

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 804 515**

51 Int. Cl.:

H04N 19/12 (2014.01)

H04N 19/124 (2014.01)

H04N 19/186 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **07.10.2015 PCT/US2015/054467**

87 Fecha y número de publicación internacional: **14.04.2016 WO16057652**

96 Fecha de presentación y número de la solicitud europea: **07.10.2015 E 15784853 (2)**

97 Fecha y número de publicación de la concesión europea: **25.03.2020 EP 3205094**

54 Título: **Derivación de Qp y desviación para la transformada de color adaptativa en la codificación de vídeo**

30 Prioridad:

07.10.2014 US 201462061099 P
15.10.2014 US 201462064347 P
06.10.2015 US 201514876594

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
08.02.2021

73 Titular/es:

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

72 Inventor/es:

RAPAKA, KRISHNAKANTH;
ZHANG, LI;
JOSHI, RAJAN LAXMAN y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 804 515 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

Derivación de Qp y desviación para la transformada de color adaptativa en la codificación de vídeo

5 **CAMPO TÉCNICO**

[0001] La presente divulgación se refiere a la codificación de vídeo, tal como la codificación de vídeo o la decodificación de vídeo.

10 **ANTECEDENTES**

[0002] Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluyendo televisiones digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, *tablets*, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos móviles o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, Codificación de vídeo avanzada (AVC), el estándar de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las extensiones de dichos estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente implementando dichas técnicas de compresión de vídeo.

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

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

[0005] XP030013382 ("Text of ISO/IEC 14496-10/FDAM1 AVC Fidelity Range Extensions", 69. REUNIÓN DE MPEG; 19-07-2004 - 23-07-2004; GINEBRA; (GRUPO DE EXPERTOS SOBRE IMÁGENES DE MOVIMIENTO O ISO/IEC JTC1/SC29/WG11), n. ° N6539, 1 de octubre de 2004 (2004-10-01), ISSN: 0000-0351) divulga el proyecto de enmienda de la Extensión de rango de fidelidad de H.264/AVC que incluye codificación 4:4:4 con transformada de color residual.

[0006] El documento US 2005/259730 A1 divulga un procedimiento de codificación de vídeo usando transformadas de color residuales. Un aspecto es la adaptación del QP usando una desviación.

55 **BREVE EXPLICACIÓN**

[0007] La presente divulgación describe técnicas relativas a la determinación de parámetros de cuantificación cuando se usa la codificación de conversión de espacio de color y, además, la presente divulgación describe técnicas para generar y analizar diversos elementos sintácticos, en un flujo de bits codificado de datos de vídeo, usado para señalar los parámetros de cuantificación cuando se use la codificación de conversión de espacio de color.

[0008] En un ejemplo, un procedimiento para decodificar los datos de vídeo incluye determinar, en base a un formato de muestreo de croma para los datos de vídeo, que la transformada de color adaptativa está activada para uno o más bloques de los datos de vídeo; determinar un parámetro de cuantificación para el uno o más bloques

en base a determinar que la transformada de color adaptativa está activada; y decuantificar los coeficientes de transformada en base al parámetro de cuantificación determinado.

[0009] En otro ejemplo, un dispositivo para decodificar datos de vídeo incluye una memoria de datos de vídeo; y uno o más procesadores configurados para determinar, en base a un formato de muestreo de croma para los datos de vídeo, que la transformada de color adaptativa está activada para uno o más bloques de los datos de vídeo; determinar un parámetro de cuantificación para el uno o más bloques en base a determinar que la transformada de color adaptativa está activada; y decuantificar coeficientes de transformada en base al parámetro de cuantificación determinado.

[0010] En otro ejemplo, un aparato para decodificar datos de vídeo incluye medios para determinar, en base a un formato de muestreo de croma para los datos de vídeo, que la transformada de color adaptativa está activada para uno o más bloques de los datos de vídeo; medios para determinar un parámetro de cuantificación para el uno o más bloques en base a determinar que la transformada de color adaptativa está activada; y medios para decuantificar coeficientes de transformada en base al parámetro de cuantificación determinado.

[0011] En otro ejemplo, un medio legible por ordenador almacena instrucciones que, cuando se ejecutan por uno o más procesadores, causan que el uno o más procesadores determinen, en base a un formato de muestreo de croma para los datos de vídeo, que la transformada de color adaptativa está activada para uno o más bloques de los datos de vídeo; determinar un parámetro de cuantificación para el uno o más bloques en base a determinar que la transformada de color adaptativa está activada; y decuantificar coeficientes de transformada en base al parámetro de cuantificación determinado.

[0012] En otro ejemplo, un procedimiento para decodificar datos de vídeo incluye determinar para uno o más bloques de los datos de vídeo que la transformada de color adaptativa está activada; recibir, en un conjunto de parámetros de imágenes, uno o más valores de desviación en respuesta a que la transformada de color adaptativa está activada; determinar un parámetro de cuantificación para un primer componente de color de un primer espacio de color en base al primero del uno o más valores de desviación; y decuantificar los coeficientes de transformada en base al parámetro de cuantificación.

[0013] En otro ejemplo, un dispositivo para decodificar datos de vídeo incluye una memoria de datos de vídeo; uno o más procesadores configurados para determinar para uno o más bloques de los datos de vídeo que la transformada de color adaptativa está activada; recibir, en un conjunto de parámetros de imágenes, uno o más valores de desviación en respuesta a que la transformada de color adaptativa está activada; determinar un parámetro de cuantificación para un primer componente de color de un primer espacio de color en base al primero del uno o más valores de desviación; y decuantificar los coeficientes de transformada en base al parámetro de cuantificación.

[0014] Un aparato para decodificar datos de vídeo incluye medios para determinar para uno o más bloques de los datos de vídeo que la transformada de color adaptativa está activada; medios para recibir, en un conjunto de parámetros de imágenes, uno o más valores de desviación en respuesta a que la transformada de color adaptativa está activada; medios para determinar un parámetro de cuantificación para un primer componente de color de un primer espacio de color en base al primero del uno o más valores de desviación; y medios para decuantificar los coeficientes de transformada en base al parámetro de cuantificación.

[0015] En otro ejemplo, un medio legible por ordenador almacena instrucciones que, cuando se ejecutan por uno o más procesadores, causan que el uno o más procesadores determinen para uno o más bloques de los datos de vídeo que la transformada de color adaptativa está activada; recibir, en un conjunto de parámetros de imágenes, uno o más valores de desviación en respuesta a que la transformada de color adaptativa está activada; determinar un parámetro de cuantificación para un primer componente de color de un primer espacio de color en base al primero del uno o más valores de desviación; y decuantificar los coeficientes de transformada en base al parámetro de cuantificación.

[0016] Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y en la descripción siguiente. La presente invención se define en las reivindicaciones independientes. La divulgación que permite la invención se encuentra en el modo de realización relativo a la sintaxis y a la semántica en el párrafo [0088]. Deben entenderse los ejemplos restantes como ejemplos útiles para entender la invención pero que no representan los modos de realización de la presente invención reivindicada. Estos ejemplos se dan solo para propósitos ilustrativos.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0017]

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y decodificación de vídeo que puede usar las técnicas descritas en la presente divulgación.

Las FIGS. 2A-2C son diagramas conceptuales que ilustran diferentes formatos de muestra para los datos de vídeo.

5 La FIG. 3 es un diagrama conceptual que ilustra una unidad de codificación 16x16 formateada de acuerdo con un formato de muestra 4:2:0.

La FIG. 4 es un diagrama conceptual que ilustra una unidad de codificación 16x16 formateada de acuerdo con un formato de muestra 4:2:2.

10 La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de un bloque de destino y una muestra de referencia para un intrabloque 8x8, de acuerdo con una o más técnicas de la divulgación actual.

15 La FIG. 6 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en la presente divulgación.

La FIG. 7 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar las técnicas descritas en la presente divulgación.

20 La FIG. 8 es un diagrama de bloques que ilustra otro ejemplo de un codificador de vídeo que puede usar técnicas para transformar datos de vídeo que tengan un espacio de color RGB en los datos de vídeo que tengan un segundo espacio de color que use una transformada de color de acuerdo con uno o más aspectos de la presente divulgación.

25 La FIG. 9 es un diagrama de bloques que ilustra otro ejemplo de un decodificador de vídeo que puede usar técnicas para transformar inversamente un bloque de datos de vídeo que tenga un primer espacio de color en un bloque de datos de vídeo que tenga un segundo espacio de color RGB que use una transformada de color inversa de acuerdo con uno o más aspectos de la presente divulgación;

30 la FIG. 10 es un diagrama de flujo que ilustra un procedimiento de decodificación de vídeo de ejemplo de acuerdo con las técnicas de la presente divulgación.

La FIG. 11 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo de ejemplo de acuerdo con las técnicas de la presente divulgación.

35 La FIG. 12 es un diagrama de flujo que ilustra un procedimiento de decodificación de vídeo de ejemplo de acuerdo con las técnicas de la presente divulgación.

40 La FIG. 13 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo a modo de ejemplo de acuerdo con las técnicas de la presente divulgación.

La FIG. 14 es un diagrama de flujo que ilustra un procedimiento de decodificación de vídeo de ejemplo de acuerdo con las técnicas de la presente divulgación.

45 **DESCRIPCIÓN DETALLADA**

[0018] La presente divulgación describe técnicas relativas a derivaciones de parámetros de cuantificación de transformada de color adaptativa. La presente divulgación identifica diversos problemas relativos a la forma en que se usa la derivación de parámetros de cuantificación cuando se usa la transformada de color adaptativa y propone soluciones para abordar estos problemas. La presente divulgación describe técnicas de codificación de vídeo, incluyendo técnicas relativas a extensiones de codificación de contenido de pantalla emergente (SCC) y extensiones de rango (RExt) del estándar de codificación de vídeo de alta eficiencia (HEVC) recientemente finalizado. Las extensiones SCC y de rango se están diseñando para admitir potencialmente una alta profundidad de bits (por ejemplo, más de 8 bits) y/o diferentes formatos de muestreo de croma, tal como 4:4:4, 4:2:2, 4:2:0, 4:0:0, etc., y por lo tanto se están diseñando para incluir nuevas herramientas de codificación no incluidas en el estándar base de HEVC.

[0019] Una de dichas herramientas de codificación es la codificación de conversión de espacio de color. En la codificación de conversión de espacio de color, un codificador de vídeo puede convertir los datos residuales desde un primer espacio de color (por ejemplo, YCbCr) en un segundo espacio de color (por ejemplo, RGB) para lograr una mejor calidad de codificación (por ejemplo, un mejor equilibrio de distorsión de velocidad). Independientemente del espacio de color de los datos residuales, un codificador de vídeo transforma típicamente los datos residuales en coeficientes de transformada y cuantifica los coeficientes de transformada. Un decodificador de vídeo realiza los procesos recíprocos de decuantificar los coeficientes de transformada y transformar de forma inversa los coeficientes de transformada para reconstruir los datos residuales. El codificador de vídeo genera, para su inclusión en el flujo de bits codificado de los datos de vídeo, un parámetro de cuantificación que indica una cantidad de

ajuste a escala usada para cuantificar los niveles de coeficiente de transformada. El decodificador de vídeo analiza el flujo de bits para determinar el parámetro de cuantificación usado por el codificador de vídeo. El parámetro de cuantificación también se puede usar por otros procesos de codificación de vídeo, tal como el filtrado de desbloqueo.

[0020] La presente divulgación describe técnicas relativas a determinar los parámetros de cuantificación cuando se usa la codificación de conversión de espacio de color y, además, la presente divulgación describe técnicas para señalar, desde un codificador a un decodificador como parte de un flujo de bits codificado de datos de vídeo, parámetros de cuantificación cuando se usa la codificación de conversión de espacio de color.

[0021] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede usar las técnicas descritas en la presente divulgación, incluyendo las técnicas para codificar bloques en un modo IBC y técnicas para el procesamiento paralelo. Como se muestra en la FIG. 1, el sistema incluye un dispositivo de origen 12 que genera los datos de vídeo codificados que se van a decodificar en un momento posterior por un dispositivo de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores plegables (es decir, portátiles), *tablets*, decodificadores, auriculares de teléfono tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisiones, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0022] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se vayan a decodificar por medio de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo que pueda mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita los datos de vídeo codificados directamente a un dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0023] De forma alternativa, los datos codificados se pueden emitir desde la interfaz de salida 22 a un dispositivo de almacenamiento 17. De forma similar, una interfaz de entrada puede acceder a los datos codificados del dispositivo de almacenamiento 17. El dispositivo de almacenamiento 17 puede incluir cualquiera de una variedad de medios de almacenamiento de datos accedidos localmente o distribuidos, tales como una unidad de disco duro, discos Blu-ray, DVD, CD-ROM, una memoria flash, una memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 17 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados en el dispositivo de almacenamiento 17 por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para una página web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, modulado de acuerdo con un estándar inalámbrico tal como, por ejemplo, un estándar de red de área local inalámbrica tal como WiFi o un estándar de telecomunicaciones inalámbricas tal como LTE u otro estándar de comunicación móvil), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 17 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0024] Las técnicas de la presente divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, a través de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

[0025] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contenga un vídeo capturado previamente, una interfaz de alimentación de vídeo para recibir un vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos informáticos para generar datos de gráficos informáticos como el vídeo de origen, o una combinación de dichas fuentes. Como ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en la presente divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o por cable.

[0026] El vídeo capturado, precapturado o generado por ordenador se puede codificar por el codificador de vídeo 20. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados se pueden almacenar también (o de forma alternativa) en el dispositivo de almacenamiento 17 para un acceso posterior por el dispositivo de destino 14 u otros dispositivos, para su decodificación y/o reproducción.

[0027] El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 17, pueden incluir una variedad de elementos sintácticos generados por el codificador de vídeo 20 para su uso por un decodificador de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

[0028] El dispositivo de visualización 32 se puede integrar 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 visualiza los datos de vídeo decodificados ante un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0029] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la HEVC, y se pueden ajustar a un modelo de prueba de HEVC (HM). Un borrador de trabajo del estándar de HEVC, denominado "HEVC Working Draft 10" o "HEVC WD10" se describe en Bross *et al.*, "Editors' proposed corrections to HEVC version 1", Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 13.^a reunión, Incheon, Corea, abril de 2013. Una memoria descriptiva del borrador de HEVC está disponible en http://phenix.intevry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-01003-v2.zip. Las técnicas descritas en la presente divulgación también pueden funcionar de acuerdo con las extensiones del estándar de HEVC que están actualmente en desarrollo.

[0030] De forma alternativa o adicionalmente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con otros estándares privados o industriales, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC), o extensiones de dichas normas. Sin embargo, las técnicas de la presente divulgación no están limitadas a ningún estándar de codificación particular. Otros ejemplos de estándares de compresión de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC).

[0031] El diseño de la HEVC se ha finalizado recientemente por el JCT-VC del Grupo de Expertos sobre Codificación de Vídeo (VCEG) del ITU-T y el Grupo de Expertos sobre Imágenes en Movimiento ISO/IEC (MPEG). Las extensiones de rango a la HEVC, denominado HEVC-Rext, también se está desarrollando por el JCT-VC. Un borrador de trabajo (WD) reciente de las extensiones de rango, denominado SHVC WD7 a continuación en el presente documento, está disponible en: http://phenix.intevry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1005-v4.zip.

[0032] La presente divulgación se referirá en general al texto de la memoria descriptiva de HEVC recientemente finalizado como HEVC versión 1 o HEVC base. La memoria descriptiva de la extensión de rango se puede convertir en la HEVC versión 2. Con respecto a muchas herramientas de codificación, tales como la predicción de vectores de movimiento, la HEVC versión 1 y la memoria descriptiva de la extensión de rango son técnicamente similares.

Por lo tanto, siempre que la presente divulgación describa cambios relativos a la HEVC versión 1, los mismos cambios también se pueden aplicar a la memoria descriptiva de la extensión de rango, que incluye en general la memoria descriptiva base de la HEVC, más algunas herramientas de codificación adicionales. Además, se puede suponer en general que los módulos de HEVC versión 1 también se pueden incorporar en un decodificador que implemente la extensión de rango de la HEVC.

