de hoja del RQT.

35 [0083] La unidad de procesamiento de transformación 104 puede generar uno o más bloques de coeficientes de transformación para cada TU de una CU aplicando una o más transformaciones a un bloque de vídeo residual asociado con la TU. Cada uno de los bloques de coeficientes de transformación puede ser una matriz en 2D de coeficientes de transformación. La unidad de procesamiento de transformación 104 puede aplicar varias transformaciones al bloque de vídeo residual asociado con una TU. Por ejemplo, la unidad de procesamiento de transformación 104 puede aplicar una transformación de coseno discreta (DCT), una transformación direccional o una transformación conceptualmente similar al bloque de vídeo residual asociado con una TU.

40 [0084] Después de que la unidad de procesamiento de transformación 104 genere un bloque de coeficientes de transformación asociado con una TU, la unidad de cuantificación 106 puede cuantificar los coeficientes de transformación en el bloque de coeficientes de transformación. La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes de transformación asociado con una TU de una CU basándose en un valor de QP asociado con la CU.

45 [0085] El codificador de vídeo 20 puede asociar un valor QP con una CU de varias maneras. Por ejemplo, el codificador de vídeo 20 puede realizar un análisis de velocidad-distorsión en un bloque de árbol asociado con la CU. En el análisis de velocidad-distorsión, el codificador de vídeo 20 puede generar múltiples representaciones codificadas del bloque de árbol realizando una operación de codificación varias veces en el bloque de árbol. El codificador de vídeo 20 puede asociar diferentes valores de QP con la CU cuando el codificador de vídeo 20 genera diferentes representaciones codificadas del bloque de árbol. El codificador de vídeo 20 puede señalar que un valor QP determinado está asociado con la CU cuando el valor QP determinado está asociado con la CU en una representación codificada del bloque de árbol que tiene una velocidad de bits y una métrica de distorsión más bajas.

50 [0086] La unidad de cuantificación inversa 108 y la unidad de transformación inversa 110 pueden aplicar la cuantificación inversa y las transformaciones inversas al bloque de coeficientes de transformación, respectivamente, para reconstruir un bloque de vídeo residual a partir del bloque de coeficientes de transformación. La unidad de reconstrucción 112 puede añadir el bloque de vídeo residual reconstruido a las muestras correspondientes de uno o más bloques de vídeo predichos generados por la unidad de procesamiento de predicción 100 para producir un bloque de vídeo reconstruido asociado con una TU. Mediante la reconstrucción de

bloques de vídeo para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir el bloque de vídeo de la CU.

[0087] Después de que la unidad de reconstrucción 112 reconstruya el bloque de vídeo de una CU, la unidad de filtro 113 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo en el bloque de vídeo asociado con la CU. Después de realizar las una o más operaciones de desbloqueo, la unidad de filtro 113 puede almacenar el bloque de vídeo reconstruido de la CU en la memoria intermedia de imágenes descodificadas 114. La unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden utilizar una imagen de referencia que contiene el bloque de vídeo reconstruido para realizar la interpredicción en las PU de las imágenes posteriores. Además, la unidad de intrapredicción 126 puede usar bloques de vídeo reconstruidos en la memoria intermedia de imágenes descodificadas 114 para realizar la intrapredicción en otras PU en la misma imagen que la CU.

[0088] La unidad de codificación por entropía 116 puede recibir datos de otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 116 puede recibir bloques de coeficientes de la unidad de cuantificación 106 y puede recibir elementos sintácticos de la unidad de procesamiento de predicción 100. Cuando la unidad de codificación por entropía 116 recibe los datos, la unidad de codificación por entropía 116 puede realizar una o más operaciones de codificación por entropía para generar datos codificados por entropía. Por ejemplo, el codificador de vídeo 20 puede realizar una operación de CAVLC, una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de división de intervalo de probabilidad (PIPE) u otro tipo de operación de codificación por entropía en los datos. La unidad de codificación por entropía 116 puede emitir un flujo de bits que incluye los datos codificados por entropía.

[0089] Como parte de la realización de una operación de codificación por entropía en los datos, la unidad de codificación por entropía 116 puede seleccionar un modelo de contexto. Si la unidad de codificación por entropía 116 está realizando una operación CABAC, el modelo de contexto puede indicar las estimaciones de las probabilidades de que un bin particular tenga unos valores particulares. En el contexto de CABAC, el término "bin" se utiliza para referirse a un bit de una versión binarizada de un elemento sintáctico.

DESCODIFICADOR DE VÍDEO

[0090] La **figura 3** es un diagrama de bloques que ilustra un ejemplo del descodificador de vídeo 30 que puede implementar las técnicas de acuerdo con aspectos descritos en esta divulgación. El descodificador de vídeo 30 puede configurarse para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el descodificador de vídeo 30 se puede configurar para realizar cualquiera o todas las técnicas de esta divulgación. En algunos ejemplos, las técnicas descritas en esta divulgación pueden compartirse entre los diversos componentes del descodificador de vídeo 30. En algunos ejemplos, adicionalmente o de forma alternativa, un procesador (no mostrado) puede configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación.

[0091] Con fines de explicación, esta divulgación describe el descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la **figura 3** es para un códec de una sola capa. Sin embargo, en ciertas implementaciones, parte o la totalidad del descodificador de vídeo 30 puede duplicarse para procesar un códec multicapa.

[0092] En el ejemplo de la **figura 3**, el descodificador de vídeo 30 incluye una pluralidad de componentes funcionales. Los componentes funcionales del descodificador de vídeo 30 incluyen una unidad de descodificación por entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 159 y una memoria intermedia de imágenes descodificadas 160. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 162, una unidad de intrapredicción 164 y una unidad de predicción entre capas 166. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación, en general, recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la **figura 2**. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

[0093] El descodificador de vídeo 30 puede recibir un flujo de bits que comprende datos de vídeo codificados. El flujo de bits puede incluir una pluralidad de elementos sintácticos. Cuando el descodificador de vídeo 30 recibe el flujo de bits, la unidad de descodificación por entropía 150 puede realizar una operación de análisis sintáctico en el flujo de bits. Como resultado de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede extraer elementos sintácticos del flujo de bits. Como parte de la realización de la operación de análisis sintáctico, la unidad de descodificación por entropía 150 puede realizar la descodificación por entropía de elementos sintácticos sometidos a codificación por entropía en el flujo de bits. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de transformación inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 159 pueden realizar una operación de

reconstrucción que genera datos de vídeo descodificados basados en los elementos sintácticos extraídos del flujo de bits.

5 **[0094]** Como se analizó anteriormente, el flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de conjunto de parámetros de vídeo, unidades NAL de conjunto de parámetros de secuencia, unidades NAL de conjunto de parámetros de imagen, unidades NAL SEI, etc. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía conjuntos de parámetros de secuencia a partir de unidades NAL de conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen a partir de unidades NAL de conjuntos de parámetros de imagen, datos SEI a partir de unidades NAL SEI, etc.

15 **[0095]** Además, las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía fragmentos codificados a partir de las unidades de NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos pertenecientes a un fragmento. Los elementos sintácticos en la cabecera del fragmento pueden incluir un elemento sintáctico que identifica un conjunto de parámetros de imagen asociado con una imagen que contiene el fragmento. La unidad de descodificación por entropía 150 puede realizar operaciones de descodificación por entropía, tales como operaciones de descodificación de CABAC, en los elementos sintácticos de la cabecera del fragmento codificado para recuperar la cabecera del fragmento.

25 **[0096]** Como parte de extraer los datos del fragmento de unidades NAL de fragmentos codificados, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen los elementos sintácticos de las CU codificadas en los datos del fragmento. Los elementos sintácticos extraídos pueden incluir elementos sintácticos asociados con bloques de coeficientes de transformación. La unidad de descodificación por entropía 150 puede entonces realizar operaciones de descodificación de CABAC en algunos de los elementos sintácticos.

30 **[0097]** Después de que la unidad de descodificación por entropía 150 realice una operación de análisis sintáctico en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en la CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir un bloque de vídeo residual asociado con la CU.

35 **[0098]** Como parte de la realización de una operación de reconstrucción en una TU, la unidad de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, un bloque de coeficientes de transformación asociado con la TU. La unidad de cuantificación inversa 154 puede cuantificar de forma inversa el bloque de coeficientes de transformación de una manera similar a los procesos de cuantificación inversa propuestos para HEVC o definidos por la norma de descodificación H.264. La unidad de cuantificación inversa 154 puede utilizar un parámetro de cuantificación QP calculado por el codificador de vídeo 20 para una CU del bloque de coeficientes de transformación para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa para la unidad de cuantificación inversa 154 a aplicar.

45 **[0099]** Después de que la unidad de cuantificación inversa 154 cuantifique de forma inversa un bloque de coeficientes de transformación, la unidad de transformación inversa 156 puede generar un bloque de vídeo residual para la TU asociada con el bloque de coeficientes de transformación. La unidad de transformación inversa 156 puede aplicar una transformación inversa al bloque de coeficientes de transformación para generar el bloque de vídeo residual para la TU. Por ejemplo, la unidad de transformación inversa 156 puede aplicar una DCT inversa, una transformación de número entero inversa, una transformación de Karhunen-Loeve (KLT) inversa, una transformación de rotación inversa, una transformación direccional inversa u otra transformación inversa al bloque de coeficientes de transformación. En algunos ejemplos, la unidad de transformación inversa 156 puede determinar una transformación inversa para aplicar al bloque de coeficientes de transformación basándose en la señalización del codificador de vídeo 20. En dichos ejemplos, la unidad de transformación inversa 156 puede determinar la transformación inversa basada en una transformación señalizada en el nodo raíz de un árbol cuaternario para un bloque de árbol asociado con el bloque de coeficientes de transformación. En otros ejemplos, la unidad de transformación inversa 156 puede inferir la transformación inversa a partir de una o más características de codificación, tales como tamaño de bloque, modo de codificación o similares. En algunos ejemplos, la unidad de transformación inversa 156 puede aplicar una transformación inversa en cascada.

50 **[0100]** En algunos ejemplos, la unidad de compensación de movimiento 162 puede refinar el bloque de vídeo predicho de una PU mediante interpolación basándose en filtros de interpolación. Los identificadores para los filtros de interpolación que van a usarse para la compensación de movimiento con una precisión de submuestra pueden incluirse en los elementos sintácticos. La unidad de compensación de movimiento 162 puede usar los mismos

5 filtros de interpolación usados por el codificador de vídeo 20 durante la generación del bloque de vídeo predicho de la PU para calcular valores interpolados para muestras de subenteros de un bloque de referencia. La unidad de compensación de movimiento 162 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo con la información sintáctica recibida y usar los filtros de interpolación para producir el bloque de vídeo predicho.

10 **[0101]** Si se codifica una PU usando intrapredicción, la unidad de intrapredicción 164 puede realizar la intrapredicción para generar un bloque de vídeo predicho para la PU. Por ejemplo, la unidad de intrapredicción 164 puede determinar un modo de intrapredicción para la PU basándose en los elementos sintácticos del flujo de bits. El flujo de bits puede incluir elementos sintácticos que la unidad de intrapredicción 164 puede usar para predecir el modo de intrapredicción de la PU.

15 **[0102]** En algunos casos, los elementos sintácticos pueden indicar que la unidad de intrapredicción 164 tiene que utilizar el modo de intrapredicción de otra PU para determinar el modo de intrapredicción de la PU actual. Por ejemplo, puede ser probable que el modo de intrapredicción de la PU actual sea el mismo que el modo de intrapredicción de una PU próxima. En otras palabras, el modo de intrapredicción de la PU próxima puede ser el modo más probable para la PU actual. Por lo tanto, en este ejemplo, el flujo de bits puede incluir un elemento sintáctico pequeño que indica que el modo de intrapredicción de la PU es el mismo que el modo de intrapredicción de la PU próxima. La unidad de intrapredicción 164 puede entonces utilizar el modo de intrapredicción para generar datos de predicción (por ejemplo, muestras predichas) para la PU basándose en los bloques de vídeo de las PU espacialmente próximas.