[0033] Actualmente se están desarrollando nuevas herramientas de codificación para material de contenido de pantalla, tales como texto y gráficos con movimiento, y que se están contemplando para su inclusión en futuros estándares de codificación de vídeo, incluyendo la futura versión de la HEVC. Estas nuevas herramientas de codificación mejoran potencialmente la eficiencia de codificación para el contenido de la pantalla. Como existe evidencia de que se pueden obtener mejoras significativas en la eficiencia de la codificación explotando las características del contenido de la pantalla con nuevas herramientas de codificación dedicadas, se ha emitido una convocatoria de propuestas (CfP) con el objetivo de desarrollar posibles extensiones futuras del estándar de la HEVC, incluyendo las herramientas específicas para SCC). Se ha invitado a empresas y a las organizaciones a presentar propuestas en respuesta a esta convocatoria. Los casos y los requisitos de uso de esta CfP se describen en el documento MPEG N14174. Durante la 17ª Reunión JCT-VC, se establece el modelo de prueba SCC (SCM). Un borrador de trabajo (WD) reciente de SCC es JCTVC-U1005 y está disponible en http://phenix.intevry.fr/jct/doc_end_user/documents/21_Warsaw/wg11/JCTVC-U1005-v1.zip.

[0034] La presente divulgación contempla que el codificador de vídeo 20 del dispositivo de origen 12 se puede configurar para codificar los datos de vídeo de acuerdo con cualquiera de estos estándares actuales o futuros. De manera similar, la presente divulgación también contempla que el decodificador de vídeo 30 del dispositivo de destino 14 se puede configurar para decodificar datos de vídeo de acuerdo con cualquiera de estos estándares actuales o futuros.

[0035] Aunque no se muestra en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar integrados cada uno con un codificador y un decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, en algunos ejemplos, las unidades MUX-DEMUX se pueden ajustar al protocolo de multiplexor ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0036] Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 se pueden implementar como cualquiera de 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 por campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de la presente divulgación. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 se pueden incluir en uno o más codificadores o decodificadores, de los que cualquiera se puede integrar como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

[0037] Como se presenta anteriormente, el equipo JCT-VC ha terminado recientemente de desarrollar el estándar de HEVC. Los trabajos de estandarización de HEVC se basaron en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo relativos a los dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar hasta treinta y cinco modos de codificación de intrapredicción.

[0038] En la HEVC y otras memorias descriptivas de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también se pueden denominar "tramas". Una imagen puede incluir tres matrices de muestras, indicadas como S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luma. S_{Cb} es una matriz bidimensional de muestras de croma Cb. S_{Cr} es una matriz bidimensional de muestras de croma Cr. Las muestras de croma también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de luma.

[0039] Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades en árbol de codificación (CTU). Cada una de las CTU puede comprender un bloque en árbol de codificación de muestras de luma, dos bloques en árbol de codificación correspondientes de muestras de croma y unas estructuras sintácticas usadas para codificar las muestras de los bloques en árbol de codificación. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una CTU puede comprender un solo bloque en árbol de codificación y unas estructuras sintácticas usadas para codificar las muestras del bloque en árbol de codificación. Un bloque en árbol de codificación puede ser un bloque de muestras $N \times N$. Una CTU

también se puede denominar "bloque en árbol" o "unidad de codificación de máximo tamaño" (LCU). Las CTU de la HEVC pueden ser análogas en términos generales a los macrobloques de otros estándares, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño en particular y puede incluir una o más unidades de codificación (CU). Un segmento puede incluir un número entero de CTU ordenadas consecutivamente en un orden de escaneo de barrido.

[0040] Para generar una CTU codificada, el codificador de vídeo 20 puede realizar de forma recursiva una partición en árbol cuaternario en los bloques en árbol de codificación de una CTU para dividir los bloques en árbol de codificación en bloques de codificación, de ahí el nombre "unidades en árbol de codificación". Un bloque de codificación puede ser un bloque de muestras NxN. Una CU puede comprender un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras Cb y una matriz de muestras Cr y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de codificación.

[0041] El codificador de vídeo 20 puede particionar un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las cuales se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras sintácticas usadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una PU puede comprender un único bloque de predicción y estructuras sintácticas usadas para predecir el bloque de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

[0042] El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa intrapredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a muestras decodificadas de la imagen asociada con la PU. Si el codificador de vídeo 20 usa interpredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a muestras decodificadas de una o más imágenes distintas a la imagen asociada con la PU.

[0043] Después de que el codificador de vídeo 20 genere bloques predictivos de luma, Cb y Cr para una o más PU de una CU, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra del bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques predictivos de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra del bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb de uno de los bloques predictivos de Cb de la CU y una muestra correspondiente del bloque de codificación de Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra del bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

[0044] Además, el codificador de vídeo 20 puede usar una partición en árbol cuaternario para descomponer los bloques residuales de luma, Cb y Cr de una CU en uno o más bloques de transformada de luma, Cb y Cr. Un bloque de transformada es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras a las que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede comprender un bloque de transformada de muestras de luma, dos bloques correspondientes de transformada de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformada. Por tanto, cada TU de una CU puede estar asociada con un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado a la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una TU puede comprender un único bloque de transformada y estructuras sintácticas usadas para transformar las muestras del bloque de transformada.

[0045] El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficientes de luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cb de una TU para generar un bloque de coeficientes de Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

[0046] Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes de Cb o un bloque de coeficientes de Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere, en general, a un proceso en el que coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indiquen los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos sintácticos que indiquen los coeficientes de transformada cuantificados.

[0047] El codificador de vídeo 20 puede emitir un flujo de bits que incluya una secuencia de bits que forme una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades NAL. Una unidad NAL es una estructura sintáctica que contiene una indicación del tipo de datos en la unidad NAL y unos bytes que contienen esos datos en forma de una RBSP entremezclada como sea necesario con bits de prevención de emulación. Cada una de las unidades NAL incluye una cabecera de unidad NAL y encapsula una RBSP. La cabecera de unidad NAL puede incluir un elemento sintáctico que indique un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Una RBSP puede ser una estructura sintáctica que contenga un número entero de bytes que esté encapsulado dentro de una unidad NAL. En algunos casos, una RBSP incluye cero bits.

[0048] Diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad NAL puede encapsular una RBSP para un PPS, un segundo tipo de unidad NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad NAL puede encapsular una RBSP para mensajes SEI, y así sucesivamente. Las unidades NAL que encapsulan RBSP para los datos de codificación de vídeo (a diferencia de RBSP para conjuntos de parámetros y mensajes SEI) se pueden denominar unidades NAL VCL.

[0049] El decodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el decodificador de vídeo 30 puede analizar el flujo de bits para obtener elementos sintácticos a partir del flujo de bits. El decodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base a, al menos en parte, los elementos sintácticos obtenidos a partir del flujo de bits. El procedimiento para reconstruir los datos de vídeo puede ser, en general, recíproco al procedimiento realizado por el codificador de vídeo 20. Además, el decodificador de vídeo 30 puede realizar una cuantificación inversa de los bloques de coeficientes asociados con TU de una CU actual. El decodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes para reconstruir los bloques de transformada asociados a las TU de la CU actual. El decodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el decodificador de vídeo 30 puede reconstruir la imagen.

[0050] Un formato de muestreo de vídeo, que también se puede denominar formato de croma, puede definir la cantidad de muestras de croma incluidas en una CU con respecto a la cantidad de muestras de luma incluidas en una CU. Dependiendo del formato de muestreo de vídeo para los componentes de croma, el tamaño, en términos de número de muestras, de los componentes U y V puede ser el mismo o diferente del tamaño del componente Y. En el estándar de HEVC, un valor llamado chroma_format_idc se define para indicar diferentes formatos de muestreo de los componentes de croma, relativos al componente de luma. En la HEVC, chroma_format_idc se señala en el SPS. La Tabla 1 ilustra la relación entre los valores de chroma_format_idc y los formatos de croma asociados.

Tabla 1: diferentes formatos de croma definidos en la HEVC

chroma_format_idc	formato de croma	SubWidthC	SubHeightC
0	monocroma	-	-
1	4:2:0	2	2
2	4:2:2	2	1
3	4:4:4	1	1

[0051] En la Tabla 1, las variables SubWidthC y SubHeightC se pueden usar para indicar la relación de frecuencia de muestreo horizontal y vertical entre el número de muestras para el componente de luma y el número de muestras para cada componente de croma. En los formatos de croma descritos en la Tabla 1, los dos componentes de croma tienen la misma velocidad de muestreo. Por tanto, en el muestreo 4:2:0, cada una de las dos matrices de croma tiene la mitad de la altura y la mitad del ancho la matriz de luma, mientras que, en el muestreo 4:2:2, cada una de las dos matrices de croma tiene la misma altura y la mitad del ancho de la matriz de luma. En el muestreo 4:4:4, cada una de las dos matrices de croma puede tener la misma altura y anchura que la matriz de luma, o, en algunos casos, los tres planos de color se pueden procesar todos por separado como imágenes muestreadas de monocroma.

[0052] En el ejemplo de la Tabla 1, para el formato 4:2:0, la frecuencia de muestreo para el componente de luma es el doble que la de los componentes de croma para las direcciones horizontal y vertical. Como resultado, para una unidad de codificación formateada de acuerdo con el formato 4:2:0, el ancho y la altura de una matriz de muestras para el componente de luma son el doble de la de cada matriz de muestras para los componentes de croma. De forma similar, para una unidad de codificación formateada de acuerdo con el formato 4:2:2, el ancho de una matriz de muestras para el componente de luma es el doble del ancho de una matriz de muestras para cada componente de croma, pero la altura de la matriz de muestras para el componente de luma es igual a la altura de una matriz de muestras para cada componente de croma. Para una unidad de codificación formateada de acuerdo con el formato 4:4:4, una matriz de muestras para el componente de luma tiene el mismo ancho y alto que una matriz de muestras para cada componente de croma. Cabe destacar que, además del espacio de color YUV, los datos de vídeo se pueden definir de acuerdo con un color de espacio RGB. De esta manera, los formatos de croma descritos en el presente documento se pueden aplicar al espacio de color YUV o RGB. Los formatos de croma RGB se muestrean típicamente de modo que la cantidad de muestras rojas, la cantidad de muestras verdes y la cantidad de muestras azules son iguales. Por tanto, el término "formato de croma 4:4:4" como se usa en el presente documento se puede referir a un espacio de color YUV o a un espacio de color RGB en el que el número de muestras es igual para todos los componentes de color.

[0053] Las FIGS. 2A-2C son diagramas conceptuales que ilustran diferentes formatos de muestra para los datos de vídeo. La FIG. 2A es un diagrama conceptual que ilustra el formato de muestra 4:2:0. Como se ilustra en la FIG. 2B, para el formato de muestra 4:2:0, los componentes de croma tienen un cuarto del tamaño del componente de luma. Por tanto, para una CU formateada de acuerdo con el formato de muestra 4:2:0, hay cuatro muestras de luma para cada muestra de un componente de croma. La FIG. 2B es un diagrama conceptual que ilustra el formato de muestra 4:2:2. Como se ilustra en la FIG. 2B, para el formato de muestra 4:2:2, los componentes de croma tienen la mitad del tamaño del componente de luma. Por tanto, para una CU formateada de acuerdo con el formato de muestra 4:2:2, hay dos muestras de luma por cada muestra de un componente de croma. La FIG. 2C es un diagrama conceptual que ilustra el formato de muestra 4:4:4. Como se ilustra en la FIG. 2C, para el formato de muestra 4:4:4, los componentes de croma son del mismo tamaño que el componente de luma. Por tanto, para una CU formateada de acuerdo con el formato de muestra 4:4:4, hay una muestra de luma para cada muestra de un componente de croma.

[0054] La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de una unidad de codificación 16x16 formateada de acuerdo con un formato de muestra 4:2:0. La FIG. 3 ilustra la posición relativa de las muestras de croma con respecto a las muestras de luma dentro de una CU. Como se describe anteriormente, una CU se define típicamente de acuerdo con el número de muestras de luma horizontales y verticales. Por tanto, como se ilustra en la FIG. 3, una CU 16x16 formateada de acuerdo con el formato de muestra 4:2:0 incluye muestras 16x16 de componentes de luma y muestras 8x8 para cada componente de croma. Además, como se describe anteriormente, una CU se puede particionar en CU más pequeñas. Por ejemplo, la CU ilustrada en la FIG. 3 se puede particionar en cuatro CU 8x8, donde cada CU 8x8 incluya muestras 8x8 para el componente de luma y muestras 4x4 para cada componente de croma.

[0055] La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de una unidad de codificación 16x16 formateada de acuerdo con un formato de muestra 4:2:2. La FIG. 4 ilustra la posición relativa de las muestras de croma con respecto a las muestras de luma dentro de una CU. Como se describe anteriormente, una CU se define típicamente de acuerdo con el número de muestras de luma horizontales y verticales. Por tanto, como se ilustra en la FIG. 4, una CU 16x16 formateada de acuerdo con el formato de muestra 4:2:2 incluye muestras 16x16 de componentes de luma y muestras 8x8 para cada componente de croma. Además, como se describe anteriormente, una CU se puede particionar en CU más pequeñas. Por ejemplo, la CU ilustrada en la FIG. 4 se puede particionar en cuatro CU 8x8, donde cada CU incluye muestras 8x8 para el componente de luma y muestras 4x4 para cada componente de croma.

[0056] De acuerdo con las técnicas descritas en la presente divulgación, se propone una transformada de espacio de color en bucle para las señales residuales (es decir, bloques residuales) para las secuencias en formato de croma 4:4:4; sin embargo, las técnicas no se limitan al formato 4:4:4. El proceso de transformada del espacio de color en bucle transforma las señales de error de predicción (es decir, señales residuales) en formato de croma RGB/YUV en aquellas en un espacio de color subóptimo. La transformada del espacio de color en bucle puede reducir aún más la correlación entre los componentes de color. La matriz de transformada se puede derivar de los valores de muestra de píxeles para cada unidad de codificación mediante una descomposición en valores singulares (SVD). La transformada de color-espacio se puede aplicar al error de predicción de la intramodalidad y la intermodalidad.

[0057] En algunos ejemplos, cuando el proceso de transformada de color-espacio se aplica a la intermodalidad, el residual se convierte primero en un dominio diferente con la matriz de transformada obtenida. Después de la conversión de color-espacio, las etapas de codificación convencional, tales como DCT/DST, cuantificación y codificación por entropía, se realizan en orden.

[0058] Cuando la transformada del espacio de color se aplica a una CU codificada usando un intramodo, la predicción y el bloque actual se convierten en primer lugar en un dominio diferente con la matriz de transformada derivada, respectivamente. Después de la conversión de color-espacio, el residual entre el bloque actual y un predictor se transforma aún más con DCT/DST, se cuantifica y se codifica por entropía.

[0059] Un dispositivo de codificación de vídeo, tal como el codificador de vídeo 20, realiza una operación hacia adelante, donde una matriz de transformada de espacio de color que comprende los valores de conversión a, b, c, d, e, f, g, h, e i se aplica a tres planos G, B y R para derivar valores para los componentes de color P, Q y S como sigue:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix} = \begin{bmatrix} P \\ Q \\ S \end{bmatrix}$$

[0060] Los valores resultantes se pueden recortar dentro del rango de la memoria descriptiva de HEVC, ya que los valores se pueden ampliar hasta $\sqrt{3}$ veces en el peor de los casos. Un dispositivo de decodificación de vídeo, tal como el decodificador de vídeo 30, realiza una operación inversa, donde una matriz de transformada de espacio de color que comprende valores de conversión a^t, b^t, c^t, d^t, e^t, f^t, g^t, h^t, y i^t se aplica a los tres componentes de color P', Q' y R' para derivar los tres planos G', B' y R' como sigue,

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^t \begin{bmatrix} P' \\ Q' \\ S' \end{bmatrix} = \begin{bmatrix} G' \\ B' \\ R' \end{bmatrix}$$

[0061] La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de un bloque de destino y una muestra de referencia para un intrabloque 8x8, de acuerdo con una o más técnicas de la divulgación actual. Se puede derivar una matriz de transformada usando la descomposición en valores singulares (SVD) a partir de los valores de muestra de referencia. Un dispositivo de codificación de vídeo (por ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30) puede usar diferentes muestras de referencia para el intracaso y el intercaso. Para el caso de un bloque intracodificado, se muestran un bloque de destino y muestras de referencia como se muestra en la FIG. 5. En la FIG. 5, el bloque objetivo consta de muestras con trama cruzada 8x8 94, y las muestras de referencia 96 anteriores se muestran en forma de rayas, y las muestras de referencias izquierdas 98 se muestran con puntos.

[0062] Para el caso de un bloque intercodificado, las muestras de referencia para la derivación de la matriz pueden ser las mismas que las muestras de referencia para la compensación de movimiento. Las muestras de referencia en el bloque de predicción de movimiento avanzado (AMP) se pueden submuestrear de modo que se reduzca el número de muestras de referencia. Por ejemplo, el número de muestras de referencia en un bloque 12x16 se reduce 2/3.

[0063] En algunos de los ejemplos anteriores, el proceso de transformada de espacio de color se puede aplicar siempre. Por lo tanto, no es necesario señalar si el proceso de transformada se invoca o no. Además, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden usar el mismo procedimiento para derivar la matriz de transformada para evitar la sobrecarga para señalar la matriz de transformada.

[0064] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar diversas matrices de transformada de espacio de color. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden aplicar diferentes matrices de transformada de espacio de color para diferentes espacios de color. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar un par de matrices de transformada YCbCr para convertir los valores de muestra del espacio de color RGB en el espacio de color YCbCr y viceversa. Las siguientes ecuaciones muestran un conjunto de ejemplos de matrices de transformada YCbCr:

Directa:

$$\begin{bmatrix} S \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0,2126 & 0,7152 & 0,0722 \\ -0,1172 & -0,3942 & 0,5114 \\ 0,5114 & -0,4645 & -0,0469 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Inversa:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1,5397 \\ 1 & -0,1831 & -0,4577 \\ 1 & 1,8142 & 0 \end{bmatrix} \begin{bmatrix} S \\ Cb \\ Cr \end{bmatrix}$$

[0065] En otro ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar un par de matrices de transformada YCoCg para convertir los valores de muestra del espacio de color RGB en el espacio de color YCoCg y viceversa. Las siguientes ecuaciones muestran un conjunto de ejemplo de matrices de transformada YCoCg:

5

Directa:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

10

Inversa:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}$$

15

[0066] Otra matriz de este tipo es la matriz YCoCg-R, que es una versión revisable de la matriz YCoCg-R que ajusta a escala los componentes de Co y Cg en un factor de dos. Al usar una técnica de elevación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden lograr la transformada directa e inversa mediante las siguientes ecuaciones:

Directa:

20

$$Co = R - B$$

$$t = B + [Co/2]$$

$$Cg = G - t$$

25

$$Y = t + [Cg/2]$$

Inversa:

30

$$t = Y - [Cg/2]$$

$$G = Cg + t$$

$$B = t - [Co/2]$$

35

$$R = B + Co$$

[0067] En las ecuaciones y matrices anteriores, las transformaciones directas se pueden realizar antes del proceso de codificación (por ejemplo, mediante un codificador de vídeo). Por el contrario, las transformaciones inversas se pueden realizar después del proceso de decodificación (por ejemplo, por un decodificador de vídeo). Cabe destacar también que el codificador de vídeo 20 incluye un bucle de decodificación para reconstruir la fecha codificada para su uso en la predicción de otros datos de vídeo. En consecuencia, al igual que el decodificador de vídeo 30, el bucle de decodificación del codificador de vídeo 20 también puede realizar las transformaciones inversas.

[0068] Las técnicas de la presente divulgación abordan potencialmente uno o más problemas y, más específicamente, problemas potenciales con la derivación de QP cuando `cu_residual_act_flag` está activado. Por ejemplo, de acuerdo con las soluciones existentes, cuando la transformada de color adaptativa está activa, durante el proceso de escala y transformada, se añade una desviación de QP de -5 para el componente de luma y de croma Cb, y se añade -3 para el componente de croma Cr. Sin embargo, el valor resultante de Qp puede desbordar el rango de Qp permitido. Por ejemplo, en el modelo de prueba actual es posible que el Qp resultante se desborde a -5 cuando el rango permitido por la HEVC esté entre 0 y 51. La presente divulgación también describe técnicas para señalar desviaciones adaptativas de QP cuando la transformada de color adaptativa esté activada.

[0069] A continuación se expone una parte del proceso de ajuste a escala y de transformación.

55

8.6.2 Proceso de ajuste a escala y de transformación

[0070] El parámetro de cuantificación qP se deriva como sigue en el modelo de prueba actual, si `cldx` es igual a 0,

$$qP = Qp'_Y + (cu_residual_act_flag[xTbY][yTbY] ? -5 : 0) \quad (8-261)$$

- De otro modo, si cldx es igual a 1,

$$qP = Qp'_{Cb} + (cu_residual_act_flag[xTbY][yTbY] ? -5 : 0) \quad (8-262)$$

- De otro modo, (cldx es igual a 2),

$$qP = Qp'_{Cr} + (cu_residual_act_flag[xTbY][yTbY] ? -3 : 0) \quad (8-263)$$

donde cldx especifica el componente de color del bloque actual y cu_residual_act_flag especifica si la transformada de color adaptativa se aplica a las muestras residuales de la unidad de codificación actual.

[0071] La presente divulgación describe diversas técnicas que pueden abordar los problemas presentados anteriormente. Cada una de las siguientes técnicas se puede implementar por separado o conjuntamente con una o más de las otras. De acuerdo con una técnica de la presente divulgación, el decodificador de vídeo 30 puede recortar los Qp resultantes de la ecuación 8-261, 8-262, 8-263 de la sección 8.6.2 - proceso de ajuste de escala y de transformación (después de añadir la desviación cuando la transformada de color adaptativa esté activada) a un rango de Qp HEVC que es 0, 51 + QpBdOffset_Y. De acuerdo con otra técnica de la presente divulgación, el codificador de vídeo 20 puede señalar al decodificador de vídeo 30 la desviación de Qp que se aplicará en la sección 8.6.2 (proceso de ajuste a escala y de transformación) cuando la transformada de color adaptativa esté activada. Esta señalización de la desviación de Qp se puede realizar en diversos niveles de granularidad como VPS, SPS, PPS, cabecera de fragmento o su extensión. La desviación de Qp se puede señalar para todos los componentes (luma + croma) o solo algunos de los componentes (por ejemplo, croma)

[0072] De acuerdo con otra técnica de la presente divulgación, el codificador de vídeo 20 puede señalar al decodificador de vídeo 30 un indicador que indique si se aplicará o no la desviación de QP en la sección 8.6.2 (proceso de ajuste a escala y de transformación) cuando la transformada de color adaptativa esté activada. Esta señalización de un indicador se puede hacer en diversos niveles de granularidad como VPS, SPS, PPS, cabecera de fragmento o su extensión. La señalización del indicador se puede señalar para todos los componentes (luma + croma) o solo algunos de los componentes (por ejemplo, croma).

[0073] Las implementaciones de ejemplo de las técnicas introducidas anteriormente se describirán ahora con más detalle. De acuerdo con una técnica de la presente divulgación, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para recortar los Qp dentro del rango de HEVC del Qp. Para mantener el rango de Qp permitido como el mismo que se usa en la HEVC cuando se usa la transformada de color adaptativa, la presente divulgación describe técnicas para recortar el rango de los valores de Qp al del rango de Qp de la HEVC. Los cambios propuestos al modelo de prueba están *escritos en cursiva* a continuación.

8.6.2 Proceso de ajuste a escala y de transformación

[0074]

- Si cldx es igual a 0,

$$qP = Clip3(0, 51 + QpBdOffset_Y, \\ Qp'_Y + (cu_residual_act_flag[xTbY][yTbY] ? -5 : 0))$$

Cuando ChromaArrayType no es igual a 0,

- Si cldx es igual a 1,

$$qP = Clip3(0, 51 + \\ QpBdOffset_C, Qp'_{Cb} + (cu_residual_act_flag[xTbY][yTbY] ? -5 : 0)) \quad (8-262)$$

- De otro modo, (cldx es igual a 2),

$$qP = Clip3(0, 51 + QpBdOffset_C, Qp'_{Cr} \\ + (cu_residual_act_flag[xTbY][yTbY] ? -3 : 0) \quad (8-262)$$

Señalización flexible de la desviación de QP para la transformada de color adaptativa

[0075]

pps_scc_extension () {	Descriptor
...	
if(residual_adaptive_colour_transform_enabled_flag){	
pps_slice_act_qp_offsets_present_flag	
pps_act_y_qp_offset	se(v)
pps_act_cb_qp_offset qp_offset	
pps_act_cr_qp_offset qp_offset offset	
}	
...	
}	
slice_segment_header() {	Descriptor
first_slice_segment_in_pic_flag	u(1)
.....	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
}	
if(chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(pps_slice_act_qp_offsets_present_flag){	
slice_act_y_qp_offset	u(1)
slice_act_cb_qp_offset	
slice_segment_header() {	
slice_act_cr_qp_offset cr_qp_offset offset	
}	
.....	

[0076] Se propone recortar el rango de los valores de QP para el componente de luma y de croma. Si cldx es igual a 0,

5

$$qP = \text{Clip3}(0, 51 + QpBdOffset_Y, Qp'Y + (cu_residual_act_flag[xTbY][yTbY] ? pps_act_y_qp_offset + slice_act_y_qp_offset : 0))$$

Cuando ChromaArrayType no es igual a 0,

10 Si cldx es igual a 1,

$$qP = \text{Clip3}(0, 51 + QpBdOffset_C, Qp'_{Cb} + (cu_residual_act_flag[xTbY][yTbY] ? pps_act_pps_cb_qp_offset + slice_act_cb_qp_offset : 0)) \quad (8-262)$$

Otherwise (cldx is equal to 2),

$$qP = \text{Clip3}(0, 51 + QpBdOffset_C, Qp'_{Cr} + (cu_residual_act_flag[xTbY][yTbY] ? pps_act_pps_cr_qp_offset + slice_act_cr_qp_offset : 0)) \quad (8-263)$$

15

pps_act_y_qp_offset, pps_act_cb_qp_offset y pps_act_cr_qp_offset especifican desviaciones para el parámetro de cuantificación de luma, cb y cr qP derivado en la sección 8.6.2, respectivamente. Los valores de pps_act_y_qp_offset, pps_cb_qp_offset and pps_cr_qp_offset estarán en el intervalo de -12 a +12, ambos inclusive. Cuando ChromaArrayType es igual a 0, pps_act_cb_qp_offset y pps_act_cr_qp_offset no se usan en el proceso de decodificación y los decodificadores deberán ignorar su valor.

20

pps_slice_act_qp_offsets_present_flag igual a 1 especifica que slice_act_y_qp_offset, slice_act_cb_qp_offset, slice_act_cr_qp_offset están presentes en la cabecera de fragmento. pps_slice_act_qp_offsets_present_flag igual a 0 especifica que slice_act_y_qp_offset, slice_act_cb_qpoffset, slice_act_cr_qp_offset no están presentes en la cabecera de fragmento. Cuando no está presente, se infiere que el valor de chroma_qp_offset_enabled_flag es igual a 0.

25

5 **slice_act_y_qp_offset**, **slice_cb_qp_offset** y **slice_cr_qp_offset** especifican desviaciones para el parámetro de cuantificación de luma, cb y cr qP derivado en la sección 8.6.2, respectivamente. Los valores de slice_act_y_qpoffset, slice_cb_qp_offset and slice_cr_qp_offset estarán en el rango de -12 a +12, ambos inclusive. Cuando ChromaArrayType es igual a 0, slice_act_cb_qp_offset y slice_act_cr_qp_offset no se usan en el proceso de decodificación y los decodificadores deberán ignorar su valor.

10 **[0077]** Ahora se describirán técnicas para señalar la presencia de la desviación de QP para la transformada de color adaptativa. Como se analiza en la técnica (1) anteriormente, la desviación de QP negativa fija cuando la transformada de color adaptativa está activada reduce el rango de Qp en los Qp más altos. Por ejemplo, con la definición actual cuando la transformada de color adaptativa está activada, es imposible llegar a QP de más de 46 + QpBdOffset_y, que en algunos escenarios son necesarios para cumplir con la tasa de bits objetivo. En la solución a continuación, se propone señalar un indicador para indicar si se debe añadir o no la desviación de Qp. Los cambios propuestos al modelo de prueba se resaltan en amarillo.

15

slice segment header() {	Descriptor
first_slice_segment_in_pic_flag	u(1)
.....	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
}	
if(chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(residual_adaptive_colour_transform_enabled_flag)	
slice_act_qp_offset_present_flag	u(1)
.....	

20 **slice_act_qp_offset_present_flag** igual a 1 especifica que se aplica una desviación de Qp para las unidades de codificación con cu_residual_act_flag igual a 1. slice_act_qp_offset_present_flag igual a 0 especifica que no se aplica una desviación de Qp a las unidades de codificación con cu_residual_act_flag igual a 1. Cuando no está presente, se infiere que el valor de cu_chroma_qp_offset_enabled_flag es igual a 0.

Si cldx es igual a 0,

$$qP = \text{Clip3}(0, 51 + QpBdOffset_y, Qp'Y + (\text{cu_residual_act_flag}[xTbY][yTbY] \&\& \text{slice_act_qp_offset_present_flag?} -5 : 0))$$

25 Cuando ChromaArrayType no es igual a 0,

Si cldx es igual a 1,

$$qP = \text{Clip3}(0, 51 + QpBdOffset_c, Qp'_{cb} + (\text{cu_residual_act_flag}[xTbY][yTbY] \&\& \text{slice_act_qp_offset_present_flag?} -5 : 0)) \quad (8-262)$$

De otro modo, (cldx es igual a 2),

$$qP = \text{Clip3}(0, 51 + QpBdOffset_c, Qp'_{cr} + (\text{cu_residual_act_flag}[xTbY][yTbY] \&\& \text{slice_act_qp_offset_present_flag?} -3 : 0)) \quad (8-263)$$

35 **[0078]** Ahora se describirá otra implementación de ejemplo de la desviación de QP para la transformada de color adaptativa. La presente divulgación propone lo siguiente:

- 40 a) Señalar el indicador activado de transformada de color adaptativa en el conjunto de parámetros de imágenes en lugar del conjunto de parámetros de secuencia. Potencialmente, esto se beneficia de poder adaptar a nivel de imagen el uso de la transformada de color adaptativa.

- b) Se propone una restricción de flujo de bits para desactivar la transformada de color adaptativa cuando el formato de croma no es 4:4:4. En un ejemplo, se propone aplicar esta restricción en el indicador de activación de transformada de color adaptativa (`residual_adaptive_colour_transform_enabled_flag`)

5

[0079] A continuación se detalla un ejemplo de sintaxis y semántica.

	Descriptor
<code>pps_scc_extension() {</code>	
...	
<code>residual_adaptive_colour_transform_enabled_flag</code>	u(1)
<code>if(residual_adaptive_colour_transform_enabled_flag){</code>	
<code>pps_slice_act_qp_offsets_present_flag</code>	
<code>pps_act_y_qp_offset act_y_qp_offset</code>	se(v)
<code>pps_act_cb_qp_offset</code>	
<code>pps_act_cr_qp_offset act_cr_qp_offset</code>	
<code>}</code>	
...	
<code>}</code>	

	Descriptor
<code>slice_segment_header() {</code>	
<code>first_slice_segment_in_pic_flag</code>	u(1)
.....	
<code>slice_qp_delta</code>	se(v)
<code>if(pps_slice_chroma_qp_offsets_present_flag) {</code>	
<code>slice_cb_qp_offset</code>	se(v)
<code>slice_cr_qp_offset</code>	se(v)
<code>}</code>	
<code>if(chroma_qp_offset_list_enabled_flag)</code>	
<code>co_chroma_qp_offset_enabled_flag</code>	u(1)
<code>if(pps_slice_act_qp_offsets_present_flag){</code>	
<code>slice_act_y_qp_offset</code>	se(v)
<code>slice_act_cb_qp_offset</code>	se(v)
<code>slice_act_cr_qp_offset</code>	se(v)
<code>}</code>	
.....	