25 **[0103]** Como se analizó anteriormente, el descodificador de vídeo 30 también puede incluir la unidad de predicción entre capas 166. La unidad de predicción entre capas 166 está configurada para predecir un bloque actual (por ejemplo, un bloque actual en la capa de mejora) usando una o más capas diferentes que están disponibles en SHVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 166 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando así la eficacia de codificación y reduciendo los requisitos de recursos informáticos. Algunos ejemplos de predicción entre capas incluyen la intrapredicción entre capas, la predicción de movimiento entre capas y la predicción residual entre capas. La intrapredicción entre capas utiliza la reconstrucción de bloques ubicados conjuntamente en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa la información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción residual entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. Cada uno de los esquemas de predicción entre capas se analiza a continuación con mayor detalle.

35 **[0104]** La unidad de reconstrucción 158 puede utilizar los bloques de vídeo residuales asociados con las TU de una CU y los bloques de vídeo predichos de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir el bloque de vídeo de la CU. De este modo, el descodificador de vídeo 30 puede generar un bloque de vídeo predicho y un bloque de vídeo residual basándose en los elementos sintácticos del flujo de bits y puede generar un bloque de vídeo basándose en el bloque de vídeo predicho y el bloque de vídeo residual.

45 **[0105]** Después de que la unidad de reconstrucción 158 reconstruya el bloque de vídeo de la CU, la unidad de filtro 159 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados con la CU. Después de que la unidad de filtro 159 realiza una operación de desbloqueo para reducir los artefactos de bloqueo asociados con la CU, el descodificador de vídeo 30 puede almacenar el bloque de vídeo de la CU en la memoria intermedia de imágenes descodificadas 160. La memoria intermedia de imágenes descodificadas 160 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la **figura 1A** o **1B**. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques de vídeo de la memoria intermedia de imágenes descodificadas 160, operaciones de intrapredicción o de interpredicción en las PU de otras CU.

55 **Modo de codificación de paleta**

60 **[0106]** En contraste con la intrapredicción e interpredicción convencionales que eliminan principalmente la redundancia entre diferentes unidades de codificación, la codificación de paleta aborda la redundancia de los valores/patrones de píxeles repetitivos dentro de la unidad de codificación. En el modo de codificación de paleta, se señala primero una tabla de búsqueda llamada paleta que asigna los valores de los píxeles a los índices de la tabla (también llamados índices de la paleta). En algunas implementaciones, la paleta tiene un tamaño máximo especificado (por ejemplo, 32 valores de píxeles). La paleta incluye entradas numeradas por los índices de la tabla que representan los valores o intensidades de los componentes de color (por ejemplo, RGB, YUV, etc.) que se pueden usar como predictores para muestras de bloques o como muestras de bloques reconstruidas finales. En algunas implementaciones, las muestras en un bloque de paleta se codifican utilizando tres modos de ejecución,

es decir, 'modo de copia de la izquierda' (o modo de ejecución), 'modo de copia de arriba' y 'modo de escape' (o modo de píxel).

5 **[0107]** Para una posición en el bloque de paleta que se codifica en el modo de copia de la izquierda, primero se
 señala un índice de la paleta seguido de "run_length" (o "palette_run") (por ejemplo, M). No es necesario señalar
 información adicional para la posición actual y las siguientes M posiciones en el bloque de la paleta porque la
 posición actual y las siguientes M posiciones en el bloque de la paleta tienen el mismo índice de la paleta que se
 10 señala para la posición actual. El índice de la paleta (por ejemplo, i) es compartido por los tres componentes de
 color, lo que significa que los valores de píxeles reconstruidos son (Y, U, V) = (paletay[i], paletau[i], paletav[i])
 (suponiendo que el espacio de color es YUV).

15 **[0108]** Para una posición en el bloque de la paleta que se codifica en el modo de copia de arriba, se señala un
 valor "run_length" (o "copy_run") (por ejemplo, N) para indicar que para las siguientes N posiciones (N + 1
 posiciones en total, incluyendo la actual) en el bloque de la paleta, el índice de la paleta es igual al índice de la
 paleta de la posición que está directamente arriba en el bloque de la paleta.

20 **[0109]** Para una posición en el bloque de la paleta que se codifica en modo de escape (o modo de píxel), se
 señala un valor de píxel correspondiente a la posición actual en el bloque de la paleta. El modo de escape puede
 señalarse utilizando un indicador de escape (por ejemplo, un valor de indicador de 1 indica que la posición actual
 25 está codificada en el modo de escape) o un índice de la paleta (por ejemplo, un valor de índice que no corresponde
 a ninguna de las entradas de la paleta o un valor de índice que es mayor o igual que el tamaño de la paleta).

Flujo de bits de la paleta

25 **[0110]** En las implementaciones existentes, un flujo de bits de la paleta (por ejemplo, un flujo de bits que incluye
 unidades de codificación codificadas en el modo de codificación de paleta) se organiza de la siguiente manera:

Tabla 1. Flujo de bits del modo de paleta

palette_entries
palette_index_map

30 **[0111]** palette_entries incluye uno o más valores de píxel cada uno asignado a un índice de la tabla. Por ejemplo,
 si una unidad de codificación dada incluye tres valores de píxeles únicos (por ejemplo, rojo, verde y azul), las
 35 entradas de la paleta pueden incluir tres entradas (0, rojo), (1, verde) y (2, azul). palette_index_map incluye uno o
 más bloques de paleta codificados usando las entradas de la paleta, donde los índices de la tabla de la paleta (por
 ejemplo, 0, 1 y 2 en el ejemplo anterior) se utilizan para indicar los valores de píxeles en el bloque de paleta.

40 **[0112]** La figura 4 ilustra una configuración de ejemplo de la CU de entrada 410, el bloque de índice 420, el píxel
 de escape 430 y la paleta 440. Como se muestra en la figura 4, la CU de entrada 410 contiene tres valores de
 píxeles únicos: blanco, gris y negro. Según la frecuencia de blanco y gris, solo se incluyen los valores de píxeles
 blancos y grises en la paleta 440, donde un valor de índice de 0 se asocia con el valor de píxel blanco y un valor
 45 de índice de 1 se asocia con el valor de píxel gris. El valor de píxel negro que no se incluye en la paleta se etiqueta
 como un píxel de escape 430, que se codifica independientemente de la paleta. Como se muestra en la figura 4,
 el bloque de índice 420 incluye un valor de índice para cada posición en el bloque. Dos posiciones en el bloque de
 índice 420 se codifican como en modo de escape (por ejemplo, sin hacer referencia a los índices de la paleta 0 o
 1). Aunque solo se utiliza un solo píxel de escape y solo dos entradas de la paleta en el ejemplo de la figura 4, los
 modos de realización de la presente solicitud no están limitados como tal, y puede usarse cualquier número de
 50 píxeles de escape y entradas de la paleta. En algunos modos de realización, el tamaño de la paleta está limitado
 a 32 entradas, y cualquier valor de píxel no asociado con una de las 32 entradas se convierte en píxeles de escape.
 El tamaño máximo de la paleta se puede establecer en cualquier número. Además, el tamaño de la CU no se limita
 a 8 píxeles por 8 píxeles, y puede ser 16x16 o cualquier otro tamaño.

Ejemplo de sintaxis del flujo de bits de la paleta

55 **[0113]** En algunas implementaciones, en el mapa de índice de la paleta, un bloque codificado en el modo de
 codificación de paleta puede tomar la siguiente forma en el flujo de bits:

while (not end) {
run_mode_flag
if (run_mode_flag == COPY_ABOVE)
run_length
else if (run_mode_flag == COPY_LEFT) {
index
if (index == ESCAPE_INDEX)
escape_pixel_value
else
run_length
}

Tabla 2. flujo de bits de palette_index_map (configuración predeterminada)

5 **[0114]** En el ejemplo ilustrado en la **Tabla 2**, según el valor de run_mode_flag, se señalizan diferentes elementos
 sintácticos en el flujo de bits. Si el indicador de modo de ejecución indica que la posición actual en el bloque de la
 paleta está codificada en el modo de copia de arriba, el cuerpo de ejecución incluye un valor de longitud de
 ejecución (el primer caso de "run_length" de arriba). Si el indicador de modo de ejecución indica que la posición
 10 actual en el bloque de la paleta está codificada en el modo "copiar de la izquierda", el cuerpo de ejecución incluye
 un valor de índice ("índice") seguido de un valor de longitud de ejecución (el segundo caso de "run_length" de
 arriba), a menos que el valor del índice corresponda a un índice de escape, en cuyo caso se señalizan los valores
 de píxeles de escape cuantificados ("escape_pixel_value").

15 **[0115]** En una implementación alternativa, se utiliza un indicador de escape explícito. Más específicamente, el
 mapa de índice de la paleta puede tomar la siguiente forma en el flujo de bits:

while (not end) {
escape_flag
if (escape_flag)
escape_pixel_value
else {
run_mode_flag
if (run_mode_flag == COPY_LEFT)
index
run_length
}

Tabla 3. flujo de bits de palette_index_map (configuración alternativa)

20 **[0116]** En el ejemplo ilustrado en la **Tabla 3**, según el valor de escape_flag, se señalizan diferentes elementos
 sintácticos en el flujo de bits. Si el indicador de escape tiene un valor de 1, el cuerpo de ejecución incluye valores
 de píxeles de escape cuantificados. Si el indicador de escape tiene un valor de 0, se señala un indicador de modo de
 25 ejecución para diferenciar los modos de "copiar de arriba" y "copiar de la izquierda". Si el indicador de modo de
 ejecución indica que la posición actual en el bloque de paleta está codificada en el modo "copiar de la izquierda",
 el flujo de bits incluye un valor de índice seguido de un valor de longitud de ejecución. De lo contrario, solo se
 señala un valor de longitud de ejecución en el flujo de bits.

30 **[0117]** La **figura 5** es un diagrama de flujo que ilustra un procedimiento 500 para codificar datos de vídeo no
 naturales en un flujo de bits de acuerdo con aspectos de la presente divulgación. Las etapas ilustradas en la **figura**
5 puede ser realizadas por un codificador de vídeo (por ejemplo, el codificador de vídeo 20), un descodificador de
 vídeo (por ejemplo, el descodificador de vídeo 30), o cualquier otro componente. Por conveniencia, el

procedimiento 500 se describe como realizado por un codificador de vídeo (también denominado simplemente codificador), que puede ser el codificador de vídeo 20, el descodificador de vídeo 30 u otro componente.

[0118] El procedimiento 500 comienza en el bloque 501. En el bloque 505, el codificador procesa una unidad de codificación (CU) que tiene una pluralidad de píxeles basándose en una paleta asociada con la CU. Como se describió anteriormente, un codificador puede obtener la paleta en base al contenido de la CU y señalar la paleta en el flujo de bits para que un descodificador pueda procesar la CU utilizando la paleta asociada con la CU. La paleta puede incluir una pluralidad de entradas de paleta que están asociadas cada una con un valor de índice y un valor de píxel. El valor de píxel puede estar asociado con uno de los píxeles en la CU. En algunos modos de realización, cada entrada de la paleta está asociada a un valor de píxel único que se encuentra en la CU. El bloque 505 puede comprender una o más etapas y/o procedimientos descritos con referencia a las **figuras 6-8**. El procedimiento termina en el bloque 510.

Agrupación de bins de desviación

[0119] En H.264, HEVC y muchas otras normas modernas de codificación de vídeo, para un elemento sintáctico, después de la binarización, el flujo de bins 0/1 se alimenta al codificador aritmético binario adaptable al contexto (CABAC), en el que el modelo de probabilidad (denominado "contexto") se selecciona y actualiza de forma adaptativa para rastrear la distribución de probabilidad no estacionaria. Como caso especial, es posible que el modelo de probabilidad no se actualice para mejorar el rendimiento del codificador por entropía. Los bins codificados con este procedimiento simplificado sin actualización de contexto se denominan bins de desviación.