10 **residual_adaptive_colour_transform_enabled_flag** igual a 1 especifica que se puede aplicar una transformada de color adaptativa al residual en el proceso de decodificación para las imágenes que se refieren al PPS. **residual_adaptive_colour_transform_enabled_flag** igual a 0 especifica que la transformada de color adaptativa no se aplica al residual para las imágenes que se refieren al PPS. Cuando no está presente, se infiere que el valor de `poc_reset_flag` es igual a 0. Cuando `lchroma_format_idc` no es igual a 3, **residual_adaptive_colour_transform_enabled_flag** será igual a 0.

15

pps_act_y_qp_offset, **pps_act_cb_qp_offset** y **pps_act_cr_qp_offset** especifican desviaciones para el parámetro de cuantificación de luma, cb y cr qP derivado en la sección 8.6.2, respectivamente. Los valores de `pps_act_y_qp_offset`, `pps_cb_qp_offset` y `pps_cr_qp_offset` estarán en el rango de -12 a +12, inclusive. Cuando `ChromaArrayType` es igual a 0, `pps_act_cb_qp_offset` y `pps_act_cr_qp_offset` no se usan en el proceso de decodificación y los decodificadores deberán ignorar su valor.

20

pps_slice_act_qp_offsets_present_flag igual a 1 especifica que `slice_act_y_qp_offset`, `slice_act_cb_qp_offset`, `slice_act_cr_qp_offset` están presentes en la cabecera de fragmento. **pps_slice_act_qp_offsets_present_flag** igual a 0 especifica que `slice_act_y_qp_offset`, `slice_act_cb_qp_offset`, `slice_act_cr_qp_offset` no están presentes en la cabecera de fragmento. Cuando no está presente, se infiere que el valor de `cu_chroma_qp_offset_enabled_flag` es igual a 0.

25

slice_act_y_qp_offset, **slice_cb_qp_offset** y **slice_cr_qp_offset** especifican desviaciones para el parámetro de cuantificación de luma, cb y cr qP derivado en la sección 8.6.2, respectivamente. Los valores de `slice_act_y_qp_offset`, `slice_cb_qp_offset` y `slice_cr_qp_offset` estarán en el rango de -12 a +12, ambos inclusive. Cuando `ChromaArrayType` es igual a 0, `slice_act_cb_qp_offset` y `slice_act_cr_qp_offset` no se usan en el proceso de decodificación y los decodificadores deberán ignorar su valor.

30

35 [0080] La FIG. 6 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas descritas en la presente divulgación. El codificador de vídeo 20 se puede configurar para emitir vídeos a una entidad de posprocesamiento 27. La entidad de posprocesamiento 27 pretende representar un

ejemplo de una entidad de vídeo, tal como un elemento de red consciente de medios (MANE) o un dispositivo de empalme/edición, que puede procesar datos de vídeo codificados desde el codificador de vídeo 20. En algunos casos, la entidad de posprocesamiento 27 puede ser un ejemplo de una entidad de red, tal como un MANE, pero, en otros casos, la entidad de posprocesamiento 27 se puede considerar parte del codificador 20. Por ejemplo, en algunos sistemas de codificación de vídeo, la entidad de posprocesamiento 27 y el codificador de vídeo 20 pueden ser partes de dispositivos separados, mientras que, en otros casos, la funcionalidad descrita con respecto a la entidad de posprocesamiento 27 se puede realizar por el mismo dispositivo que comprende el codificador de vídeo 20. En algunos ejemplos más, la entidad de posprocesamiento 27 se puede implementar como parte del dispositivo de almacenamiento 17 de la FIG. 1.

[0081] El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación y la codificación IMC de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial de un vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) se puede referir a cualquiera de varios modos de compresión de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), se pueden referir a cualquiera de varios modos de compresión de base temporal. Los modos de codificación IMC, como se describe anteriormente, pueden eliminar la redundancia espacial de una trama de datos de vídeo, pero, a diferencia de los intramodos tradicionales, los códigos de codificación IMC se pueden usar para localizar bloques predictivos en un área de búsqueda más grande dentro de la trama y referirse a los bloques predictivos con vectores de desviación, en lugar de basarse en los modos de codificación de intrapredicción.

[0082] En el ejemplo de la FIG. 6, el codificador de vídeo 20 incluye una memoria de datos de vídeo 33, una unidad de partición 35, una unidad de procesamiento de predicción 41, una unidad de filtro 63, una memoria intermedia de imágenes decodificadas 64, un sumador 50, una unidad de procesamiento de transformada 52, una unidad de cuantificación 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 41 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44 y una unidad de procesamiento de intrapredicción 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantificación inversa 58, una unidad de procesamiento de transformada inversa 60 y un sumador 62. La unidad de filtro 63 pretende representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF) y un filtro de desviación adaptativa de muestras (SAO). Aunque la unidad de filtro 63 se muestra en la FIG. 6 siendo un filtro de bucle, en otras configuraciones la unidad de filtro 63 se puede implementar como un filtro posbucle.

[0083] La memoria de datos de vídeo 33 puede almacenar los datos de vídeo que se vayan a codificar mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 33 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18. La memoria intermedia de imágenes decodificadas 64 puede ser una memoria de imágenes de referencia que almacene datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en los modos de intracodificación o intercodificación o IMC. La memoria de datos de vídeo 33 y la memoria intermedia de imágenes decodificadas 64 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como la memoria dinámica de acceso aleatorio (DRAM), incluyendo la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 33 y una memoria intermedia de imágenes decodificadas 64. En diversos ejemplos, la memoria de datos de vídeo 33 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera del chip relativo a esos componentes.

[0084] Como se muestra en la FIG. 6, el codificador de vídeo 20 recibe datos de vídeo y almacena los datos de vídeo en la memoria de datos de vídeo 33. La unidad de partición 35 particiona los datos en bloques de vídeo. Esta partición también puede incluir la partición en fragmentos, mosaicos u otras unidades mayores, así como la partición en bloques de vídeo, por ejemplo, de acuerdo con una estructura en árbol cuaternario de LCU y CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican los bloques de vídeo dentro de un fragmento de vídeo que se va a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 41 puede seleccionar uno de entre una pluralidad de posibles modos de codificación, tal como uno de entre una pluralidad de modos de intracodificación, o uno de entre una pluralidad de modos de intercodificación o uno de entre una pluralidad de modos de codificación IMC, para el bloque de vídeo actual en base a resultados de error (por ejemplo, la tasa de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intra o intercodificado o IMC resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

[0085] La unidad de procesamiento de intrapredicción 46, dentro de la unidad de procesamiento de predicción 41, puede realizar la codificación intrapredictiva del bloque de vídeo actual relativo a uno o más bloques vecinos en la misma trama o fragmento que el bloque actual que vaya a codificarse para proporcionar compresión espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 dentro de la unidad de

procesamiento de predicción 41 realizan la codificación interpredictiva del bloque de vídeo actual relativo a uno o más bloques predictivos en una o más imágenes de referencia, para proporcionar compresión temporal. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 dentro de la unidad de procesamiento de predicción 41 pueden realizar también la codificación IMC del bloque de vídeo actual relativo a uno o más bloques predictivos en la misma imagen para proporcionar compresión temporal.

[0086] La unidad de estimación de movimiento 42 se puede configurar para determinar el modo de interpredicción o modo IMC para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar fragmentos de vídeo en la secuencia como fragmentos P, fragmentos B o fragmentos GPB. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado para propósitos conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar la desviación de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual relativa a un bloque predictivo dentro de una imagen de referencia. En el caso de la codificación IMC, un vector de movimiento, que se puede denominar vector de desviación en IMC, puede indicar la desviación de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual relativa a un bloque predictivo dentro de la trama de vídeo actual.

[0087] Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxel, que se puede determinar mediante una suma de diferencia absoluta (SAD), una suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel subentero de las imágenes de referencia almacenadas en la memoria intermedia de imágenes decodificadas 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento relativa a las posiciones de píxel completo y a las posiciones de píxel fraccionario, y emitir un vector de movimiento con una precisión de píxel fraccionario.

[0088] La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar de entre una primera lista de imágenes de referencia (lista 0) o una segunda lista de imágenes de referencia (lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria intermedia de imágenes decodificadas 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.

[0089] De acuerdo con algunas técnicas de la presente divulgación, cuando se codifica un bloque de vídeo usando un modo IMC, la unidad de estimación de movimiento 42 puede determinar un vector de movimiento, o vector de desviación, para un componente de luma del bloque de vídeo, y determinar un vector de desviación para un componente de croma del bloque de vídeo en base al vector de desviación para el componente de luma. En otro ejemplo, cuando se codifica un bloque de vídeo usando un modo IMC, la unidad de estimación de movimiento 42 puede determinar un vector de movimiento, o vector de desviación, para un componente de croma del bloque de vídeo, y determinar un vector de desviación para un componente de luma del bloque de vídeo en base al vector de desviación para el componente de croma. Por tanto, el codificador de vídeo 20 puede señalar en el flujo de bits solo un vector de desviación, a partir del cual se pueden determinar los vectores de desviación para los componentes de croma y de luma del bloque de vídeo.

[0090] La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar extraer o generar el bloque predictivo en base al vector de movimiento determinado mediante la estimación de movimiento, realizando posiblemente interpolaciones con una precisión de subpíxel. El filtrado de interpolación puede generar muestras de píxeles adicionales a partir de muestras de píxeles conocidas, lo que aumenta potencialmente el número de bloques predictivos candidatos que se pueden usar para codificar un bloque de vídeo. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al cual apunta el vector de movimiento en una de las listas de imágenes de referencia, o en la codificación de IMC, dentro de la imagen que se esté codificando. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se esté codificando, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia tanto de luma como de croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento 44 también puede generar elementos sintácticos asociados con los bloques de vídeo y el fragmento de vídeo para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

[0091] La unidad de procesamiento de intrapredicción 46 puede intrapredicir un bloque actual, como alternativa a la interpredicción y a la IMC realizadas por la unidad de estimación de movimiento 42 y la unidad de

compensación de movimiento 44, como se describe anteriormente. En particular, la unidad de procesamiento de intrapredicción 46 puede determinar un modo de intrapredicción para su uso para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intrapredicción 46 puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pases de codificación separados, y la unidad de procesamiento de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo adecuado de intrapredicción para su uso a partir de los modos probados. Por ejemplo, la unidad de procesamiento de intrapredicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tenga las mejores características de velocidad-distorsión entre los modos modelos probados. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una tasa de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de procesamiento de intrapredicción 46 puede calcular relaciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0092] En cualquier caso, tras seleccionar un modo de intrapredicción para un bloque, la unidad de procesamiento de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indique el modo de intrapredicción seleccionado de acuerdo con las técnicas de la presente divulgación. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices del modo de intrapredicción y una pluralidad de tablas de índices del modo de intrapredicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices del modo de intrapredicción y una tabla de índices del modo de intrapredicción modificada para su uso para cada uno de los contextos.

[0093] Después de que la unidad de procesamiento de predicción 41 genere el bloque predictivo para el bloque de vídeo actual (por ejemplo, por medio de interpredicción o intrapredicción, o IMC), el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo al bloque de vídeo actual. Los datos de vídeo residuales del bloque residual se pueden incluir en una o más TU y aplicarse a la unidad de procesamiento de transformada 52. La unidad de procesamiento de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales usando una transformada, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar. La unidad de procesamiento de transformada 52 puede convertir los datos de vídeo residuales de un dominio de píxel en un dominio de transformada, tal como un dominio de frecuencia.

[0094] La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir aún más la tasa de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos, o con la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar a continuación un escaneo de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar el escaneo.

[0095] Tras la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología o técnica de codificación por entropía. Seguidamente a la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al decodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación por el decodificador de vídeo 30. La unidad de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se esté codificando.

[0096] La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican una cuantificación inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel para su uso posterior como bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros, para su uso en la estimación de movimiento. El filtrado de interpolación puede generar muestras de píxeles adicionales a partir de muestras de píxeles conocidas, lo que aumenta potencialmente el número de bloques predictivos candidatos que se pueden usar para codificar un bloque de vídeo. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, producido por la unidad de compensación de movimiento 44, para producir un bloque de referencia

para su almacenamiento en la memoria intermedia de imágenes decodificadas 64. El bloque de referencia se puede usar por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como bloque de referencia para realizar una interpredicción de un bloque en una trama o imagen de vídeo subsiguiente.

5 **[0097]** La FIG. 7 es un diagrama de bloques que ilustra un decodificador de vídeo 30 de ejemplo que puede implementar las técnicas descritas en la presente divulgación. En el ejemplo de la FIG. 7, el decodificador de vídeo 30 incluye una memoria de datos de vídeo 78, una unidad de decodificación por entropía 80, una unidad de procesamiento de predicción 81, una unidad de cuantificación inversa 86, una unidad de procesamiento de transformada inversa 88, un sumador 90, una unidad de filtro 91 y una memoria intermedia de imágenes codificadas 92. La unidad de procesamiento de predicción 81 incluye la unidad de compensación de movimiento 82 y la unidad de procesamiento de intrapredicción 84. En algunos ejemplos, el decodificador de vídeo 30 puede realizar un pase de decodificación, en general, recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 de la FIG. 6.

15 **[0098]** Durante el proceso de decodificación, el decodificador de vídeo 30 recibe, del codificador de vídeo 20, datos de un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados. El decodificador de vídeo 30 puede recibir los datos de vídeo de la entidad de red 29 y almacenar los datos de vídeo en la memoria de datos de vídeo 78. La memoria de datos de vídeo 78 puede almacenar los datos de vídeo, tales como un flujo de bits de vídeo codificado, que los componentes del decodificador de vídeo 30 vayan a decodificar. Los datos de vídeo almacenados en la memoria de datos de vídeo 78 se pueden obtener, por ejemplo, a partir de un dispositivo de almacenamiento 17, por ejemplo a partir de una fuente de vídeo local, tal como una cámara, por medio de comunicación de red cableada o inalámbrica de datos de vídeo o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 78 puede formar una memoria intermedia de imágenes codificadas que almacene datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar por tanto una memoria de datos de vídeo 78 y una memoria intermedia de imágenes decodificadas 92, aunque se muestran por separado en la FIG. 7. La memoria de datos de vídeo 78 y la memoria intermedia de imágenes decodificadas 92 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como la memoria dinámica de acceso aleatorio (DRAM), incluyendo la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. En diversos ejemplos, la memoria de datos de vídeo 78 puede estar en un chip con otros componentes del decodificador de vídeo 30, o fuera del chip relativo a esos componentes.

35 **[0099]** La entidad de red 29 puede ser, por ejemplo, un servidor, un MANE, un editor/empalmador de vídeo u otro dispositivo de este tipo configurado para implementar una o más de las técnicas descritas anteriormente. La entidad de red 29 puede incluir o no un codificador de vídeo, tal como el codificador de vídeo 20. Algunas de las técnicas descritas en la presente divulgación se pueden implementar por la entidad de red 29 antes de que la entidad de red 29 transmita el flujo de bits de vídeo codificado al decodificador de vídeo 30. En algunos sistemas de decodificación de vídeo, la entidad de red 29 y el decodificador de vídeo 30 pueden ser partes de dispositivos separados, mientras que, en otros casos, la funcionalidad descrita con respecto a la entidad de red 29 se puede realizar por el mismo dispositivo que comprende el decodificador de vídeo 30. La entidad de red 29 puede ser un ejemplo del dispositivo de almacenamiento 17 de la FIG. 1 en algunos casos.