[0120] En los ejemplos de las **Tablas 2 y 3**, puede haber algunas redundancias en el flujo de bits. Estas redundancias se pueden eliminar omitiendo la señalización de ciertos elementos sintácticos cuando se cumplen ciertas condiciones. Además, algunos elementos sintácticos pueden introducir la dependencia del análisis sintáctico. Por ejemplo, en la **Tabla 2**, el elemento sintáctico `run_mode_flag` puede no necesitar señalizarse si el píxel actual está en la primera línea del bloque, ya que el descodificador puede inferir que el modo de ejecución es el modo de copia de índice (por ejemplo, el modo de copia de la izquierda). Además, en el ejemplo de la **Tabla 2**, el descodificador descodifica el valor del índice primero, y dependiendo del valor del índice descodificado, el descodificador decide si el modo es el modo de copia del índice o el modo de escape (por ejemplo, basado en si el valor del índice representa o no un valor de índice de escape). Si el descodificador determina que el modo es el modo de copia de índice, el analizador del descodificador continúa analizando la longitud de ejecución. Si el descodificador determina que el modo es el modo de escape, el analizador del descodificador puede continuar analizando los valores de escape y/o la longitud de ejecución. Como los analizadores generalmente operan a una velocidad mucho mayor que los descodificadores, tal dependencia entre el motor de descodificación y el motor de análisis puede afectar el rendimiento del analizador (por ejemplo, dado que el motor de análisis puede necesitar esperar a que el motor de descodificación descodifique los bits analizados). Por lo tanto, se desea un procedimiento mejorado de procesamiento de bloques codificados en el modo de codificación de paleta. En esta solicitud, se describen varios procedimientos novedosos para organizar los elementos de la paleta en el flujo de bits para evitar o reducir la dependencia del análisis en el modo de paleta.

Modo de realización # 1: poner los valores cuantificados de píxeles de escape al final de un bloque de modo de paleta

[0121] En algunos modos de realización, todos los valores de píxeles de escape cuantificados se señalizan al final de un bloque de modo de paleta en el flujo de bits. En tales modos de realización, el reinicio del codificador por entropía se puede aplicar después de la codificación (índice, longitud de ejecución). Por ejemplo, después de codificar todos los pares posibles (índice, longitud de ejecución) en el bloque, la variable `ivlCurrRange` del motor de codificación aritmética (por ejemplo, una variable que especifica el rango del intervalo de codificación aritmética actual) se establece en 256. Con este procedimiento, el descodificador puede leer los bits del flujo de bits y tratarlos como están sin necesidad de invocar el codificador CABAC. Sin este procedimiento de restablecer la variable a 256, mientras que el contexto puede no necesitar ser actualizado, el descodificador aún puede necesitar invocar el codificador CABAC para tomar decisiones binarias. Por lo tanto, los valores de píxeles de escape cuantificados se pueden analizar en paralelo después de analizar y/o descodificar todos los pares (índice, longitud de ejecución). En un modo de realización, si los píxeles de escape se codifican utilizando un código de longitud fija, entonces los píxeles de escape se pueden analizar y descodificar en paralelo después de analizar el bloque índice-ejecución.

[0122] En otro modo de realización, si los píxeles de escape se codifican utilizando un código binario truncado, entonces cada componente de color del píxel de escape puede tomar ' k ' o ' $k + 1$ ' bits dependiendo de su intensidad cuantificada. Por ejemplo, en la codificación binaria truncada, para un elemento sintáctico con valor X , suponiendo que se conoce su valor máximo posible Max y que $n = Max + 1$ y $k = \text{suelo}(\log_2(n))$ de manera que $2^k \leq n < 2^{k+1}$ y sea $u = 2^{k+1} - n$, si $X < u$, la palabra de código binaria truncada se especifica mediante la representación binaria de X con longitud k . De lo contrario, la palabra de código binaria truncada se especifica mediante la representación binaria de $X + u$ con longitud $k+1$. En tal modo de realización, los primeros ' k ' bits de cada componente de color para todos los píxeles de escape en el bloque actual pueden ser agrupados juntos, seguido por el $(k + 1)$ ésimo bit opcional. Con dicha organización, los primeros ' k ' bits de cada componente de color para todos los píxeles de

escape se pueden analizar y descodificar en paralelo. Todavía puede existir alguna dependencia al analizar el $(k + 1)^{\text{ésimo}}$ bit opcional.

5 [0123] La **figura 6** es un diagrama de flujo que ilustra un procedimiento 600 para descodificar datos de vídeo no naturales en un flujo de bits de acuerdo con aspectos de la presente divulgación. Las etapas ilustradas en la **figura 6** pueden ser realizadas por un descodificador de vídeo (por ejemplo, el descodificador de vídeo 30) o cualquier otro componente. Por conveniencia, el procedimiento 600 se describe como realizado por un codificador de vídeo (también denominado simplemente codificador), que puede ser el descodificador de vídeo 30 u otro componente.

10 [0124] El procedimiento 600 comienza en el bloque 601. En el bloque 605, el codificador analiza una paleta asociada con la CU proporcionada en el flujo de bits. La paleta puede incluir una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice. Un ejemplo de la paleta se ilustra en la **figura 4**.

15 [0125] En el bloque 610, el codificador analiza una o más longitudes de ejecución asociadas con una CU. Como se describió anteriormente, cada longitud de ejecución indica el número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba.

20 [0126] En el bloque 615, el codificador analiza uno o más valores de índice asociados con la CU. Como se describió anteriormente, cada valor de índice indica un valor de píxel en la paleta que está asociado con la posición actual en la CU. En el ejemplo de la **figura 4**, un valor de índice de 0 indica que la posición actual en la CU tiene un valor de píxel blanco, y un valor de índice de 1 indica que la posición actual en la CU tiene un valor de píxel gris.

25 [0127] En el bloque 620, el codificador analiza uno o más valores de píxel de escape asociados con la CU. Como se describió anteriormente, cada valor de píxel de escape indica un valor de píxel que no se encuentra en la paleta asociada con la CU. En el ejemplo de la **figura 4**, las dos posiciones en la CU que tienen valores de píxel negro están codificadas en modo de escape y el codificador señala los valores de píxel negro en el flujo de bits como valores de píxel de escape. En algunos modos de realización, los valores de píxel de escape se analizan desde posiciones consecutivas en el flujo de bits (por ejemplo, al final de la porción del flujo de bits asociado con la CU). Por ejemplo, las posiciones consecutivas de los valores de píxel de escape aparecen en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU. En dichos modos de realización, después de que se hayan analizado todas las longitudes de ejecución y los valores de índice, los valores de píxel de escape se pueden procesar (por ejemplo, analizar) en paralelo. En el bloque 625, el codificador descodifica la CU basándose en la paleta analizada, las longitudes de ejecución analizadas, los valores de índice analizados y los valores de escape analizados. El procedimiento termina en el bloque 630.

35 [0128] En el procedimiento 600, uno o más de los bloques mostrados en la **figura 6** pueden eliminarse (por ejemplo, no realizarse) y/o puede conmutarse el orden en que se realiza el procedimiento. Por ejemplo, el bloque 610 y el bloque 615 se pueden realizar juntos para analizar cada par de longitud de ejecución y valor de índice asociado con la CU. En algunos modos de realización, se pueden agregar bloques adicionales al procedimiento 600. Los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la **figura 6**, y otras variaciones pueden implementarse sin apartarse del espíritu de esta divulgación.

45 **Modo de realización # 2: poner valores de índice al final de un bloque de modo de paleta**

[0129] En algunos modos de realización, todos los valores de índice se señalizan al final de un bloque de modo de paleta en el flujo de bits. En un modo de realización, todos los valores de escape cuantificados se señalizan al final del bloque de modo de paleta en el flujo de bits, siguiendo el grupo de todas las apariciones de valores de índice. En otro modo de realización, todos los valores de índice se señalizan justo antes de los valores de escape cuantificados en el flujo de bits.

55 [0130] El reinicio del codificador por entropía se puede aplicar después de la codificación de longitud de ejecución. Por ejemplo, después de codificar todas las posibles longitudes de ejecución en el bloque, la variable `ivlCurrRange` del motor de codificación aritmética (por ejemplo, una variable que especifica el rango del intervalo de codificación aritmética actual) se establece en 256. Por lo tanto, los valores de índice y/o los valores de escape se pueden analizar en paralelo después de analizar y/o descodificar todas las longitudes de ejecución en el bloque de paleta. En un modo de realización, si los valores de índice se codifican utilizando un código de longitud fija, entonces los valores de índice pueden analizarse y descodificarse en paralelo después de analizar el bloque de longitud de ejecución.

60 [0131] En otro modo de realización, si los valores de índice se codifican utilizando un código binario truncado, los índices pueden tomar 'k' o 'k + 1' bits dependiendo de su valor. En tal modo de realización, los primeros 'k' bits de cada componente de color para todos los valores de índice y/o los píxeles de escape en el bloque actual pueden ser agrupados juntos, seguidos por el opcional $(k + 1)^{\text{ésimo}}$ bit opcional. Con tal organización, los primeros 'k' bits

de todos los valores de índice y/o los valores de escape en el bloque actual se pueden analizar y descodificar en paralelo. Todavía puede existir alguna dependencia al analizar el $(k + 1)$ ésimo bit opcional.

5 [0132] La **figura 7** es un diagrama de flujo que ilustra un procedimiento 700 para descodificar datos de vídeo no naturales en un flujo de bits de acuerdo con aspectos de la presente divulgación. Las etapas ilustradas en la **figura 7** pueden ser realizadas por un descodificador de vídeo (por ejemplo, el descodificador de vídeo 30) o cualquier otro componente. Por conveniencia, el procedimiento 700 se describe como realizado por un codificador de vídeo (también denominado simplemente codificador), que puede ser el descodificador de vídeo 30 u otro componente.

10 [0133] El procedimiento 700 comienza en el bloque 701. En el bloque 705, el codificador analiza una paleta asociada con la CU proporcionada en el flujo de bits. La paleta puede incluir una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice. Un ejemplo de la paleta se ilustra en la **figura 4**.

15 [0134] En el bloque 710, el codificador analiza una o más longitudes de ejecución asociadas con una CU. Como se describió anteriormente, cada longitud de ejecución indica el número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba.

20 [0135] En el bloque 715, el codificador analiza uno o más valores de índice asociados con la CU. Como se describió anteriormente, cada valor de índice indica un valor de píxel en la paleta que está asociado con la posición actual en la CU. En el ejemplo de la **figura 4**, un valor de índice de 0 indica que la posición actual en la CU tiene un valor de píxel blanco, y un valor de índice de 1 indica que la posición actual en la CU tiene un valor de píxel gris. En el ejemplo de la **figura 7**, los valores de índice pueden analizarse desde posiciones consecutivas en el
25 flujo de bits (por ejemplo, después de todas las longitudes de ejecución asociadas con la CU). En dichos modos de realización, después de que se hayan analizado todas las longitudes de ejecución, los valores de índice se pueden procesar (por ejemplo, analizar) en paralelo. Por ejemplo, los valores de índice pueden proporcionarse inmediatamente antes de los valores de píxel de escape en el flujo de bits.

30 [0136] En el bloque 720, el codificador analiza uno o más valores de píxel de escape asociados con la CU. Como se describió anteriormente, cada valor de píxel de escape indica un valor de píxel que no se encuentra en la paleta asociada con la CU. En el ejemplo de la **figura 4**, las dos posiciones en la CU que tienen valores de píxel negro están codificadas en modo de escape y el codificador señala los valores de píxel negro en el flujo de bits como valores de píxel de escape. En algunos modos de realización, los valores de píxel de escape pueden analizarse desde posiciones consecutivas en el flujo de bits (por ejemplo, al final de la porción del flujo de bits asociada con la CU). Por ejemplo, las posiciones consecutivas de los valores de píxel de escape pueden aparecer en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU. En dichos modos de realización, después de que se hayan analizado todas las longitudes de ejecución y los valores de índice, los valores de píxel de escape se pueden procesar (por ejemplo, analizar) en paralelo. En el bloque 725, el codificador
35 descodifica la CU basándose en la paleta analizada, las longitudes de ejecución analizadas, los valores de índice analizados y los valores de escape analizados. El procedimiento termina en el bloque 730.

40 [0137] En el procedimiento 700, uno o más de los bloques mostrados en la **figura 7** pueden eliminarse (por ejemplo, no realizarse) y/o puede conmutarse el orden en que se realiza el procedimiento. En algunos modos de realización, se pueden agregar bloques adicionales al procedimiento 700. Los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la **figura 7**, y otras variaciones pueden implementarse sin apartarse del espíritu de esta divulgación.

50 **Modo de realización # 3: usar los desplazamientos para especificar la posición del valor del índice y el valor del píxel de escape**

[0138] En algunos modos de realización, se pueden señalar dos desplazamientos al comienzo de un bloque de modo de paleta en el flujo de bits, donde los dos desplazamientos especifican las posiciones iniciales del grupo de valores de índice y el grupo de valores de píxel de escape analizados anteriormente, señalados, por ejemplo, por S0 y S1, respectivamente. En el caso en que el grupo de valores de índice esté por delante del grupo de valores de píxel de escape, solo se puede señalar el valor delta entre las dos posiciones de inicio para el segundo desplazamiento (por ejemplo, se pueden señalar los desplazamientos S0 y S1-S0). Los dos desplazamientos pueden señalizarse utilizando diversos procedimientos de codificación por entropía, por ejemplo, Rice truncado, Golomb exponencial, binario truncado, longitud fija, unario, unario truncado, etc. En algunos modos de realización, también se puede señalar un valor de desplazamiento S2 que indica el final del bloque de modo de paleta. De forma alternativa, el valor delta entre el segundo desplazamiento y S2 puede ser señalado (por ejemplo, S2-S1).

Implementación de agrupación de bins de desviación

65 [0139] De acuerdo con los Modos de realización #1 y #3 descritos anteriormente, el flujo de bits palette_index_map se puede modificar de la siguiente manera:

while (byte position != S0) {
run_mode_flag
if (run_mode_flag == COPY_ABOVE)
run_length
else if (run_mode_flag == COPY_LEFT) {
index
if (index != ESCAPE_INDEX)
run_length
}
while (not end of block) {
escape_pixel_value
}

Tabla 4. Ejemplo de flujo de bits palette_index_map modificado

5

[0140] De acuerdo con los modos de realización 1, 2 y 3 descritos anteriormente, el flujo de bits palette_index_map se puede modificar de la siguiente manera:

while (byte position != S0) {	
escape_and_run_mode_flag	/*PUEDE SER '1' '01' '00'*/
if (escape_and_run_mode_flag != 1)	/* NO ES UN PÍXEL DE ESCAPE */
run_length	
}	
while (byte position != S1) {	
index	
}	
while (byte position != S2) {	
escape_pixel_value	
}	

10

Tabla 5. Ejemplo de flujo de bits palette_index_map

Comprobación de redundancia de índice

15

[0141] En algunos modos de realización, cuando se codifican los valores de índice, se puede aplicar una comprobación de redundancia. Por ejemplo, si la posición vecina anterior en el orden de escaneo de trama (denotada como posición 'x - 1') es el final de un modo de ejecución de copia de la izquierda, entonces el valor de índice actual no puede ser el mismo que el valor de índice de su posición vecina anterior. En otras palabras, si la posición 'x - 1' es válida (por ejemplo, está dentro del bloque actual o está fuera del bloque actual pero tiene un valor determinista, por ejemplo, a través del relleno del borde) y es el final de una ejecución de copia de la izquierda, entonces el valor de índice para la posición 'x' no puede ser igual al valor de índice en la posición 'x - 1' (Caso 1). La razón es que si los dos valores de índice fueran los mismos, se habrían fusionado en una ejecución de copia de la izquierda más larga. En otro ejemplo, si la posición vecina anterior en el orden de escaneo de trama es el final de un modo de ejecución de copia de arriba, y/o si se cumple una restricción adicional de que el vecino de la posición actual no sea un píxel de escape, entonces el valor actual no puede ser el mismo que el valor de índice de su vecino superior. En otras palabras, si la posición 'x - 1' es válida y es el final de una ejecución de copia de arriba, y/o si se cumple una restricción adicional de que el valor del píxel de la posición de arriba 'x' no sea un píxel de escape, entonces el valor de índice para la posición 'x' no puede ser igual al valor de índice de su vecino superior

20

25

(Caso 2). La razón es que si los dos valores de índice fueran los mismos, se habrían fusionado en una ejecución de copia de arriba más larga. Por lo tanto, estos ejemplos asumen que el codificador sigue el principio de "ejecución más larga posible". En cualquiera de estos casos, el rango (por ejemplo, el valor *Max* descrito anteriormente) se puede reducir en uno y se pueden lograr ahorros de bits.

[0142] Sin embargo, para el Caso 1, la descodificación del valor del índice en la posición 'x' depende de la reconstrucción del valor del índice en la posición 'x - 1' (por ejemplo, dado que el descodificador necesita conocer el valor del índice en la posición 'x - 1' para determinar cuál no puede ser el valor del índice en la posición 'x'). Sin embargo, es posible que el valor de índice para la posición 'x-1' no esté fácilmente disponible en el momento en que se descodifique el valor de índice en la posición 'x'. Por lo tanto, esta dependencia puede causar algún retraso en el proceso de descodificación. Para eliminar esta dependencia, en algunos modos de realización, la verificación condicional para el Caso 1 puede estar deshabilitada. En algunos de dichos modos de realización, la verificación condicional para el Caso 2 aún puede realizarse (por ejemplo, dado que es más probable que esté disponible el valor de índice de una posición sobre el píxel actual). En dichos modos de realización, la verificación condicional para el caso 1 está completamente deshabilitada. De forma alternativa, la verificación condicional para el Caso 1 puede estar deshabilitada solo para un caso específico. Por ejemplo, la verificación condicional para el Caso 1 puede estar deshabilitada solo cuando la función 'ejecución limitada' está habilitada, donde se indica el valor de índice de la paleta máximo para el que se codifica la longitud de ejecución (o el valor de índice de la paleta mínimo para el cual no se encuentra codificada la longitud de ejecución).

[0143] En el Caso 2, la comprobación de si el píxel de la posición superior 'x' es o no un píxel de escape puede eliminarse si el píxel de escape se admite en las ejecuciones de copia de la izquierda o copia de arriba. Por ejemplo, a los píxeles de escape se les pueden asignar uno o más valores de índice que no están en la paleta y tienen sus propias ejecuciones (por ejemplo, al igual que los valores de píxeles en la paleta). De manera similar, la comprobación de si el píxel a la izquierda de la posición 'x' es o no un píxel de escape (por ejemplo, una etapa que debe realizarse antes de analizar el valor de índice actual en la posición x, si la implementación dada no permite que los píxeles de escape sean copiados desde la izquierda o desde arriba) se pueden eliminar si el píxel de escape se admite en las ejecuciones de copia de la izquierda o copia de arriba.

[0144] La **figura 8** es un diagrama de flujo que ilustra un procedimiento 800 para codificar datos de vídeo no naturales en un flujo de bits de acuerdo con aspectos de la presente divulgación. Las etapas ilustradas en la **figura 8** puede ser realizadas por un codificador de vídeo (por ejemplo, el codificador de vídeo 20), un descodificador de vídeo (por ejemplo, el descodificador de vídeo 30), o cualquier otro componente. Por conveniencia, el procedimiento 800 se describe como realizado por un codificador de vídeo (también denominado simplemente codificador), que puede ser el codificador de vídeo 20, el descodificador de vídeo 30 u otro componente.

[0145] El procedimiento 800 comienza en el bloque 801. En el bloque 805, el codificador determina que una posición a la izquierda de una posición actual en la CU está asociada con el final de una ejecución de copia de arriba. Como se describió anteriormente, cuando una o más posiciones en la CU se codifican en modo copia de arriba, se señala una longitud de ejecución que indica el número de posiciones consecutivas, comenzando desde e incluyendo la posición inicial en la CU, que están asociadas con el modo copia de arriba. En base a la longitud de la ejecución, el codificador puede determinar que una posición determinada (por ejemplo, una posición que precede inmediatamente a la posición actual en la CU) es el final de una ejecución de copia de arriba.

[0146] En el bloque 810, el codificador, en respuesta a determinar que la posición a la izquierda de la posición actual está asociada con el final de una ejecución de copia de arriba, determina el valor de índice asociado con la posición actual sin determinar si la posición superior a la posición actual está asociada con un valor de píxel de escape. Como se describió anteriormente, típicamente, el codificador necesita determinar si la posición sobre la posición actual está asociada con un valor de píxel de escape antes de determinar el valor de índice del píxel actual (por ejemplo, si la posición sobre la posición actual está asociada con un píxel de escape, la suposición de que el valor de índice de la posición actual no es igual al valor de índice de la posición por encima de la posición actual puede llegar a ser inexacta. Sin embargo, en algunos modos de realización de la presente divulgación, los píxeles de escape pueden ser parte de las ejecuciones de copia de la izquierda o copia de arriba. Por lo tanto, no es necesaria una comprobación por separado de la posición por encima de la posición actual. El procedimiento termina en el bloque 815.

[0147] En el procedimiento 800, uno o más de los bloques mostrados en la **figura 8** pueden eliminarse (por ejemplo, no realizarse) y/o puede conmutarse el orden en que se realiza el procedimiento. En algunos modos de realización, se pueden agregar bloques adicionales al procedimiento 800. Los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la **figura 8**, y otras variaciones pueden implementarse sin apartarse del espíritu de esta divulgación.

Mantener el número máximo de bins de índice constante

[0148] En algunas implementaciones de codificación de modo de paleta, los índices de la paleta se codifican utilizando código binario truncado. Suponiendo que el índice más grande en la CU actual es *N* (por ejemplo, el

valor de índice se elige entre $\{0, 1, 2, \dots, M\}$, inclusive, entonces el número de bins para codificar cada índice puede ser $\lceil \log_2(N+1) \rceil$ o $\lfloor \log_2(N+1) \rfloor$, si estos dos valores no son iguales. A medida que al píxel de escape se le asigna el índice más grande (por ejemplo, después de que a todos los valores de píxel en la paleta se les asignan sus valores de índice), la codificación del píxel de escape toma $\lceil \log_2(N+1) \rceil$ bins.

[0149] En algunos casos, al aprovechar dependencias, como los procedimientos descritos anteriormente, el valor de símbolo más grande para el índice actual puede reducirse en uno. En otras palabras, el píxel de escape puede tomar contenedores $\lceil \log_2 M \rceil$ o $\lfloor \log_2(N+1) \rfloor$ bins dependiendo de si la condición de eliminación de redundancia está habilitada o no. Como resultado, el descodificador puede necesitar primero calcular si el valor del símbolo de índice más grande es N o $N-1$ para determinar cuántos bins se necesitan para descodificar el índice. Esto introdujo retraso del chip adicional y tiene un efecto negativo en la canalización del descodificador. En algunos modos de realización, para eliminar este retraso y cualquier efecto negativo en la canalización del descodificador, este mecanismo de eliminación de redundancia puede estar restringido. Por ejemplo, el número máximo de bins utilizados para la codificación de índice se puede establecer en una constante. En un ejemplo, el valor de índice del píxel de escape siempre puede usar $\text{techo}(\log_2(N+1))$ bins. En otro ejemplo, si $\text{techo}(\log_2(N+1))$ es igual a $\text{techo}(\log_2(N))$, se habilita el procedimiento de eliminación de redundancia para píxeles de escape, y de lo contrario, se desactiva el procedimiento de eliminación de redundancia para píxeles de escape.

Diagrama de flujo lateral del codificador

[0150] La **figura 9** es un diagrama de flujo que ilustra un procedimiento 900 para codificar datos de vídeo en un flujo de bits, de acuerdo con aspectos de la presente divulgación. Por ejemplo, los datos de vídeo pueden ser datos de vídeo no naturales que incluyen contenidos de pantalla generados por ordenador. Las etapas ilustradas en la **figura 9** pueden ser realizadas por un codificador de vídeo (por ejemplo, el codificador de vídeo 20) o cualquier otro componente. Por conveniencia, el procedimiento 900 se describe como realizado por un codificador de vídeo (también denominado simplemente codificador), que puede ser el codificador de vídeo 20 u otro componente.

[0151] El procedimiento 900 comienza en el bloque 901. En el bloque 905, el codificador analiza una pluralidad de píxeles en una unidad de codificación (CU). Cada píxel en la CU puede estar asociado con un valor de píxel. Por ejemplo, varios píxeles en la CU pueden tener el mismo valor de píxel.

[0152] En el bloque 910, el codificador genera una paleta basada en la pluralidad de píxeles en la CU. La paleta puede incluir una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice. Un ejemplo de la paleta se ilustra en la **figura 4**.

[0153] En el bloque 915, el codificador determina una o más longitudes de ejecución asociadas con la CU. Como se describió anteriormente, cada longitud de ejecución indica el número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba.

[0154] En el bloque 920, el codificador determina uno o más valores de índice asociados con la CU. Como se describió anteriormente, cada valor de índice indica un valor de píxel en la paleta que está asociado con la posición actual en la CU. En el ejemplo de la **figura 4**, un valor de índice de 0 indica que la posición actual en la CU tiene un valor de píxel blanco, y un valor de índice de 1 indica que la posición actual en la CU tiene un valor de píxel gris.