45 **[0100]** La unidad de decodificación por entropía 80 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de decodificación por entropía 80 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 81. El decodificador de vídeo 30 puede recibir los elementos sintácticos a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

50 **[0101]** Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de procesamiento de intrapredicción 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual en base a un modo de intrapredicción señalado y datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B, P o GPB) o cuando un bloque se codifica con IMC, la unidad de compensación de movimiento 82 de la unidad de procesamiento de predicción 81 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual en base a los vectores de movimiento y a otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 80. Para la interpredicción, los bloques predictivos se pueden producir a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de tramas de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción predeterminadas en base a las imágenes de referencia almacenadas en la memoria intermedia de imágenes decodificadas 92. Para la codificación IMC, los bloques predictivos se pueden producir a partir de la misma imagen que el bloque que se esté prediciendo.

65 **[0102]** La unidad de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se esté

5 decodificando. Por ejemplo, la unidad de compensación de movimiento 82 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intra o interpredicción) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, un fragmento B, un fragmento P o un fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, el estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para decodificar los bloques de vídeo en el fragmento de vídeo actual.

10 **[0103]** La unidad de compensación de movimiento 82 también puede realizar la interpolación en base a filtros de interpolación. La unidad de compensación de movimiento 82 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular los valores interpolados para píxeles subenteros de los bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para producir bloques predictivos.

15 **[0104]** De acuerdo con algunas técnicas de la presente divulgación, cuando se codifica un bloque de vídeo usando un modo de IMC, la unidad de compensación de movimiento 82 puede determinar un vector de movimiento, o vector de desviación, para un componente de luma del bloque de vídeo, y determinar un vector de movimiento para un componente de croma del bloque de vídeo en base al vector de movimiento para el componente de luma. En otro ejemplo, al codificar un bloque de vídeo usando un modo IMC, la unidad de compensación de movimiento 82 puede determinar un vector de movimiento, o vector de desviación, para un componente de croma del bloque de vídeo, y determinar un vector de movimiento para un componente de luma del bloque de vídeo en base al vector de movimiento para el componente de croma. Por tanto, el decodificador de vídeo 30 puede recibir en el flujo de bits solo un vector de desviación, a partir del cual se pueden determinar los vectores de desviación para los componentes de croma y de luma del bloque de vídeo.

20 **[0105]** Al decodificar un bloque de vídeo usando el modo IMC, la unidad de compensación de movimiento 82 puede, por ejemplo, modificar un vector de movimiento, denominado vector de desviación para el modo IMC, para que un componente de luma determine un vector de desviación para un componente de croma. La unidad de compensación de movimiento 82 puede, por ejemplo, modificar uno o ambos de un componente x y un componente y del vector de desviación del bloque de luma en base a un formato de muestreo para el bloque de vídeo y en base a una precisión de una posición de subpíxel a la cual apunta el vector de desviación. Por ejemplo, si el bloque de vídeo se codifica usando el formato de muestreo 4:2:2, entonces la unidad de compensación de movimiento 82 solo puede modificar el componente x, no el componente y, del vector de desviación de luma para determinar el vector de desviación para el componente de croma. Como se puede ver en la FIG. 4, en el formato de muestreo 4:2:2, los bloques de croma y los bloques de luma tienen el mismo número de muestras en la dirección vertical, haciendo por tanto la modificación del componente y potencialmente innecesaria. La unidad de compensación de movimiento 82 solo puede modificar el vector de desviación de luma, si, cuando se usa para localizar un bloque predictivo de croma, el vector de desviación de luma apunta a una posición sin una muestra de croma (por ejemplo, en una posición de subpíxel en la muestra de croma de la imagen actual que incluye el bloque actual). Si el vector de desviación de luma, cuando se usa para localizar un bloque predictivo de croma, apunta a una posición donde está presente una muestra de croma, entonces la unidad de compensación de movimiento 82 puede no modificar el vector de desviación de luma.

30 **[0106]** En otros ejemplos, si el bloque de vídeo se codifica usando el formato de muestreo 4:2:0, entonces la unidad de compensación de movimiento 82 puede modificar uno o ambos del componente x y del componente y del vector de desviación de luma para determinar el vector de desviación para el componente de croma. Como se puede ver en la FIG. 3, en el formato de muestreo 4:2:0, los bloques de croma y los bloques de luma tienen un número diferente de muestras tanto en dirección vertical como horizontal. La unidad de compensación de movimiento 82 solo puede modificar el vector de desviación de luma, si, cuando se usa para localizar un bloque predictivo de croma, el vector de desviación de luma apunta a una posición sin una muestra de croma (por ejemplo, en una posición de subpíxel en la muestra de croma de la imagen actual que incluye el bloque actual). Si el vector de desviación de luma, cuando se usa para localizar un bloque predictivo de croma, apunta a una posición donde está presente una muestra de croma, entonces la unidad de compensación de movimiento 82 puede no modificar el vector de desviación de luma.

35 **[0107]** La unidad de compensación de movimiento 82 puede modificar un vector de desviación de luma para generar un vector de movimiento modificado, también denominado vector de desviación modificado. La unidad de compensación de movimiento 82 puede modificar un vector de desviación de luma que, cuando se usa para localizar un bloque predictivo de croma, apunta a una posición de subpíxel de modo que el vector de desviación modificado, usado para el bloque de croma, apunta a una posición de subpíxel de resolución más baja o a una posición de píxel entero. Como ejemplo, un vector de desviación de luma que apunta a una posición de 1/8 de píxel se puede modificar para apuntar a una posición de 1/4 de píxel, un vector de desviación de luma que apunta a una posición de 1/4 de píxel se puede modificar para apuntar a una posición de 1/2 de píxel, etc. En otros ejemplos, la unidad de compensación de movimiento 82 puede modificar el vector de desviación de luma de modo que el vector de desviación modificado siempre apunte a una posición de píxel entero para localizar el bloque de

referencia de croma. La modificación del vector de desviación de luma para que apunte a una posición de subpíxel de resolución más baja o a una posición de píxel entero puede eliminar la necesidad de algún filtrado de interpolación y/o reducir la complejidad de cualquier filtrado de interpolación necesario.

5 **[0108]** Con referencia a las FIGS. 3 y 4, y suponiendo que la muestra superior izquierda se encuentre en la posición (0, 0), un bloque de vídeo tiene muestras de luma en las posiciones x impar y par y en las posiciones y impar y par. En un formato de muestreo 4:4:4, un bloque de vídeo también tiene muestras de croma en las posiciones x impar y par y en las posiciones y impar y par. Por tanto, para un formato de muestreo 4:4:4, la unidad de compensación de movimiento puede usar el mismo vector de desviación para localizar tanto un bloque predictivo de luma como un bloque predictivo de croma. Para un formato de muestreo 4: 2: 2, como se muestra en la FIG. 4, un bloque de vídeo tiene muestras de croma en las posiciones pares e impares pero solo en las posiciones pares x. Por tanto, para el formato de muestreo 4:2:2, si un vector de desviación de luma apunta a una posición x impar, la unidad de compensación de movimiento 82 puede modificar el componente x del vector de desviación de luma para generar un vector de desviación modificado que apunte a una posición x par para que el vector de desviación modificado se pueda usar para localizar el bloque de croma de referencia para el bloque de croma del bloque actual sin necesidad de interpolación. La unidad de compensación de movimiento 82 puede modificar el componente x, por ejemplo, redondeando hacia arriba o hacia abajo a la posición x par más cercana, es decir, cambiando el componente x de modo que apunte a la posición x izquierda más cercana o a la posición x derecha más cercana. Si el vector de desviación de luma ya apunta a una posición x par, entonces no será necesaria ninguna modificación.

25 **[0109]** Para un formato de muestreo 4:2:0, como se muestra en la FIG. 3, un bloque de vídeo tiene muestras de croma solo en las posiciones y pares y solo en las posiciones x pares. Por tanto, para el formato de muestreo 4:2:0, si un vector de desviación de luma apunta a una posición x impar o a una posición y impar, la unidad de compensación de movimiento 82 puede modificar el componente x o el componente y del vector de desviación de luma para generar un vector de desviación modificado que apunte a una posición x par para que el vector de desviación modificado se pueda usar para localizar el bloque de croma de referencia para el bloque de croma del bloque actual sin necesidad de interpolación. La unidad de compensación de movimiento 82 puede modificar el componente x, por ejemplo, redondeando hacia arriba o hacia abajo a la posición x par más cercana, es decir, cambiando el componente x de modo que apunte a la posición x izquierda más cercana o a la posición x derecha más cercana. La unidad de compensación de movimiento 82 puede modificar el componente y, por ejemplo, redondeando hacia arriba o hacia abajo a la posición y par, es decir, cambiando el componente y de modo que apunte a la posición y superior más cercana o a la posición y inferior más cercana. Si el vector de desviación de luma ya apunta a una posición x par y a una posición y par, entonces no será necesaria ninguna modificación.

35 **[0110]** La unidad de cuantificación inversa 86 cuantifica inversamente, es decir, decuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 80. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del fragmento de vídeo para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa que se debería aplicar. La unidad de procesamiento de transformada inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada de entero inversa o un procedimiento de transformada inversa conceptualmente similar, a los coeficientes de transformada para producir bloques residuales en el dominio de píxel.

45 **[0111]** Después de que la unidad de compensación de movimiento 82 genere el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y a otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformada inversa 88 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se pueden usar filtros de bucle (en el bucle de codificación o bien después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad de vídeo. La unidad de filtro 91 pretende representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF) y un filtro de desviación adaptativa de muestras (SAO). Aunque la unidad de filtro 91 se muestra en la FIG. 7 siendo un filtro de bucle, en otras configuraciones la unidad de filtro 91 se puede implementar como un filtro posbucle. Los bloques de vídeo decodificados en una trama o imagen determinada se almacenan a continuación en la memoria intermedia de imágenes decodificadas 92, que almacena imágenes de referencia usadas para una posterior compensación de movimiento. La memoria intermedia de imágenes decodificadas 92 puede ser parte de una memoria que también almacene un vídeo decodificado para su posterior presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1, o puede estar separado de dicha memoria.

60 **[0112]** La FIG. 8 es un diagrama de bloques que ilustra otro codificador de vídeo 21 de ejemplo que puede usar técnicas para transformar datos de vídeo que tengan un espacio de color RGB en bloques de datos de vídeo que tengan un segundo espacio de color usando una transformada de color de acuerdo con uno o más aspectos de la presente divulgación.

[0113] La FIG. 8 ilustra una versión más detallada del codificador de vídeo 20. El codificador de vídeo 21 puede ser un ejemplo del codificador de vídeo 20 (FIG. 2) o del codificador de vídeo 20 (FIG. 1). El ejemplo de la FIG. 8 ilustra dos posibles ejemplos para implementar las técnicas de la presente divulgación. En la primera implementación, el codificador de vídeo 21 transforma adaptativamente un primer bloque de una señal de vídeo de entrada que tiene un primer espacio de color en un segundo bloque que tiene un segundo espacio de color usando una transformada de color de una o más transformadas de color. El segundo ejemplo ilustrado realiza las mismas técnicas, pero realiza la transformación de color en bloques de datos de vídeo residuales, en lugar de en una señal de entrada.

[0114] En el ejemplo de la FIG. 8, el codificador de vídeo 21 se muestra realizando transformadas de color en bloques predictivos y residuales de datos de vídeo en base a los estados de los conmutadores 101, 105, 113, 121. Si los conmutadores 101, 105, 113 y 121 se conmutan a la posición alternativa, el codificador de vídeo 21 se configura para realizar transformadas de color en bloques de datos de vídeo de una señal original que tengan un espacio de color RGB a bloques de datos de vídeo que tengan un segundo espacio de color antes realizar estimaciones de movimiento y predicciones de movimiento, en lugar de transformar bloques de datos de vídeo predictivos y/o residuales.

[0115] Un proceso de ejemplo de realizar transformadas de color en bloques de datos de vídeo residuales como se ilustra en la FIG. 8 se describe ahora con más detalle. En el ejemplo de la FIG. 8, una señal original 100 se pasa a la unidad de procesamiento de predicción 104 (siguiendo la ruta del interruptor 101). La unidad de procesamiento de predicción 104 puede recibir datos de una o más imágenes de referencia desde la memoria de imágenes de referencia 122. La unidad de procesamiento de predicción 104 genera un bloque predictivo de datos de vídeo y combina el bloque predictivo de datos de vídeo de la señal original 100 para generar la señal residual 124. En este ejemplo, el transformador de color adaptativo 106 transforma el bloque predictivo y el bloque residual de datos de vídeo desde un espacio de color RGB en un segundo bloque predictivo y en un segundo bloque residual de vídeo que tiene un segundo espacio de color. En algunos ejemplos, el codificador de vídeo 21 puede seleccionar el segundo espacio de color y la transformada de color en base a una función de coste.

[0116] La unidad de transformada/cuantificación 108 puede realizar una transformada (por ejemplo, una transformada discreta de coseno (DCT) u otro tipo de transformada) en el segundo bloque de vídeo que tiene el segundo espacio de color. Además, la unidad de transformada/cuantificación 108 puede cuantificar el segundo bloque de vídeo (es decir, el bloque de vídeo residual transformado). El codificador por entropía 110 puede codificar por entropía el bloque de vídeo residual cuantificado. El codificador por entropía 110 puede emitir un flujo de bits que incluya el bloque de vídeo residual cuantificado para la decodificación por un decodificador de vídeo, por ejemplo, el decodificador de vídeo 30.

[0117] La unidad de decuantificación/transformada inversa 112 también puede recibir los bloques de vídeo residuales y/o coeficientes transformados cuantificados, y puede transformar inversamente y decuantificar los bloques de vídeo residuales y el coeficiente transformado. Los bloques de vídeo decuantificados e inversamente transformados aún pueden tener el segundo espacio de color en este punto. El resultado de decuantificación/transformada inversa es la señal residual reconstruida 126. El transformador de color adaptativo inverso 114 puede transformar de forma inversa el color de la señal residual reconstruida en base a la transformada de color inversa asociada con la transformada realizada por el transformador de color adaptativo 106. El coeficiente transformado de color adaptativo de forma inversa resultante y/o los bloques de vídeo residuales pueden tener un espacio de color RGB en este punto.

[0118] Después de la aplicación de una transformación de color inversa a un bloque de vídeo residual, el compensador de predicción 116 se puede añadir de nuevo en un bloque predictivo al bloque de vídeo residual. El filtro de desbloqueo 118 puede desbloquear el bloque resultante. El filtro SAO 120 puede realizar el filtrado SAO. La memoria de imágenes de referencia 122 puede almacenar entonces la señal reconstruida resultante 128 para su uso futuro.

[0119] Para transformar de color un bloque de vídeo de una señal de entrada (es decir, datos de vídeo no codificados), en lugar de un bloque de datos de vídeo residuales, el interruptor 101 se gira a la posición alternativa, y el color del transformador adaptativo 102 transforma el bloque de vídeo de entrada desde un bloque de vídeo que tiene un espacio de color RGB en un segundo espacio de color que usa una transformada de color de la una o más transformadas de color. La predicción con la unidad de procesamiento de predicción 104 procede como se describe anteriormente, pero el resultado se puede alimentar a la unidad de transformada/cuantificación 108 directamente porque el interruptor 105 está en la posición alternativa (en comparación con la posición ilustrada en la FIG. 8), en lugar de transformarse de color por el transformador de color adaptativo 106.

[0120] La unidad de transformada/cuantificación 108, el codificador por entropía 110 y la unidad de decuantificación/transformada inversa 112 pueden funcionar como se describe anteriormente con respecto al color que transforma un bloque de vídeo residual, y se genera la señal reconstruida 126, y también está en el segundo espacio de color. La señal reconstruida 126 se alimenta al compensador de predicción 116 por medio del interruptor 113. El interruptor 113 está en la posición alternativa a la posición ilustrada en la FIG. 8, y el transformador de color

adaptativo inverso 114 se deriva. El compensador de predicción 116, el filtro de desbloqueo 118 y el filtro SAO 120 pueden funcionar como se describe anteriormente con respecto al color que transforma un bloque de vídeo residual para producir la señal reconstruida 128. Sin embargo, a diferencia de la señal reconstruida 128 descrita anteriormente, en este ejemplo, un bloque de la señal reconstruida 128 todavía puede tener el segundo espacio de color, en lugar del espacio de color RGB.

[0121] La señal reconstruida 128 se puede alimentar al transformador de color adaptativo inverso 130 por medio del interruptor 121, que está en la posición alternativa a la ilustrada en la FIG. 8. El transformador de color adaptativo inverso 130 puede transformar de forma inversa los bloques de color de la señal reconstruida 128 en bloques que tengan un espacio de color RGB, y la memoria de imágenes de referencia 122 puede almacenar los bloques como bloques de una imagen de referencia para su referencia futura.

[0122] Como se describe anteriormente, el codificador de vídeo 21 puede seleccionar una transformada del uno o más espacios de color para transformar un primer bloque de los datos de vídeo que tengan un espacio de color RGB, en un segundo espacio de color. En algunos ejemplos, el codificador de vídeo 21 selecciona la transformada de color de forma adaptativa calculando los costes de distorsión de velocidad asociados con cada una de las transformadas de color. Por ejemplo, el codificador de vídeo 21 puede seleccionar la transformada de color de la pluralidad de transformadas de color que tiene el coste de distorsión asociado más bajo para una CU o un bloque de una CU. El codificador de vídeo 21 puede señalar un elemento sintáctico de índice u otros datos de sintaxis que indiquen la transformada de color seleccionada que tiene el menor coste de distorsión asociado.

[0123] En algunos ejemplos, el codificador de vídeo 21 puede usar una función de coste lagrangiano que explica el equilibrio entre la tasa de bits (por ejemplo, la compresión lograda) por la transformada de color, así como la distorsión (por ejemplo, la pérdida de fidelidad) asociada con la transformada de color. En algunos ejemplos, el coste lagrangiano corresponde a $L = D + \lambda R$, donde L es el coste lagrangiano, D es la distorsión, λ es un multiplicador de Lagrange, y R es la tasa de bits. En algunos ejemplos, el codificador de vídeo 21 puede señalar un elemento sintáctico de índice que indique la transformada de color de la pluralidad de transformadas de color que minimiza el coste lagrangiano.

[0124] En algunas aplicaciones o configuraciones de codificación de vídeo de alto rendimiento o alta fidelidad, la distorsión debe minimizarse por encima de minimizar la tasa de bits. En dichos casos, al transformar datos de vídeo desde un espacio de color RGB en un segundo espacio de color, el codificador de vídeo 21 puede seleccionar la transformada de color y el espacio de color que da como resultado la menor distorsión. El codificador de vídeo 21 puede señalar un elemento sintáctico de índice que indique la transformada de color seleccionada o el espacio de color que dé como resultado la menor distorsión.

[0125] En algunos otros casos, el codificador de vídeo 21 puede calcular el coste de transformar bloques de un espacio de color RGB en un segundo espacio de color en base a la correlación entre cada uno de los componentes de color del bloque de datos de vídeo RGB y los componentes de color del bloque del segundo espacio de color. La transformada de color que tiene el coste asociado más bajo puede ser la transformada de color que tenga componentes de color que estén más estrechamente correlacionados con los componentes de color RGB de la señal de entrada. El codificador de vídeo 21 puede señalar un elemento sintáctico de índice que indique la transformada de color seleccionada que tenga la mayor correlación entre sus componentes de color y los componentes de color RGB.

[0126] Debe reconocerse que, en algunos casos, el codificador de vídeo 21 puede seleccionar diferentes transformadas de color para diferentes CU, LCU, CTU u otras unidades de datos de vídeo. Es decir, para una sola imagen, el codificador de vídeo 21 puede seleccionar diferentes transformadas de color asociadas con diferentes espacios de color. Seleccionar múltiples transformadas de color diferentes puede optimizar mejor la eficiencia de la codificación y reducir la distorsión de la velocidad. Para indicar qué transformada de las transformadas múltiples ha seleccionado el codificador de vídeo 21 para el bloque actual, el codificador de vídeo 21 puede señalar un valor de índice correspondiente a la transformada de color seleccionada. El codificador de vídeo 21 puede señalar el valor de índice en uno o más del primer bloque de vídeo, una CTU, una CU, una PU y una TU.

[0127] Sin embargo, en algunos casos, el codificador de vídeo 21 puede determinar una sola transformada de color que se aplicará a uno o a una pluralidad de bloques, o a una secuencia de imágenes codificadas, denominada CVS. En el caso de que solo se seleccione una transformada de color, para cada bloque, el codificador de vídeo 21 puede señalar un elemento sintáctico de indicador. Un valor del elemento sintáctico de indicador puede indicar que el codificador de vídeo 21 ha aplicado la transformada única al bloque actual o a todas las imágenes en el CVS. El otro valor del elemento sintáctico de indicador indica que no se ha aplicado ninguna transformada al bloque actual. El codificador de vídeo 21 puede determinar si se aplica o no la transformada de color a cada uno de los bloques de la imagen de forma individual, por ejemplo, usando los criterios basados en costes descritos anteriormente.

[0128] En algunos ejemplos, el codificador de vídeo 21 determina si se aplica una transformada de color predefinida de la pluralidad de transformadas de color inversas a cada uno de la pluralidad de bloques. Por ejemplo,

el codificador de vídeo 21 y el decodificador de vídeo 31 pueden usar una transformada de color predefinida predeterminada/transformada de color inversa. En respuesta a determinar aplicar la transformada de color predefinida a cada uno de la pluralidad de bloques, el codificador de vídeo 21 puede transformar cada uno de la pluralidad de bloques usando la transformada de color predefinida sin decodificar datos que indiquen que la transformada de color predefinida se ha aplicado a cada uno de los bloques de pluralidad de datos de vídeo.

[0129] De manera recíproca, el decodificador de vídeo 31 se puede configurar para determinar si se aplica una transformada de color inversa predefinida de la pluralidad de transformadas de color inversa a cada uno de la pluralidad de bloques. En respuesta a determinar aplicar la transformada de color inversa predefinida a cada uno de la pluralidad de bloques, el decodificador de vídeo 31 puede transformar de forma inversa cada uno de la pluralidad de bloques usando la transformada de color predefinida sin decodificar datos que indiquen que la transformada de color predefinida se ha aplicado a cada uno de los bloques de pluralidad de datos de vídeo

[0130] Las transformadas de color de la presente divulgación pueden incluir, pero no se limitan necesariamente a, una transformada de identidad, una transformada diferencial, una transformada diferencial ponderada, una DCT, una transformada YCbCr, una transformada YCgCo y una transformada YCgCo-R al bloque de datos de vídeo Un codificador de vídeo configurado de acuerdo con las técnicas de la presente divulgación, tal como el codificador de vídeo 21, puede aplicar una o más de estas transformadas y/o sus inversas, así como otras transformadas, tales como las transformadas a/desde Adobe RGB, sRGB, scRGB Rec. 709, rec. 2020, Adobe Wide Gamut RGB, ProPhoto RGB, CMYK, Pantone, YIQ, YDbDr, YPbPr, xvYCC, ITU BT.601, ITU BT.709, HSV y otros espacios de color, espacios de color y/o formatos de submuestreo de croma no descritos específicamente en el presente documento.

[0131] Para aplicar una transformada de color a un bloque de datos de vídeo que tenga un espacio de color RGB, el codificador de vídeo 21 puede multiplicar una matriz 3 x 1 que comprenda los componentes de color rojo, verde y azul de un píxel RGB con una matriz de transformada de color. El resultado de la multiplicación es un píxel que tiene un segundo espacio de color. El codificador de vídeo puede aplicar la matriz de transformada de color a cada píxel del bloque de vídeo para producir un segundo bloque de píxeles en un segundo espacio de color. Ahora se describen diversas transformadas de color con más detalle.

[0132] En algunos ejemplos, el codificador de vídeo 21 puede aplicar una matriz de transformada de identidad o una matriz de transformada de identidad inversa. La matriz de transformada de identidad comprende:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

y la matriz de transformada inversa, que el decodificador de vídeo 30 puede aplicar, comprende:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Cuando un codificador de vídeo aplica la transformada de identidad, el valor de píxel resultante es idéntico al valor de píxel de entrada, es decir, aplicar la transformada de identidad es equivalente a no aplicar una transformada de color en absoluto. El codificador de vídeo 21 puede seleccionar la transformada de identidad cuando se requiera mantener el espacio de color RGB de los bloques de vídeo.

[0133] En otro ejemplo, el codificador de vídeo 21 puede aplicar una matriz de transformada diferencial. La matriz de transformada diferencial comprende:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix}.$$

El decodificador de vídeo 30 puede aplicar una matriz diferencial inversa recíproca, que comprende:

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

[0134] En otro ejemplo, el codificador de vídeo 21 se puede configurar para aplicar una transformada diferencial ponderada o una transformada diferencial ponderada inversa. La matriz de transformada diferencial ponderada comprende:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -\alpha_1 & 1 \\ 1 & -\alpha_2 & 0 \end{bmatrix},$$

y la matriz diferencial ponderada inversa, cuyo decodificador de vídeo 31 puede aplicar, comprende:

$$\begin{bmatrix} \alpha_2 & 0 & 1 \\ 1 & 0 & 0 \\ \alpha_1 & 1 & 0 \end{bmatrix}.$$

[0135] En las transformadas diferenciales ponderadas, α_1 y α_2 son parámetros que un codificador de vídeo puede ajustar. En algunos ejemplos, el codificador de vídeo 20 puede calcular los parámetros α_1 y α_2 de acuerdo con las siguientes ecuaciones:

$$\alpha_1 = \text{cov}(G, B) / \text{var}(G),$$

y

$$\alpha_2 = \text{cov}(G, R) / \text{var}(G).$$

El codificador de vídeo 21 puede señalar los valores de α_1 y α_2 en el flujo de bits de vídeo codificado en varios ejemplos.

[0136] En estas ecuaciones, R corresponde a un canal de color rojo, G corresponde a un canal de color verde y B corresponde a un canal de color azul del espacio de color RGB. En las ecuaciones de transformada diferencial, "cov ()" es la función de covarianza, y "var ()" es la función de varianza.

[0137] Para determinar los valores de R, G y B, un codificador o decodificador puede usar un conjunto de píxeles de referencia para garantizar que las funciones de covarianza y varianza tengan el mismo resultado o peso cuando el codificador o el decodificador lo calcule. En algunos ejemplos, los píxeles de referencia particulares se pueden señalar en el flujo de bits de vídeo codificado (por ejemplo, como elementos sintácticos en un flujo de bits de vídeo codificado). En otros ejemplos, el codificador y el decodificador se pueden preprogramar para usar determinados píxeles de referencia.

[0138] En algunos ejemplos, el codificador de vídeo 21 puede restringir los valores de α_1 y α_2 al transformar bloques que usen la transformada diferencial. El codificador de vídeo puede restringir los valores de α_1 y α_2 a un conjunto de enteros o números diádicos, por ejemplo, 1/2, 1/4, 1/8, etc. En otros ejemplos, un codificador de vídeo puede restringir α_1 y α_2 a valores de una fracción que tenga un número diádico, por ejemplo, 1/8, 2/8, 3/8, ..., 8/8. Un número diádico o fracción diádica es un número racional que tiene un denominador que es una potencia de dos, y donde el numerador es un número entero. Restringir los valores de α_1 y α_2 puede mejorar la eficiencia del flujo de bits de la codificación α_1 y α_2 .

[0139] En otros ejemplos, el codificador de vídeo 21 se puede configurar para transformar un bloque que tenga un espacio de color RGB para generar un segundo bloque, usando una transformada DCT. El DCT transforma las muestras de un bloque para expresar las muestras como una suma de sinusoides de diferentes frecuencias y amplitudes. Una transformada DCT o una transformada inversa puede transformar píxeles hacia y desde una secuencia finita de puntos de datos en términos de una suma de funciones de coseno. La matriz de transformada DCT corresponde a:

$$\begin{bmatrix} 0,5774 & 0,5774 & 0,5774 \\ 0,7071 & 0 & -0,7071 \\ 0,4082 & -0,8156 & 0,4082 \end{bmatrix}$$

De manera recíproca, el decodificador de vídeo 31 se puede configurar para aplicar una transformada inversa a los bloques transformados usando el DCT para revertir los bloques de vuelta a las muestras originales. La matriz de transformada DCT inversa corresponde a:

$$\begin{bmatrix} 0,5774 & 0,7071 & 0,4082 \\ 0,5774 & 0 & -0,8156 \\ 0,5774 & -0,7071 & 0,4082 \end{bmatrix}$$

[0140] El codificador de vídeo 21 también puede aplicar una transformada YCbCr a un bloque que tenga un espacio de color RGB para producir un bloque que tenga un espacio de color YCbCr. Como se describe anteriormente, el espacio de color YCbCr incluye un componente de luma (Y), así como componentes de croma azul (Cb) y croma rojo (Cr). La matriz de transformada YCbCr puede corresponder a:

$$\begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix}$$

5 **[0141]** El decodificador de vídeo 31 se puede configurar para aplicar una transformada inversa YCbCr para transformar un bloque que tenga un espacio de color YCbCr en un bloque que tenga un espacio de color RGB. La matriz de transformada inversa YCbCr puede corresponder a:

$$\begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix}$$

10 **[0142]** El codificador de vídeo 21 también puede aplicar una transformada YCgCo a un bloque que tenga un espacio de color RGB para producir un bloque que tenga un espacio de color YCgCo. Un espacio de color YCgCo incluye un componente de luma (Y), así como componentes de croma verde (Cg) y de croma naranja (Co). La matriz de transformada YCgCo puede corresponder a:

15
$$\begin{bmatrix} 0,2126 & 0,7152 & 0,0722 \\ -0,1172 & -0,3942 & 0,5114 \\ 0,5114 & -0,4645 & -0,0469 \end{bmatrix}$$

20 **[0143]** El decodificador de vídeo 31 se puede configurar para aplicar una transformada YCgCo inversa para convertir un bloque que tenga un espacio de color YCgCo en un bloque que tenga un espacio de color RGB. La matriz de transformada inversa YCgCo puede corresponder a:

$$\begin{bmatrix} 1 & 0 & 1,5397 \\ 1 & -0,1831 & -0,4577 \\ 1 & 1,8142 & 0 \end{bmatrix}$$

25 **[0144]** El codificador de vídeo 21 también se puede configurar para aplicar una transformada YCgCo-R a un bloque que tenga un espacio de color RGB para producir un bloque que tenga un espacio de color YCgCo-R. El espacio de color YCgCo-R incluye un componente de luma (Y), así como componentes de croma verde (Cg) y de croma naranja (Co). Sin embargo, a diferencia de la transformada YCgCo descrita anteriormente, la transformada YCgCo-R es reversible, por ejemplo, la transformada YCgCo-R puede no producir ninguna distorsión, por ejemplo debido a errores de redondeo.