[0155] En el bloque 925, el codificador determina uno o más valores de píxel de escape asociados con la CU. Como se describió anteriormente, cada valor de píxel de escape indica un valor de píxel que no se encuentra en la paleta asociada con la CU. En el ejemplo de la **figura 4**, las dos posiciones en la CU que tienen valores de píxel negro están codificadas en modo de escape y el codificador señala los valores de píxel negro en el flujo de bits como valores de píxel de escape.

[0156] En el bloque 930, el codificador codifica la CU en base a la paleta generada, las longitudes de ejecución determinadas, los valores de índice determinados y los valores de píxeles de escape determinados. En algunos modos de realización, los valores de píxel de escape están codificados en posiciones consecutivas en el flujo de bits (por ejemplo, al final de la porción del flujo de bits asociado con la CU). Por ejemplo, las posiciones consecutivas de los valores de píxel de escape aparecen en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU. En dichos modos de realización, después de que todas las longitudes de ejecución y los valores de índice hayan sido analizados por un descodificador, los valores de píxel de escape pueden procesarse (por ejemplo, analizarse) en paralelo. El procedimiento termina en el bloque 935.

[0157] En el procedimiento 900, uno o más de los bloques mostrados en la **figura 9** pueden eliminarse (por ejemplo, no realizarse) y/o puede conmutarse el orden en que se realiza el procedimiento. En algunos modos de realización, se pueden agregar bloques adicionales al procedimiento 900. Los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la **figura 9**, y otras variaciones pueden implementarse sin apartarse del espíritu de esta divulgación.

Otras Consideraciones

5 [0158] La información y las señales divulgadas en el presente documento pueden representarse usando cualquiera entre varias tecnologías y técnicas diferentes. Por ejemplo, los datos, las instrucciones, los comandos, la información, las señales, los bits, los símbolos y los chips que puedan haberse mencionado a lo largo de la descripción anterior pueden representarse mediante tensiones, corrientes, ondas electromagnéticas, campos o partículas magnéticos, campos o partículas ópticos o cualquier combinación de los mismos.

10 [0159] Los diversos bloques lógicos y etapas de algoritmo ilustrativos descritos en relación con los modos de realización divulgados en el presente documento pueden implementarse como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, anteriormente se han descrito, en general, diversos componentes, bloques y etapas ilustrativos en términos de su funcionalidad. Que dicha funcionalidad se implemente como hardware o software depende de la solicitud particular y de las restricciones de diseño impuestas al sistema global. Los expertos en la materia pueden implementar la funcionalidad descrita de distintas maneras para cada aplicación particular, pero no se debería interpretar que dichas decisiones de implementación suponen apartarse del alcance de la presente divulgación.

20 [0160] Las técnicas descritas en el presente documento pueden implementarse en hardware, software, firmware o en cualquier combinación de lo anterior. Dichas técnicas pueden implementarse en cualquiera entre una variedad de dispositivos tales como ordenadores de propósito general, equipos manuales de dispositivos de comunicación inalámbrica o dispositivos de circuitos integrados que tienen múltiples usos, incluyendo su aplicación en equipos manuales de dispositivos de comunicación inalámbrica y otros dispositivos. Cualquier característica descrita como dispositivos o componentes pueden implementarse juntas en un dispositivo lógico integrado o por separado, como dispositivos lógicos discretos pero interoperables. Si se implementan en software, las técnicas pueden realizarse, al menos en parte, mediante un medio de almacenamiento de datos legible por ordenador que comprenda código de programa que incluye instrucciones que, cuando se ejecutan, realizan uno o más de los procedimientos descritos anteriormente. El medio de almacenamiento de datos legible por ordenador puede formar parte de un producto de programa informático, que puede incluir materiales de embalaje. El medio legible por ordenador puede comprender memoria o medios de almacenamiento de datos, tales como memoria de acceso aleatorio (RAM), tal como memoria de acceso aleatorio dinámica síncrona (SDRAM), memoria de sólo lectura (ROM), memoria de acceso aleatorio no volátil (NVRAM), memoria de sólo lectura programable y borrable eléctricamente (EEPROM), memoria FLASH, medios de almacenamiento de datos magnéticos u ópticos, y similares. Las técnicas se pueden realizar, adicionalmente o de forma alternativa, al menos en parte, por un medio de comunicación legible por ordenador que lleva o comunica código de programa en forma de instrucciones o estructuras de datos y a las que se puede acceder, leer y/o ejecutar por medio de un ordenador, tales como señales u ondas propagadas.

35 [0161] El código de programa puede ser ejecutado por un procesador, que puede incluir 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 de compuertas programables en el terreno (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Un procesador de este tipo puede estar configurado para realizar cualquiera de las técnicas descritas en esta divulgación. Un procesador de propósito general puede ser un microprocesador, pero, como alternativa, el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estados convencional. Un procesador también puede implementarse como una combinación de dispositivos informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, uno o más microprocesadores junto con un núcleo de DSP o cualquier otra configuración de este tipo. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior, cualquier combinación de la estructura anterior, o cualquier otra estructura o aparato adecuados para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de software o hardware dedicados configurados para la codificación y la decodificación, o incorporados en un codificador-descodificador de vídeo combinado (CODEC). Asimismo, las técnicas podrían implementarse por completo en uno o más circuitos o elementos lógicos.

55 [0162] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Varios componentes o unidades se describen en esta divulgación para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren su realización por diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores como se describe anteriormente, junto con el software y/o firmware adecuados.

65 [0163] Aunque lo anterior ha sido descrito en relación con diversos modos de realización diferentes, las características o elementos de un modo de realización se pueden combinar con otros modos de realización sin apartarse de las enseñanzas de esta divulgación. Sin embargo, las combinaciones de características entre los

modos de realización respectivos no están necesariamente limitadas a ellas. Se han descrito diversos modos de realización de la divulgación. Estos y otros modos de realización están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para descodificar datos de vídeo proporcionados en un flujo de bits, incluyendo el flujo de bits una unidad de codificación, CU, codificada en modo de paleta, el procedimiento que comprende:
- 5 analizar una paleta asociada con la CU proporcionada en el flujo de bits, incluyendo la paleta una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice;
- 10 analizar una o más longitudes de ejecución proporcionadas en el flujo de bits que están asociadas con la CU, indicando cada longitud de ejecución un número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba;
- 15 analizar uno o más valores de índice proporcionados en el flujo de bits que están asociados con la CU, indicando cada valor de índice un valor de píxel en la paleta que está asociado con la posición actual en la CU, en el que los valores de índice se analizan desde las primeras posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución asociadas con la CU;
- 20 analizar uno o más valores de píxel de escape proporcionados en el flujo de bits que están asociados con la CU, indicando cada valor de píxel de escape un valor de píxel que no está en la paleta, en el que los valores de píxel de escape se analizan desde la segunda posición consecutiva en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU; y
- 25 descodificar la CU basándose en la paleta analizada, las longitudes de ejecución analizadas, los valores de índice analizados y los valores de escape analizados.
2. El procedimiento según la reivindicación 1, que comprende, además, restablecer una variable de codificación aritmética que especifica un rango de un intervalo actual asociado con la CU después de analizar todas las longitudes de ejecución y los valores de índice asociados con la CU.
- 30 3. El procedimiento según la reivindicación 1, en el que los valores de píxel de escape asociados con la CU se analizan en paralelo después de analizar todas las longitudes de ejecución y los valores de índice asociados con la CU.
- 35 4. El procedimiento según la reivindicación 1, que comprende, además, restablecer una variable de codificación aritmética que especifica un rango de un intervalo actual asociado con la CU después de analizar todas las longitudes de ejecución asociadas con la CU.
- 40 5. El procedimiento según la reivindicación 1, en el que los valores de índice asociados con la CU se analizan en paralelo después de analizar todas las longitudes de ejecución asociadas con la CU.
6. El procedimiento según la reivindicación 1, que comprende además:
- 45 determinar que una primera posición en la CU que precede inmediatamente a la posición actual está asociada con un final de una ejecución de copia de arriba; y
- 50 en respuesta a la determinación de que la primera posición en la CU está asociada con un final de una ejecución de copia de arriba, determinar un valor de índice asociado con la posición actual sin determinar si una segunda posición inmediatamente por encima de la posición actual en la CU está asociada con un valor de píxel de escape.
7. El procedimiento según la reivindicación 1, en el que la CU incluye una ejecución de copia de arriba o una ejecución de copia de la izquierda que incluye un valor de píxel de escape.
- 55 8. Un aparato para descodificar datos de vídeo proporcionados en un flujo de bits, que comprende:
- 60 una memoria configurada para almacenar datos de vídeo asociados con el flujo de bits, incluyendo el flujo de bits una unidad de codificación (CU) codificada en modo de paleta; y
- un procesador en comunicación con la memoria y configurado para:
- 65 analizar una paleta asociada con la CU proporcionada en el flujo de bits, incluyendo la paleta una pluralidad de entradas de la paleta que están asociadas cada una con un valor de índice y un valor de píxel asociado con el valor de índice;

analizar una o más longitudes de ejecución proporcionadas en el flujo de bits que están asociadas con la CU, indicando cada longitud de ejecución un número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba;

5

analizar uno o más valores de índice proporcionados en el flujo de bits que están asociados con la CU, indicando cada valor de índice un valor de píxel en la paleta que está asociado con la posición actual en la CU, en donde los valores de índice se analizan desde las primeras posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución asociadas con la CU;

10

analizar uno o más valores de píxel de escape proporcionados en el flujo de bits que están asociados con la CU, indicando cada valor de píxel de escape un valor de píxel que no está en la paleta, en el que los valores de píxel de escape se analizan desde las segundas posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU; y

15

descodificar la CU basándose en la paleta analizada, las longitudes de ejecución analizadas, los valores de índice analizados y los valores de escape analizados.

20

9. El aparato según la reivindicación 8, en el que el procesador está configurado, además, para restablecer una variable de codificación aritmética que especifica un rango de un intervalo actual asociado con la CU después de analizar todas las longitudes de ejecución y los valores de índice asociados con la CU.

25

10. El aparato según la reivindicación 8, en el que el procesador está configurado para analizar los valores de píxel de escape asociados con la CU en paralelo después de analizar todas las longitudes de ejecución y los valores de índice asociados con la CU.

30

11. El aparato según la reivindicación 8, en el que el procesador está configurado, además, para restablecer una variable de codificación aritmética que especifica un rango de un intervalo actual asociado con la CU después de analizar todas las longitudes de ejecución asociadas con la CU y/o en el que el procesador está configurado para analizar los valores de índice asociados con la CU en paralelo después de analizar todas las longitudes de ejecución asociadas con la CU y/o en el que el procesador está configurado, además, para:

35

determinar que una primera posición en la CU que precede inmediatamente a la posición actual está asociada con un final de una ejecución de copia de arriba; y

40

en respuesta a la determinación de que la primera posición en la CU está asociada con un final de una ejecución de copia de arriba, determinar un valor de índice asociado con la posición actual sin determinar si una segunda posición inmediatamente por encima de la posición actual en la CU está asociada con un el valor de píxel de escape y/o en el que la CU incluye una ejecución de copia de arriba o una ejecución de copia de la izquierda que incluye un valor de píxel de escape.

45

12. Un procedimiento para codificar datos de vídeo en un flujo de bits que comprende:

analizar una pluralidad de píxeles en una unidad de codificación, CU, teniendo cada píxel un valor de píxel asociado con la misma;

50

generar una paleta basándose en la pluralidad de píxeles en la CU, incluyendo la paleta una pluralidad de entradas de la paleta que están asociadas con un valor de índice y un valor de píxel asociado con el valor de índice;

55

determinar una o más longitudes de ejecución asociadas con la CU en el flujo de bits, indicando cada longitud de ejecución un número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba;

60

determinar uno o más valores de índice asociados con la CU en el flujo de bits, indicando cada valor de índice un valor de píxel en la paleta que está asociado con la posición actual en la CU;

determinar uno o más valores de píxel de escape asociados con la CU en el flujo de bits, indicando cada valor de píxel de escape un valor de píxel que no está en la paleta; y

65

codificar la CU en base a la paleta generada, las longitudes de ejecución determinadas, los valores de índice determinados y los valores de píxel de escape determinados, en el que los valores de índice se codifican en primeras posiciones consecutivas en el flujo de bits después de todas las longitudes de

ejecución asociadas con la CU y los valores de píxel de escape se codifican en segundas posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU.