30 **[0145]** La matriz de transformada YCbCr puede corresponder a:

$$Co = R - B$$

$$t = B + [Co/2]$$

35

$$Cg = G - t$$

$$Y = t + [Cg/2]$$

40 El decodificador de vídeo 31 se puede configurar para aplicar una transformada YCgCo-R inversa. La transformada inversa YCgCo-R transforma de forma inversa los bloques que tienen un espacio de color YCgCo-R en bloques que tienen un espacio de color RGB. La matriz de transformada YCgCo-R inversa puede corresponder a:

$$t = Y - [Cg/2]$$

45

$$G = Cg + t$$

$$B = t - [Co/2]$$

50

$$R = B + Co$$

55 **[0146]** Para aplicar cualquiera de las transformadas de color descritas en el presente documento, el codificador de vídeo 21 puede implementar un esquema de elevación que tenga parámetros flexibles. Un esquema de elevación es una técnica de descomposición de una transformada de ondículas discreta en una secuencia finita de pasos de filtrado simples, denominados pasos de elevación o estructuras de escalera. El codificador de vídeo 21 puede señalar los parámetros en el flujo de bits de vídeo codificado, o el codificador de vídeo 21 puede derivar

los parámetros y puede derivar los parámetros de la misma manera. Un ejemplo de un esquema de elevación es el siguiente:

$$\begin{aligned}
 R' &= R + [aB] \\
 B' &= B + [bR'] \\
 G' &= G + [cB'] \\
 R'' &= R' + [dG']
 \end{aligned}$$

donde a , b , c y d son parámetros enteros como se describe anteriormente. En este esquema de elevación, R , G y B son canales o muestras de color rojo, verde y azul, respectivamente. Al igual que con los parámetros α descritos anteriormente con respecto a la transformada diferencial ponderada, los valores de a , b , c y d se pueden restringir o limitar, por ejemplo, los signos solo pueden ser positivos o negativos. En algunos casos, puede haber pasos adicionales en el esquema de elevación, tales como:

$$\begin{aligned}
 R''' &= [eR'' + f] \\
 B'' &= [gB' + h], \\
 G'' &= [iG' + j]
 \end{aligned}$$

donde f , g , h , i , y j son parámetros. Cuando se usa el esquema de elevación, así como en otros ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden normalizar la profundidad de salida de los tres componentes, R''' , B'' y G'' que se pueden normalizar dentro de una profundidad de bits predeterminada, que puede no ser necesariamente la misma para cada componente.

[0147] La FIG. 9 es un diagrama de bloques que ilustra otro decodificador de vídeo 31 de ejemplo que puede usar técnicas para transformar inversamente los datos de vídeo que tengan un primer espacio de color en datos de vídeo que tengan un segundo espacio de color RGB usando una transformada de color inversa de acuerdo con uno o más aspectos de la presente divulgación.

[0148] La FIG. 9 ilustra una versión más detallada del decodificador de vídeo 31 relativo al decodificador de vídeo 30 de la FIG. 1 y la FIG. 7. De hecho, en algunos ejemplos, el decodificador de vídeo 31 se puede considerar un ejemplo más específico del decodificador de vídeo 30 (FIG. 7) y/o el decodificador de vídeo 30 (FIG. 1). El ejemplo de la FIG. 9 ilustra dos posibles ejemplos para implementar las técnicas de la presente divulgación. En la primera implementación, el decodificador de vídeo 31 transforma adaptativamente de forma inversa un bloque de una señal de vídeo de entrada desde un primer espacio de color (por ejemplo, un espacio de color no RGB) en un segundo bloque que tiene un segundo espacio de color RGB usando una transformada de color inversa de una pluralidad de transformadas de color inversas. El segundo ejemplo ilustrado realiza las mismas técnicas, pero realiza la transformación de color inversa en bloques de datos de vídeo residuales, en lugar de en una señal de entrada.

[0149] En el ejemplo de la FIG. 9, el decodificador de vídeo 31 se muestra realizando transformadas de color inversas en bloques de ejemplo de datos de vídeo residuales debido a la forma en que los conmutadores 145, y 156 se conmutan actualmente. Si los conmutadores 145 y 156 se conmutan a la posición alternativa, el decodificador de vídeo 31 se configura para transformar de color inversamente los bloques de datos de entrada de vídeo que tengan una primera representación en un bloque de datos de vídeo que tengan un segundo espacio de color RGB, en lugar de transformar de forma inversa bloques residuales de datos de vídeo.

[0150] El proceso de realizar transformadas de color inversas en bloques de datos de vídeo residuales como se ilustra en la FIG. 9 se describe ahora en detalle. En el ejemplo de la FIG. 9, un flujo de bits de entrada codificado 140 (también denominado señal de entrada) se pasa a la unidad de decodificación por entropía 142. La unidad de decodificación por entropía 142 puede decodificar por entropía el flujo de bits 140 para producir un bloque cuantificado de datos de vídeo residuales que tengan un primer espacio de color. Por ejemplo, la unidad de decodificación por entropía 142 puede decodificar por entropía elementos sintácticos particulares incluidos en el flujo de bits 140. La unidad de decuantificación/transformada inversa 144 puede decuantificar un bloque de coeficientes de transformada. Adicionalmente, la unidad de decuantificación/transformada inversa 144 puede aplicar una transformada inversa al bloque de coeficientes de transformada para determinar un bloque de transformada que comprenda datos de vídeo residuales. Por tanto, la unidad de decuantificación/transformada inversa 144 puede decuantificar y transformar de forma inversa bloques de datos de vídeo decodificados por entropía del flujo de bits 140. Cuando el decodificador de vídeo 31 se configura para transformar de color de forma inversa los bloques de datos residuales, el conmutador 148 alimenta un bloque de datos de vídeo residuales que

tiene un primer espacio de color al transformador de color adaptativo inverso 150. De esta manera, el transformador de color adaptativo inverso 150 puede recibir un bloque de transformada de una TU.

5 **[0151]** El transformador de color adaptativo inverso 150 puede transformar adaptativamente de forma inversa un
 bloque de datos de vídeo que tenga el primer espacio de color en un segundo bloque de datos de vídeo que tenga
 un segundo espacio de color RGB. Por ejemplo, el transformador de color adaptativo inverso 150 puede
 seleccionar una transformada inversa que se vaya a aplicar a un bloque de transformada de una TU. En este
 ejemplo, el transformador de color adaptativo inverso 150 puede aplicar la transformada inversa seleccionada al
 10 bloque de transformada para transformar el bloque de transformada desde el primer espacio de color en el espacio
 de color RGB. La unidad de compensación de predicción 152 puede combinar una imagen de referencia de la
 memoria de imágenes de referencia 154. Por ejemplo, la unidad de compensación de predicción 152 puede recibir
 un bloque de transformada de una TU de una CU. En este ejemplo, la unidad de compensación de predicción 152
 puede determinar un bloque de codificación para la CU. En este ejemplo, cada muestra del bloque de codificación
 15 de la CU puede ser igual a una suma de una muestra en el bloque de transformada y una muestra correspondiente
 en un bloque de predicción para una PU de la CU. El filtro de desbloqueo 157 puede desbloquear la imagen
 reconstruida combinada. La unidad de filtro SAO 158 puede realizar un filtrado SAO adicional si corresponde.

20 **[0152]** La salida de la unidad de filtro SAO 158 es la señal reconstruida 160. Si el decodificador de vídeo 31 está
 configurado para transformar de color de forma inversa bloques de transformada de datos de vídeo residuales, el
 conmutador 162 alimenta la señal reconstruida 160 a la memoria de imágenes de referencia 154 para su uso futuro
 como imagen de referencia. El decodificador de vídeo 31 también puede emitir la señal reconstruida 160 como
 imagen/vídeo 164.

25 **[0153]** En los ejemplos donde el decodificador de vídeo 31 está configurado para transformar de color de forma
 inversa los bloques de transformada de la señal de entrada original a diferencia de los bloques de datos de vídeo
 residuales, la unidad de decodificación por entropía 142 y la unidad de decuantificación/transformada inversa 144
 funcionan de la manera descrita anteriormente. El conmutador 148 está en la posición alternativa y alimenta la
 señal residual reconstruida directamente a la unidad de compensación de predicción 152. En este punto, el bloque
 residual proporcionado a la unidad de compensación de predicción 152 todavía está en el primer espacio de color,
 30 en lugar del espacio de color RGB.

[0154] La unidad de compensación de predicción 152 puede reconstruir un bloque de la imagen original y puede
 combinar el bloque residual con uno o más bloques de imágenes de la memoria de imágenes de referencia 154.
 El filtro de desbloqueo 157 y la unidad de filtro SAO 158 pueden funcionar como se describe anteriormente con
 35 respecto a la transformada inversa de bloques residuales de datos de vídeo. La salida de la unidad de filtro SAO
 158 es la señal reconstruida 160, cuyos bloques todavía están en el primer espacio de color, y pueden no tener el
 espacio de color RGB (por ejemplo, los bloques pueden tener aún el espacio de color RGB si se usó la
 transformada de identidad).

40 **[0155]** La señal reconstruida 160 se puede alimentar al transformador de color adaptativo inverso 166 por medio
 del conmutador 162, que está en la posición alternativa en comparación con la posición ilustrada en la FIG. 9. El
 transformador de color adaptativo inverso 166 puede transformar de color de forma inversa un bloque de señal
 reconstruida que tenga un primer espacio de color en un segundo bloque de datos de vídeo que tenga un segundo
 45 espacio de color RGB que use una transformada de color inversa de una o más transformadas de color inversas.
 En algunos ejemplos, la transformada inversa particular que usa el decodificador 31 se puede señalar en el flujo
 de bits 140. El transformador de color adaptativo inverso 166 puede alimentar el segundo bloque que tenga el
 segundo espacio de color para su salida como imagen/vídeo 164, así como a la memoria de imágenes de referencia
 154 para su almacenamiento y su uso futuro como imagen de referencia.

50 **[0156]** La FIG. 10 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con
 las técnicas de la presente divulgación. Las técnicas de la FIG. 10 se describirán con respecto a un decodificador
 de vídeo genérico. El decodificador de vídeo genérico puede, por ejemplo, corresponder al decodificador de vídeo
 30 de la FIG. 7 o al decodificador de vídeo 31 de la FIG. 9, aunque las técnicas de la FIG. 10 no se limitan a ningún
 tipo particular del decodificador de vídeo. Como los codificadores de vídeo típicamente realizan la decodificación
 55 de vídeo como parte del proceso de codificación, las técnicas de la FIG. 10 también se pueden realizar mediante
 un codificador de vídeo, tal como el codificador de vídeo 20 de la FIG. 6 y el codificador de vídeo 21 de la FIG. 8.
 El codificador de vídeo 20, por ejemplo, incluye la unidad de cuantificación inversa 58 y la unidad de procesamiento
 de transformada inversa 60, que forman parte de un bucle de decodificación, en el cual las técnicas de la FIG. 10
 se pueden implementar. Por tanto, mientras que las técnicas de la FIG. 10 se explicarán con referencia a un
 60 decodificador de vídeo, debe entenderse que este decodificador de vídeo puede ser parte de un codificador de
 vídeo.

[0157] En el ejemplo de la FIG. 10, el decodificador de vídeo determina para uno o más bloques de datos de
 vídeo que la transformada de color adaptativa está activada (210). En algunos ejemplos, el decodificador de vídeo
 65 puede determinar para el uno o más bloques de datos de vídeo que la transformada de color adaptativa está
 activada al recibir un elemento sintáctico que indica si la transformada de color adaptativa está activada. El

elemento sintáctico se puede recibir, por ejemplo, en el PPS o en otro nivel. Al analizar el elemento sintáctico recibido, el decodificador de vídeo puede determinar si la transformada de color adaptativa está activada o desactivada. En otros ejemplos, el decodificador de vídeo puede determinar para el uno o más bloques de los datos de vídeo que la transformada de color adaptativa se activa determinando un formato de croma para los datos de vídeo. Por ejemplo, en respuesta a que determinar un formato de croma para los datos de vídeo es 4:4:4, el decodificador de vídeo puede determinar que la transformada de color adaptativa está activada. En respuesta a determinar que un formato de croma para los datos de vídeo no es 4:4:4, el decodificador de vídeo puede determinar que la transformada de color adaptativa está desactivada.

[0158] Para datos de vídeo con transformada de color adaptativa activada, el decodificador de vídeo puede determinar un parámetro de cuantificación para el uno o más bloques (212). En respuesta a que un valor del parámetro de cuantificación está por debajo de un umbral, el decodificador de vídeo puede modificar el parámetro de cuantificación para determinar un parámetro de cuantificación modificado (214). El umbral puede, por ejemplo, ser cero, y un valor del parámetro de cuantificación modificado puede ser mayor o igual a cero. El parámetro de cuantificación modificado puede ser menor o igual a 51 más un valor de desviación. Para modificar el parámetro de cuantificación, el decodificador de vídeo puede añadir un valor de desviación al parámetro de cuantificación. El decodificador de vídeo puede recibir un indicador para indicar si el valor de desviación debe añadirse al parámetro de cuantificación.

[0159] El decodificador de vídeo puede, por ejemplo, recibir el valor de desviación como un elemento sintáctico. El valor de desviación puede ser una desviación al parámetro de cuantificación (cuando la transformada de color adaptativa esté activada para el bloque). El decodificador de vídeo puede decuantificar los coeficientes de transformada en base al parámetro de cuantificación modificado (216).

[0160] La FIG. 11 muestra un ejemplo de un procedimiento de codificación de datos de vídeo de acuerdo con las técnicas de la presente divulgación. Las técnicas de la FIG. 11 se describirán con respecto a un codificador de vídeo genérico. El codificador de vídeo genérico puede, por ejemplo, corresponder al codificador de vídeo 20 de la FIG. 6 o al codificador de vídeo 21 de la FIG. 8, aunque las técnicas de la FIG. 11 no se limitan a ningún tipo particular de codificador de vídeo. El codificador de vídeo selecciona un formato de muestreo de croma para los datos de vídeo (220). En respuesta a que el formato de muestreo de croma es un primer formato de muestreo de croma, el codificador de vídeo genera un elemento sintáctico para indicar si la transformada de color adaptativa está activada (222). En respuesta a que el formato de muestreo de croma no es el primer formato de muestreo de croma, el codificador de vídeo codifica los datos de vídeo sin la transformada de color adaptativa (224). El primer formato de muestreo de croma puede, por ejemplo, ser un formato de muestreo de croma 4:4:4.

[0161] La FIG. 12 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con las técnicas de la presente divulgación. Las técnicas de la FIG. 12 se describirán con respecto a un decodificador de vídeo genérico. El decodificador de vídeo genérico puede, por ejemplo, corresponder al decodificador de vídeo 30 de la FIG. 7 o al decodificador de vídeo 31 de la FIG. 9, aunque las técnicas de la FIG. 12 no se limitan a ningún tipo particular de decodificador de vídeo. El decodificador de vídeo genérico también puede corresponder a un bucle de decodificación de un codificador de vídeo, en algunos ejemplos.

[0162] En el ejemplo de la FIG. 12, el decodificador de vídeo determina, en base a un formato de muestreo de croma para los datos de vídeo, que la transformada de color adaptativa está activada para uno o más bloques de los datos de vídeo (230). El decodificador de vídeo puede, por ejemplo, determinar que la transformada de color adaptativa está activada para uno o más bloques de los datos de vídeo al determinar que el formato de muestreo de croma es un formato de muestreo 4:4:4. El decodificador de vídeo puede determinar un parámetro de cuantificación para el uno o más bloques en base a determinar que la transformada de color adaptativa está activada (232) y decuantificar los coeficientes de transformada en base al parámetro de cuantificación (234) determinado.

[0163] El decodificador de vídeo también puede, por ejemplo, determinar para uno o más segundos bloques de los datos de vídeo que un formato de muestreo de croma para los bloques de vídeo sea un formato de muestreo de croma que no sea 4:4:4 y que el formato de muestreo de croma sea otro que 4:4:4, determinando que la transformada de color adaptativa esté desactivada para el segundo uno o más bloques. El decodificador de vídeo puede, por ejemplo, determinar que la transformada de color adaptativa está desactivada para el segundo uno o más bloques sin recibir un elemento sintáctico que no sea la indicación del formato de muestreo de croma, para indicar si la transformada de color adaptativa está desactivada.

[0164] La FIG. 13 muestra un ejemplo de un procedimiento de codificación de datos de vídeo de acuerdo con las técnicas de la presente divulgación. Las técnicas de la FIG. 13 se describirán con respecto a un codificador de vídeo genérico. El codificador de vídeo genérico puede, por ejemplo, corresponder al codificador de vídeo 20 de la FIG. 6 o al codificador de vídeo 21 de la FIG. 8, aunque las técnicas de la FIG. 13 no se limitan a ningún tipo particular de codificador de vídeo. El codificador de vídeo determina para uno o más bloques de datos de vídeo que la transformada de color adaptativa se usa para codificar los bloques (240). El codificador de vídeo determina un parámetro de cuantificación para un primer componente de color de un primer espacio de color de los datos de

vídeo (242). El codificador de vídeo cuantifica los coeficientes de transformada en base al parámetro de cuantificación (244). El codificador de vídeo genera, para su inclusión en un conjunto de parámetros de imágenes, uno o más valores de desviación que representan una diferencia entre un parámetro de cuantificación para un primer componente de color de un segundo espacio de color de los datos de vídeo y el parámetro de cuantificación para el primer componente de color del segundo espacio de color de los datos de vídeo (246).