5 **13.** El procedimiento según la reivindicación 12, que comprende además:

determinar que una primera posición en la CU que precede inmediatamente a la posición actual está asociada con un final de una ejecución de copia de arriba; y

10 en respuesta a la determinación de que la primera posición en la CU está asociada con un final de una ejecución de copia de arriba,
 determinar un valor de índice asociado con la posición actual sin determinar si una segunda posición inmediatamente por encima de la posición actual en la CU está asociada con un valor de píxel de escape y/o en el que la CU incluye una de una ejecución de copia de arriba o una ejecución de copia de la izquierda que incluye un valor de píxel de escape y/o que comprende, además, determinar un primer desplazamiento que indica una posición en el flujo de bits que corresponde a un valor de índice que tiene una posición más temprana entre los valores de índice asociados con la CU y/o que comprende, además, la determinación de un segundo desplazamiento que indica una posición en el flujo de bits que corresponde a un valor de píxel de escape que tiene una posición más temprana entre los valores de píxel de escape asociados con la CU.

14. Un aparato para codificar datos de vídeo en un flujo de bits, que comprende:

25 una memoria configurada para almacenar datos de vídeo asociados con el flujo de bits, incluyendo el flujo de bits una unidad de codificación, CU, codificada en modo de paleta; y

un procesador en comunicación con la memoria y configurado para:

30 analizar una pluralidad de píxeles en una unidad de codificación, CU, teniendo cada píxel un valor de píxel asociado con la misma;

35 generar una paleta basándose en la pluralidad de píxeles en la CU, incluyendo la paleta una pluralidad de entradas de la paleta que están asociadas con un valor de índice y un valor de píxel asociado con el valor de índice;

40 determinar una o más longitudes de ejecución asociadas con la CU en el flujo de bits, indicando cada longitud de ejecución un número de posiciones consecutivas, comenzando desde e incluyendo una posición actual en la CU, que están asociadas con un modo de copia de la izquierda o un modo de copia de arriba;

determinar uno o más valores de índice asociados con la CU en el flujo de bits; indicando cada valor de índice un valor de píxel en la paleta que está asociado con la posición actual en la CU;

45 determinar uno o más valores de píxel de escape asociados con la CU en el flujo de bits; indicando cada valor de píxel de escape un valor de píxel que no está en la paleta; y

50 codificar la CU basándose en la paleta generada, las longitudes de ejecución determinadas, los valores de índice determinados y los valores de píxel de escape determinados, en el que los valores de índice se codifican en primeras posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución asociadas con la CU y los valores de píxel de escape se codifican en segundas posiciones consecutivas en el flujo de bits después de todas las longitudes de ejecución y los valores de índice asociados con la CU.

55 **15.** El aparato según el ejemplo 14, en el que el procesador está configurado además para:

determinar que una primera posición en la CU que precede inmediatamente a la posición actual está asociada con un final de una ejecución de copia de arriba; y

60 en respuesta a la determinación de que la primera posición en la CU está asociada con un final de una ejecución de copia de arriba, determinar un valor de índice asociado con la posición actual sin determinar si una segunda posición inmediatamente por encima de la posición actual en la CU está asociada con un valor de píxel de escape y/o en el que la CU incluye uno de una ejecución de copia de arriba o una ejecución de copia de la izquierda que incluye un valor de píxel de escape y/o en el que el procesador está configurado, además, para determinar un primer desplazamiento que indica una posición en el flujo de bits que corresponde a un valor de índice que tiene una posición más temprana entre los valores de índice asociados con la CU y/o que comprende, además, determinar un segundo desplazamiento que

indica una posición en el flujo de bits que corresponde a un valor de píxel de escape que tiene una posición más temprana entre los valores de píxeles de escape asociados con la CU.

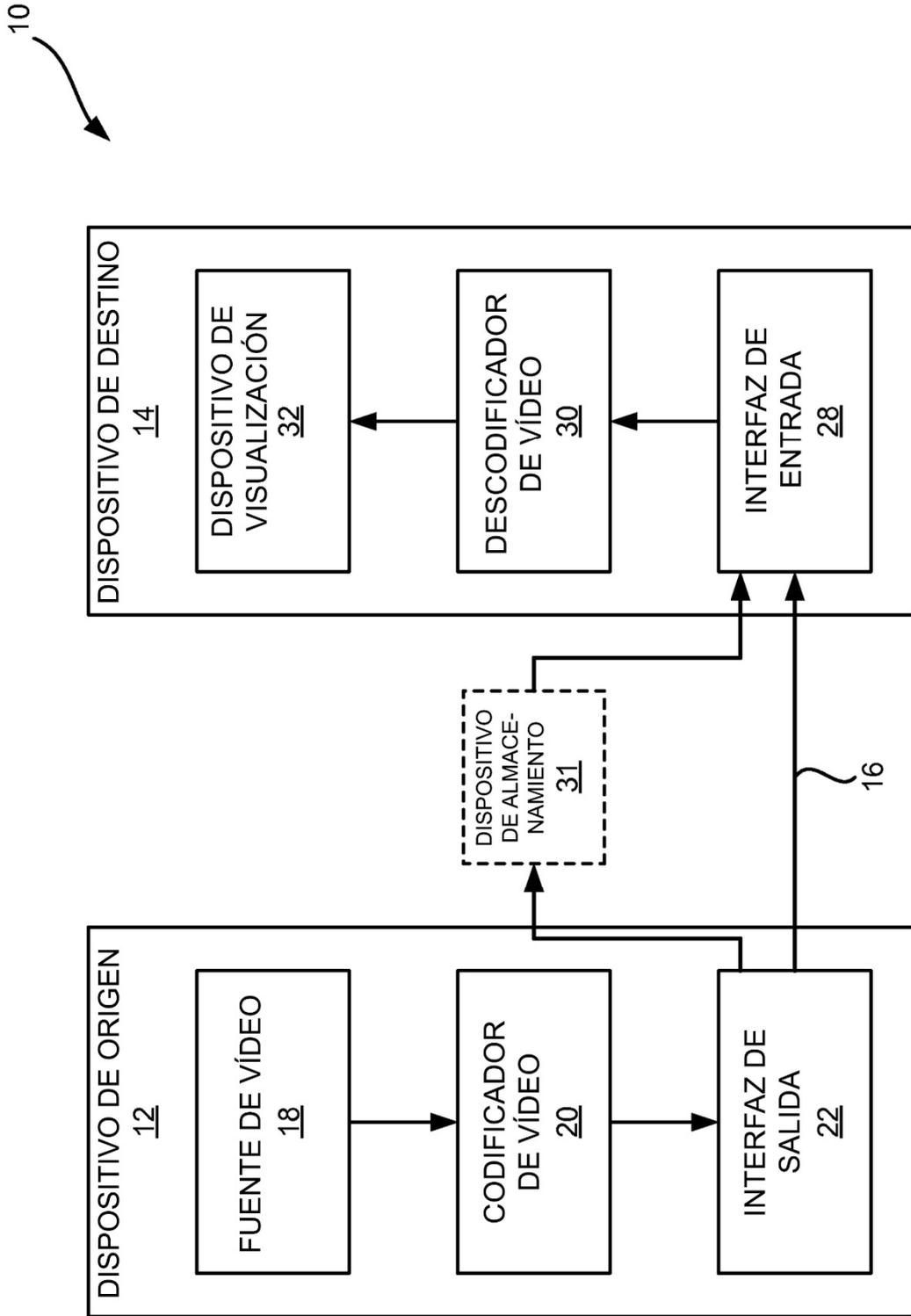


FIG. 1A

10'

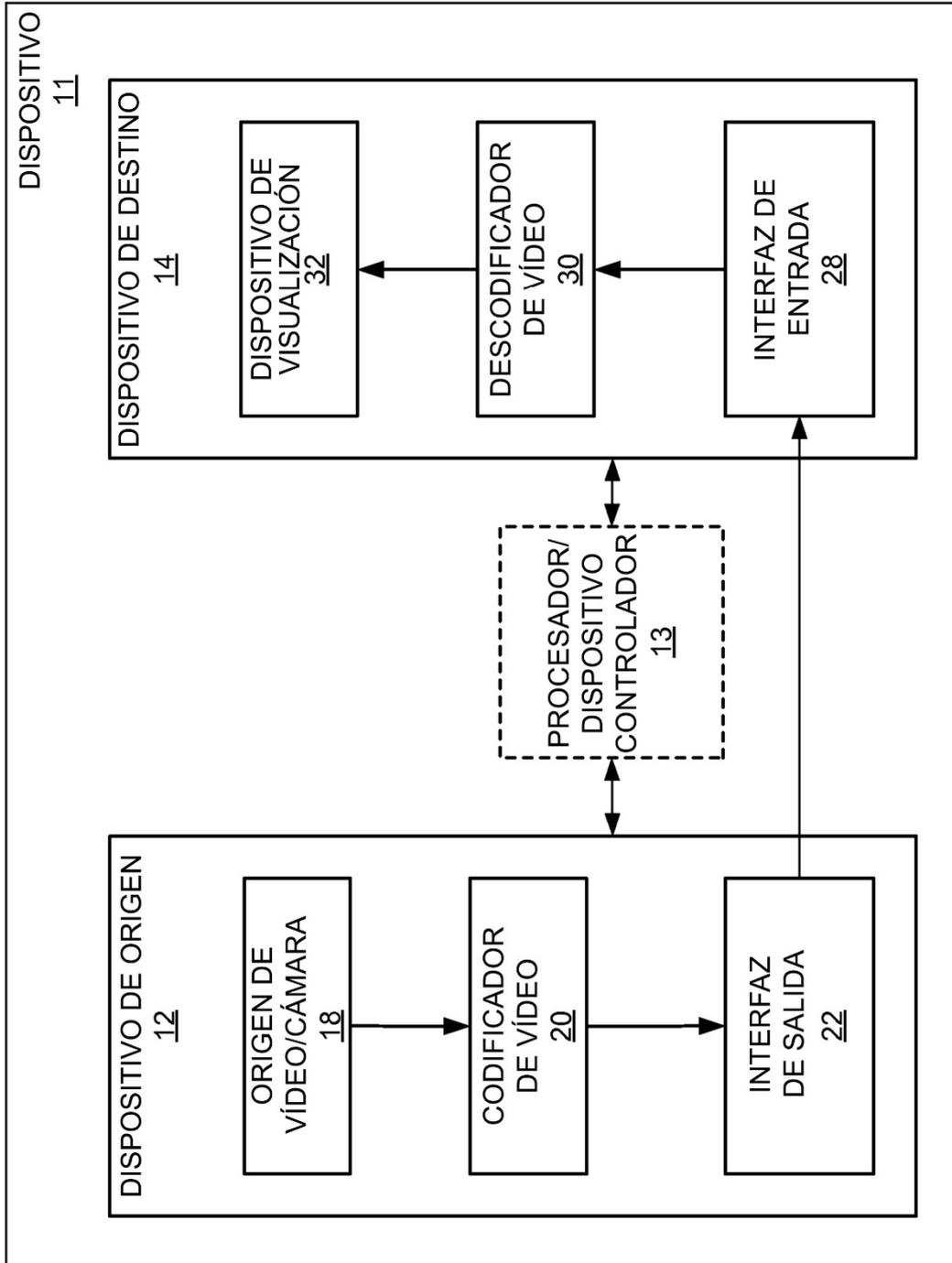


FIG. 1B

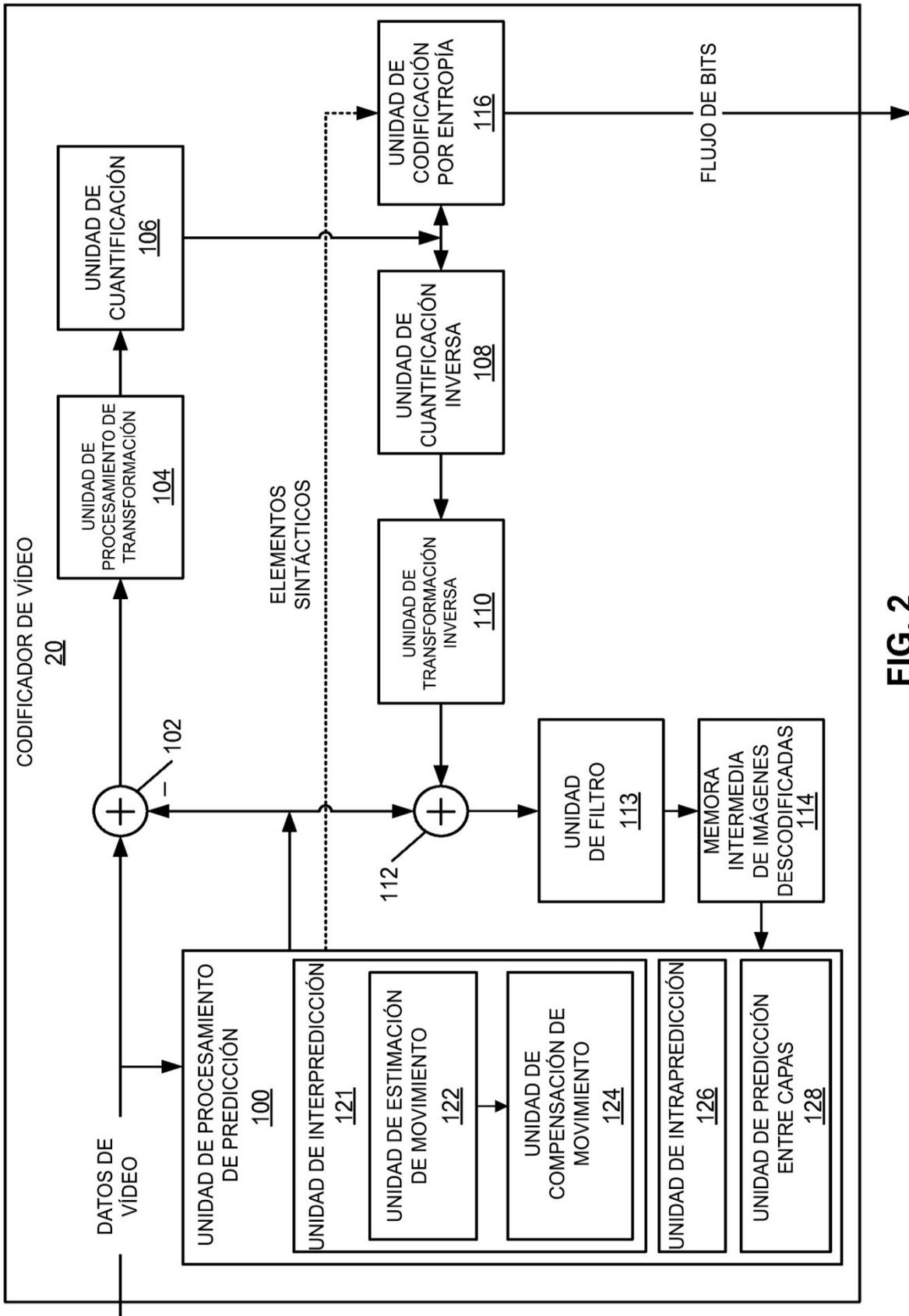


FIG. 2

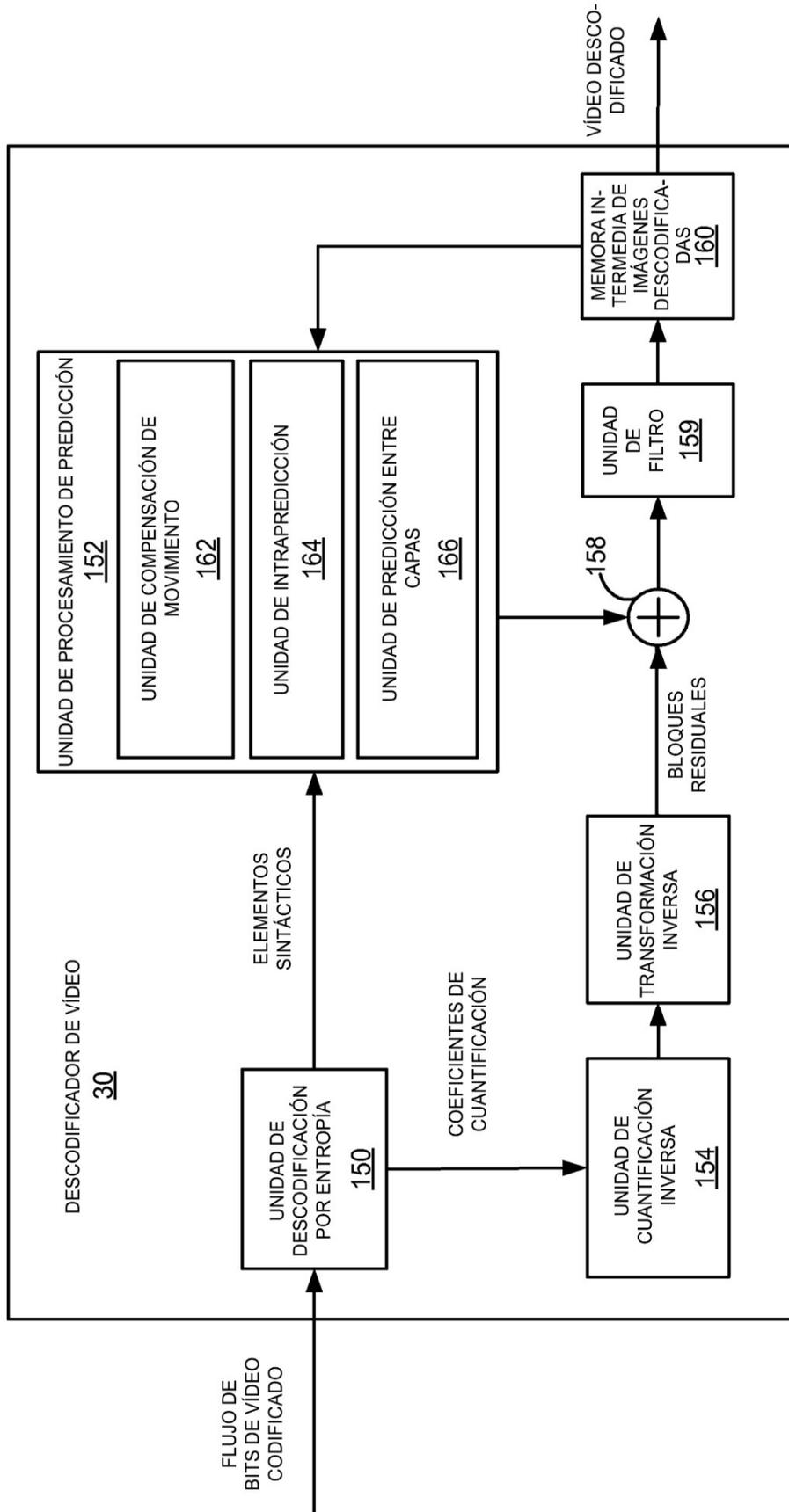


FIG. 3

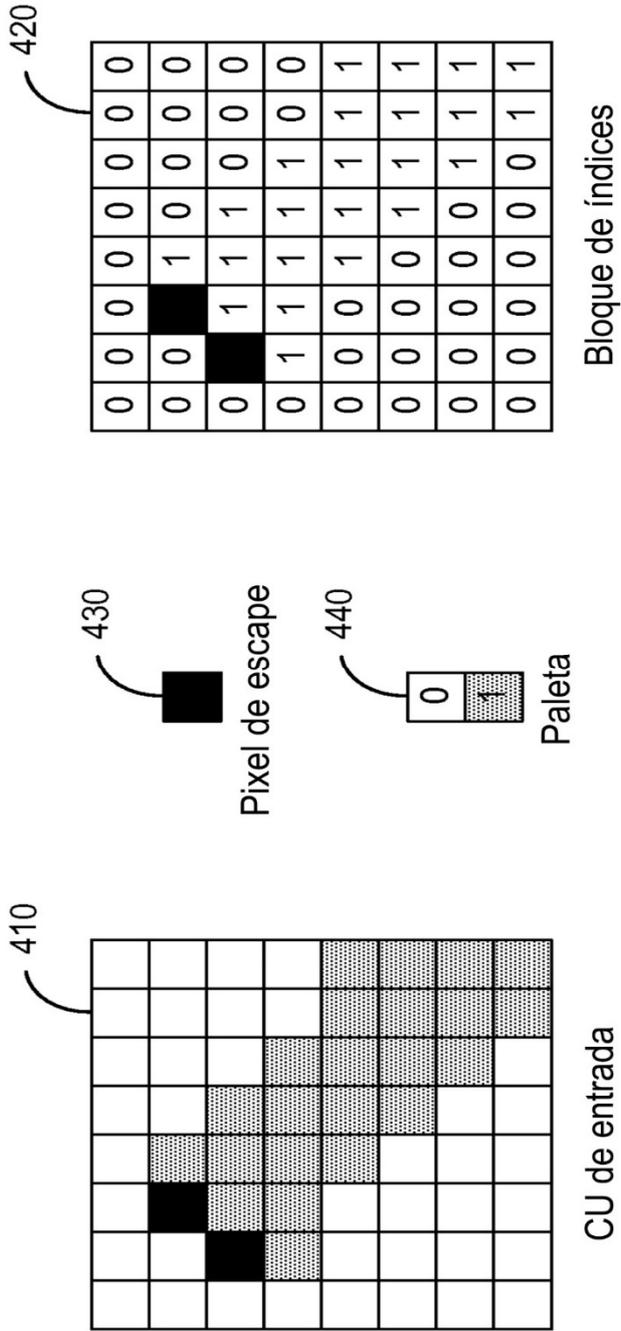


FIG. 4

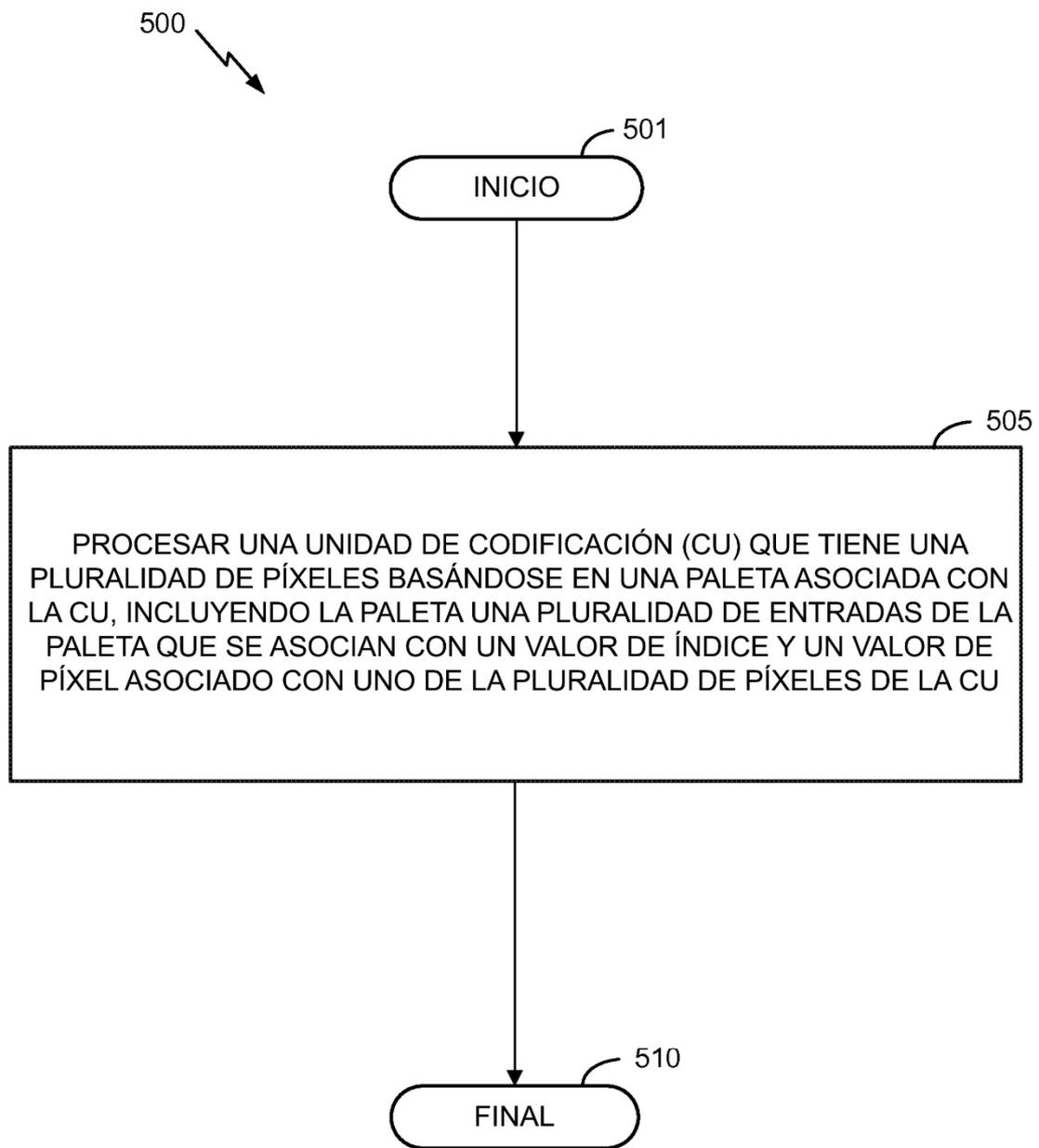


FIG. 5



FIG. 6

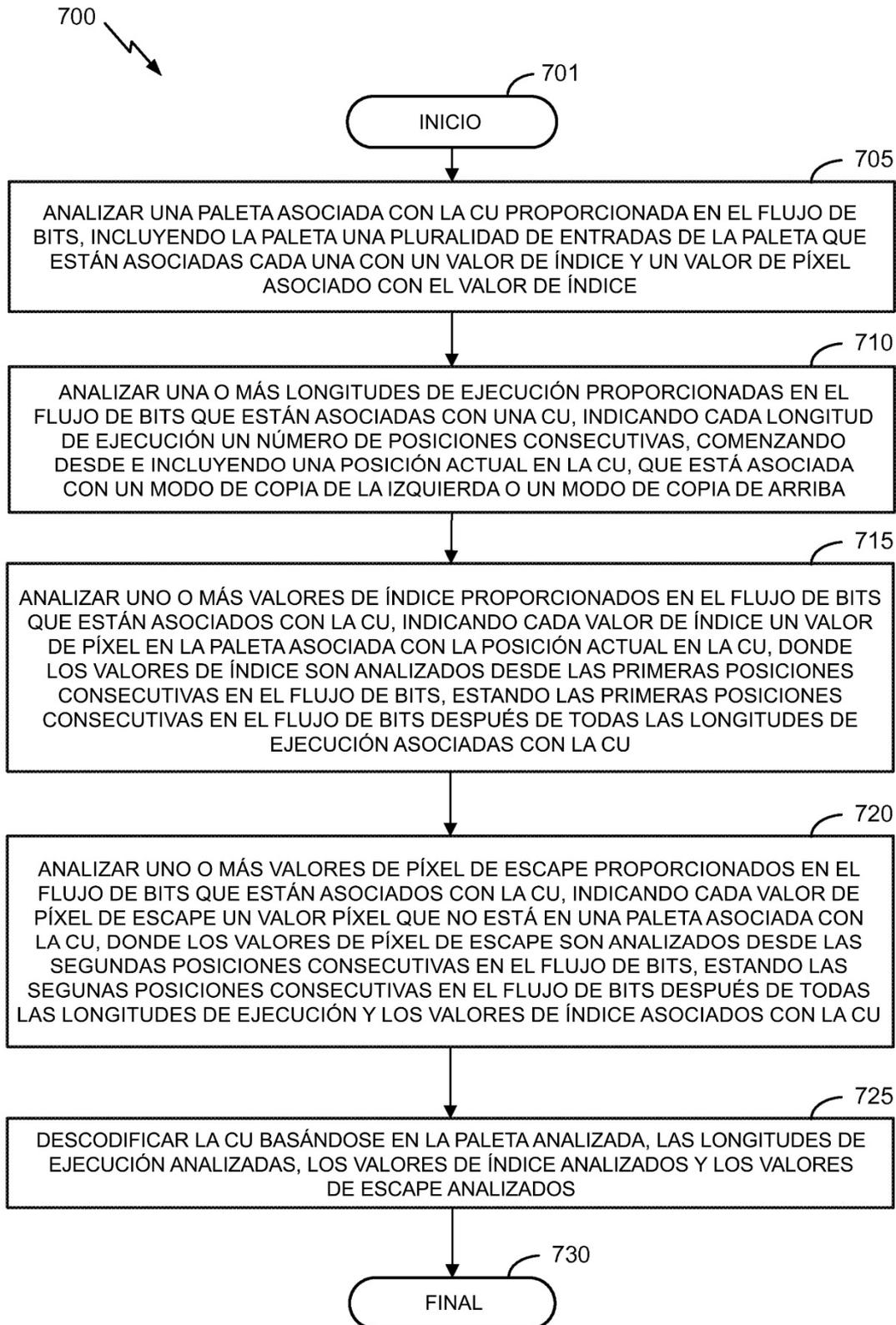


FIG. 7

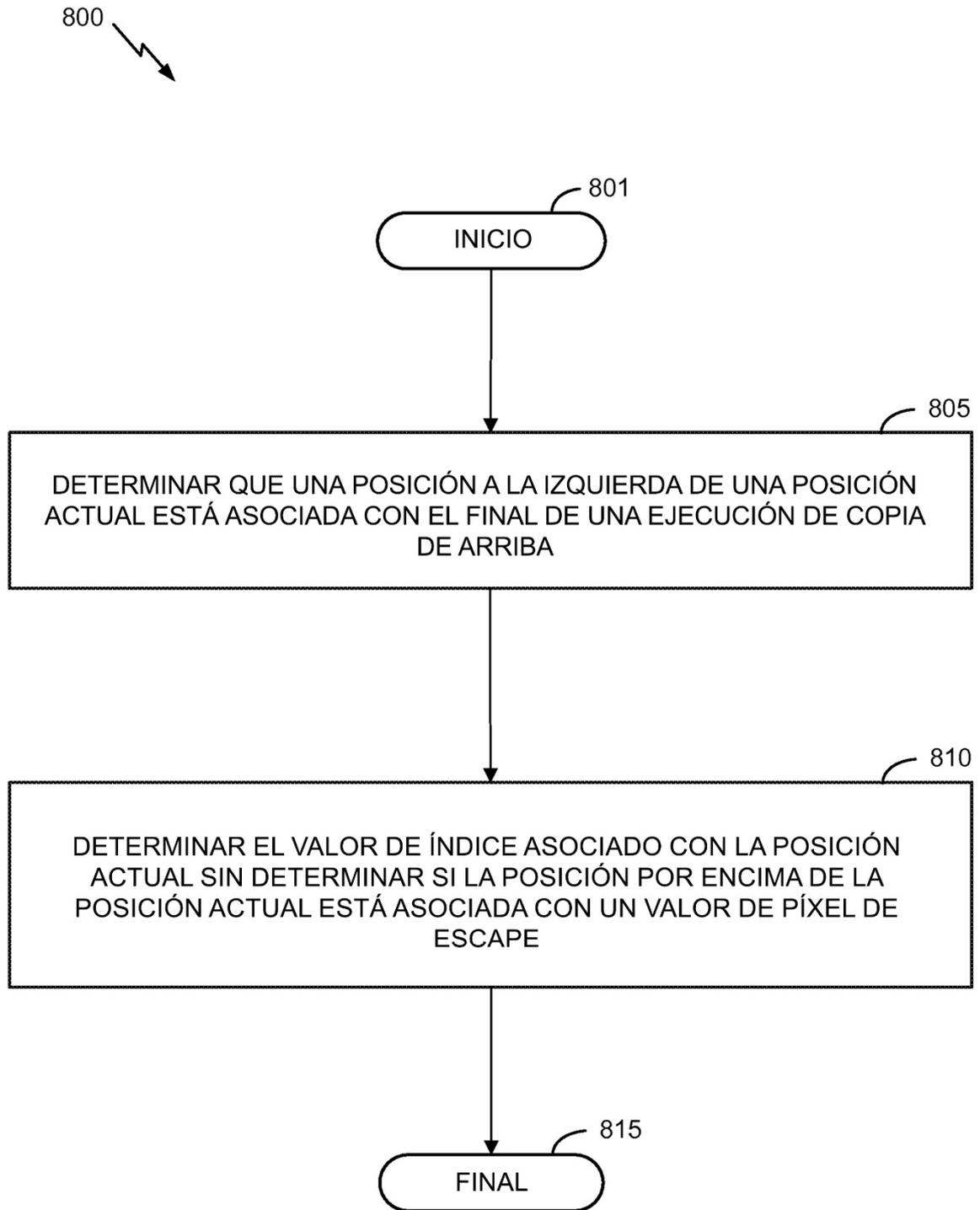


FIG. 8

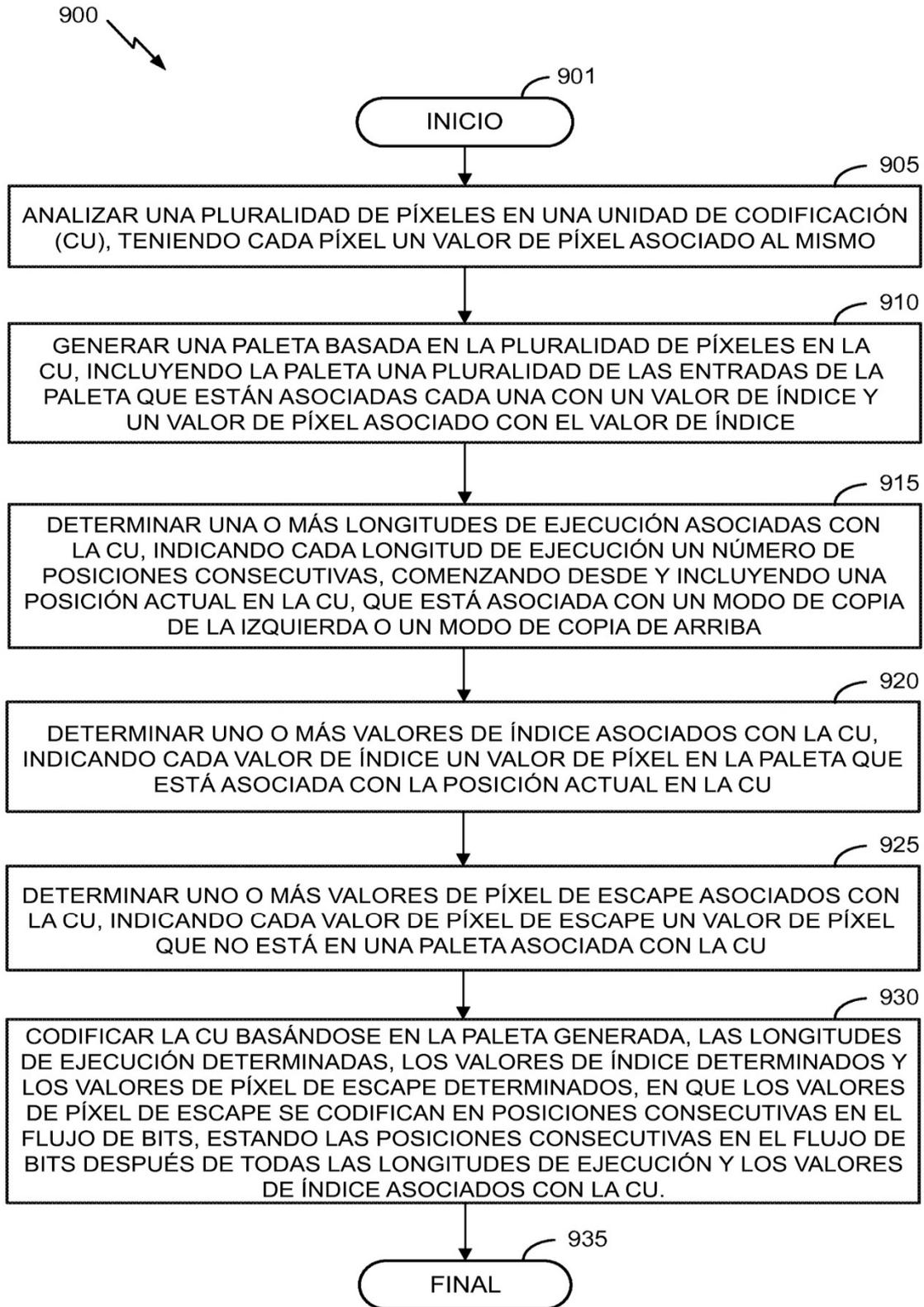


FIG. 9