[0165] La FIG. 14 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con las técnicas de la presente divulgación. Las técnicas de la FIG. 14 se describirán con respecto a un decodificador de vídeo genérico. El decodificador de vídeo genérico puede, por ejemplo, corresponder al decodificador de vídeo 30 de la FIG. 7 o al decodificador de vídeo 31 de la FIG. 9, aunque las técnicas de la FIG. 14 no se limitan a ningún tipo particular de decodificador de vídeo. En el ejemplo de la FIG. 14, el decodificador de vídeo determina para uno o más bloques de los datos de vídeo que la transformada de color adaptativa está activada (250). En respuesta a que la transformada de color adaptativa está activada, el decodificador de vídeo recibe, en un conjunto de parámetros de imágenes, uno o más valores de desviación (252). El decodificador de vídeo determina un parámetro de cuantificación para un primer componente de color de un primer espacio de color en base al primero del uno o más valores de desviación (254) y decuantifica los coeficientes de transformada en base al parámetro de cuantificación modificado (256). El uno o más valores de desviación pueden incluir un valor de desviación para el primer componente de color, un valor de desviación para un segundo componente de color y un valor de desviación para un tercer componente de color.

[0166] El decodificador de vídeo puede determinar un parámetro de cuantificación para un primer componente de color de un segundo espacio de color. Para determinar el parámetro de cuantificación para el primer componente de color del primer espacio de color en base al primero del uno o más valores de desviación, el decodificador de vídeo puede convertir el parámetro de cuantificación para el primer componente de color del segundo espacio de color en el parámetro de cuantificación para el primer componente de color del primer espacio de color añadiendo el primero del uno o más valores de desviación al parámetro de cuantificación para el primer componente de color del segundo espacio de color.

[0167] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o en cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en o transmitir a través de un medio legible por ordenador como una o más instrucciones o código, y ejecutarse por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación incluyendo cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o a (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0168] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Asimismo, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una página web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se refieren a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde unos discos reproducen normalmente los datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior también se deben incluir dentro del alcance de los medios legibles por ordenador.

[0169] Las instrucciones se pueden ejecutar mediante uno o más procesadores, tales como uno o más DSP, microprocesadores de uso general, ASIC, FPGA u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o de software dedicados configurados para codificar y

decodificar, o incorporar en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

5 **[0170]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar
10 mediante un grupo de unidades de hardware interoperativas, que incluya uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

[0171] Se han descrito diversos ejemplos. La invención se define en las reivindicaciones adjuntas.

REIVINDICACIONES

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

5 determinar un formato de muestreo de croma para los datos de vídeo;
 en respuesta a que el formato de muestreo de croma es 4:4:4, recibir en un conjunto de parámetros de imágenes, PPS, un elemento sintáctico que indique si la transformada de color adaptativa está activada para las imágenes de los datos de vídeo que se refieren al PPS;
 en respuesta a que la transformada de color adaptativa está activada, recibir, en el PPS, uno o más
 10 valores de desviación de nivel PPS;
 en respuesta a que la transformada adaptativa de color está activada, recibir, en una cabecera de fragmento de un fragmento que se refiere al PPS, uno o más valores de desviación de nivel de fragmento;
 determinar un parámetro de cuantificación para uno o más bloques del fragmento en base al uno o más valores de desviación de nivel de PPS y al uno o más valores de desviación de nivel de fragmento; y
 15 decuantificar los coeficientes de transformada en base al parámetro de cuantificación determinado.

2. El procedimiento de la reivindicación 1, que comprende además:

20 determinar para uno o más segundos bloques de los datos de vídeo que un formato de muestreo de croma para los bloques de vídeo es un formato de muestreo de croma distinto de 4:4:4;
 en base a que el formato de muestreo de croma es distinto de 4:4:4, determinar que la transformada de color adaptativa está desactivada para el segundo uno o más bloques.

3. El procedimiento de la reivindicación 2, en el que determinar que la transformada de color adaptativa está desactivada para el segundo uno o más bloques comprende determinar que la transformada de color adaptativa está desactivada para el segundo uno o más bloques sin recibir un elemento sintáctico, que no sea la indicación del formato de muestreo de croma, que indica si la transformada de color adaptativa está desactivada.

4. Un dispositivo para decodificar datos de vídeo, comprendiendo el dispositivo:

30 una memoria de datos de vídeo;
 uno o más procesadores configurados para:
 35 determinar un formato de muestreo de croma para los datos de vídeo;
 en respuesta a que el formato de muestreo de croma es 4:4:4, recibir en un conjunto de parámetros de imágenes, PPS, un elemento sintáctico que indique si la transformada de color adaptativa está activada para las imágenes de los datos de vídeo que se refieren al PPS;
 en respuesta a que la transformada de color adaptativa está activada, recibir, en el PPS, uno o
 40 más valores de desviación de nivel PPS;
 en respuesta a que la transformada de color adaptativa está activada, recibir, en una cabecera de fragmento de un fragmento que se refiere al PPS, uno o más valores de desviación de nivel de fragmento;
 determinar un parámetro de cuantificación para uno o más bloques del fragmento en base al uno o más valores de desviación de nivel de PPS y al uno o más valores de desviación de nivel de
 45 fragmento; y
 decuantificar los coeficientes de transformada en base al parámetro de cuantificación determinado.

5. El dispositivo de la reivindicación 4, en el que el uno o más procesadores están configurados además para:

50 determinar para uno o más segundos bloques de datos de vídeo que un formato de muestreo de croma para los bloques de vídeo es un formato de muestreo de croma distinto de 4:4:4;
 determinar que la transformada de color adaptativa está desactivada para el segundo uno o más bloques en base al formato de muestreo de croma que no sea 4:4:4.

6. El dispositivo de la reivindicación 5, en el que, para determinar que la transformada de color adaptativa está desactivada para el segundo uno o más bloques, el uno o más procesadores están configurados además para determinar que la transformada de color adaptativa está desactivada para el segundo uno o más bloques sin recibir un elemento sintáctico, aparte de la indicación del formato de muestreo de croma, que indica si la transformada de color adaptativa está desactivada.

7. El dispositivo, de acuerdo con la reivindicación 4, en el que el dispositivo comprende al menos uno de:

65 un circuito integrado;
 un microprocesador; o
 un dispositivo de comunicación inalámbrica.

8. El dispositivo de la reivindicación 4, en el que el dispositivo comprende un dispositivo de comunicación inalámbrica, que comprende además: un receptor configurado para recibir datos de vídeo codificados.
- 5 9. El dispositivo de acuerdo con la reivindicación 8, en el que el dispositivo de comunicación inalámbrica comprende un teléfono, en el que el receptor está configurado para recibir los datos de vídeo codificados en una señal modulada de acuerdo con un estándar de telecomunicaciones inalámbricas.
- 10 10. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan por uno o más procesadores, causan que el uno o más procesadores:
- determine un formato de muestreo de croma para los datos de vídeo;
 en respuesta a que el formato de muestreo de croma es 4:4:4, recibir en un conjunto de parámetros de imágenes, PPS, un elemento sintáctico que indique si la transformada de color adaptativa está activada para las imágenes de los datos de vídeo que se refieren al PPS;
 15 en respuesta a que la transformada de color adaptativa está activada, recibir, en el PPS, uno o más valores de desviación de nivel PPS;
 en respuesta a que la transformada de color adaptativa está activada, recibir, en una cabecera de fragmento de un fragmento que se refiere al PPS, uno o más valores de desviación de nivel de fragmento;
 20 determine un parámetro de cuantificación para uno o más bloques del fragmento en base al uno o más valores de desviación de nivel de PPS y al uno o más valores de desviación de nivel de fragmento; y
 decuantifique los coeficientes de transformada en base al parámetro de cuantificación determinado.
- 25 11. El procedimiento de la reivindicación 1, en el que el uno o más valores de desviación comprenden un valor de desviación para un primer componente de color, un valor de desviación para un segundo componente de color y un valor de desviación para un tercer componente de color.
- 30 12. El procedimiento de la reivindicación 1, en el que el parámetro de cuantificación comprende un parámetro de cuantificación para un primer componente de color de un primer espacio de color, y en el que determinar el parámetro de cuantificación para el primer componente de color del primer espacio de color comprende:
 convertir un parámetro de cuantificación para un primer componente de color de un segundo espacio de color en el parámetro de cuantificación para el primer componente de color del primer espacio de color añadiendo uno del uno o más valores de desviación de nivel PPS y uno del uno o más valores de desviación de nivel de fragmento al parámetro de cuantificación para el primer componente de color del segundo espacio de color.
- 35 13. El procedimiento de la reivindicación 12, en el que el primer espacio de color comprende un espacio de color RGB, y en el que el segundo espacio de color comprende un espacio de color YCbCr.
- 40 14. El dispositivo de la reivindicación 4, en el que el uno o más valores de desviación comprenden un valor de desviación para un primer componente de color, un valor de desviación para un segundo componente de color y un valor de desviación para un tercer componente de color y/o en el que el parámetro de cuantificación comprende un parámetro de cuantificación para un primer componente de color de un primer espacio de color, y en el que el uno o más procesadores están configurados además para:
- 45 determinar el parámetro de cuantificación para el primer componente de color del primer espacio de color convirtiendo un parámetro de cuantificación para un primer componente de color de un segundo espacio de color en el parámetro de cuantificación para el primer componente de color del primer espacio de color añadiendo uno del uno o más valores de desviación de nivel de PPS y uno del uno o más valores de desviación de nivel de fragmento al parámetro de cuantificación para el primer componente de color del segundo espacio de color y preferentemente en el que el primer espacio de color comprende un espacio de color RGB, y en el
 50 que el segundo espacio de color comprende un espacio de color YCbCr.

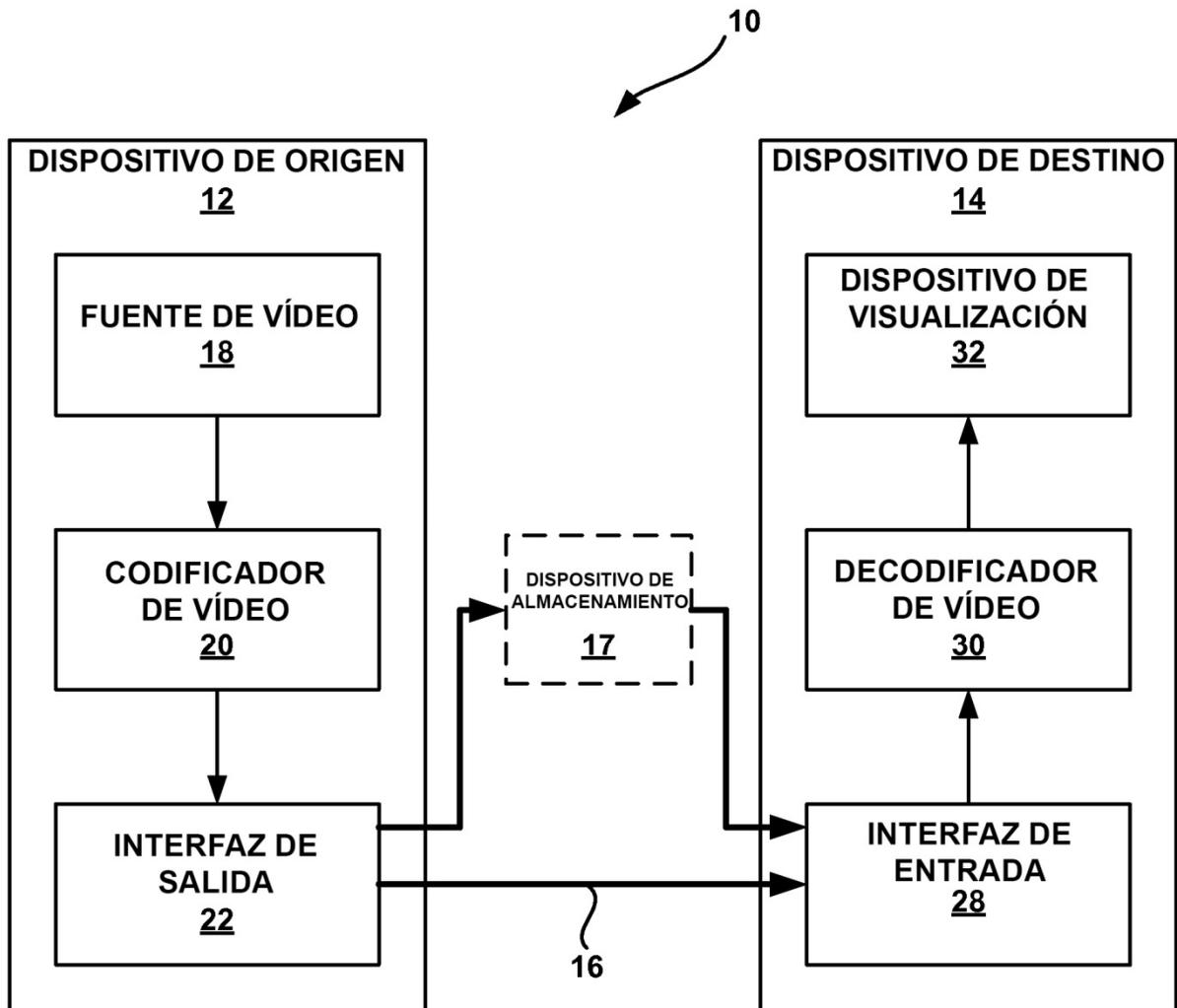
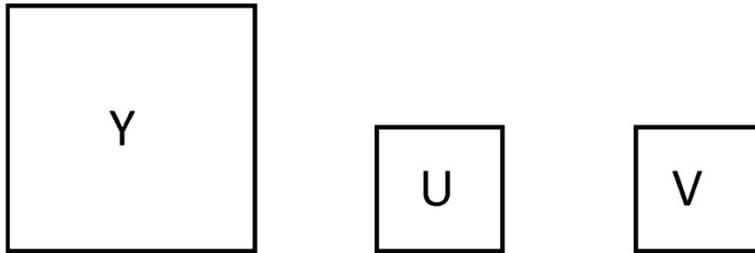
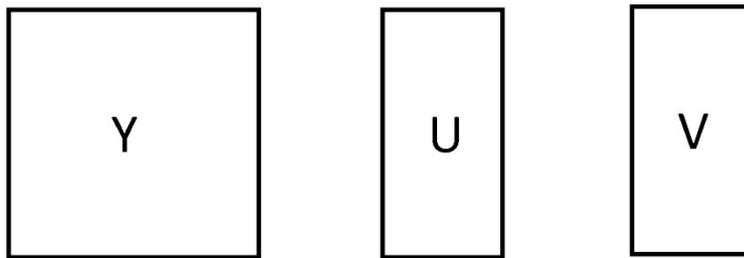


FIG. 1



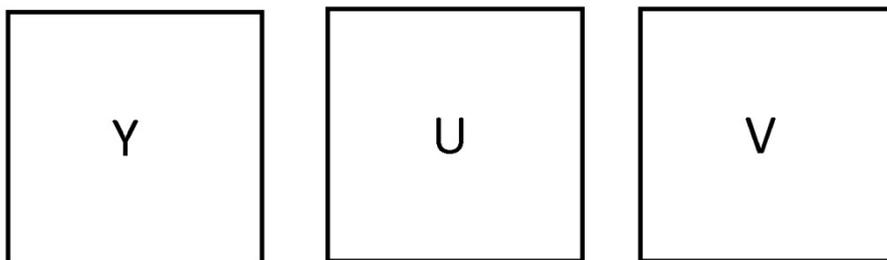
Formato de muestra 4:2:0

FIG. 2A



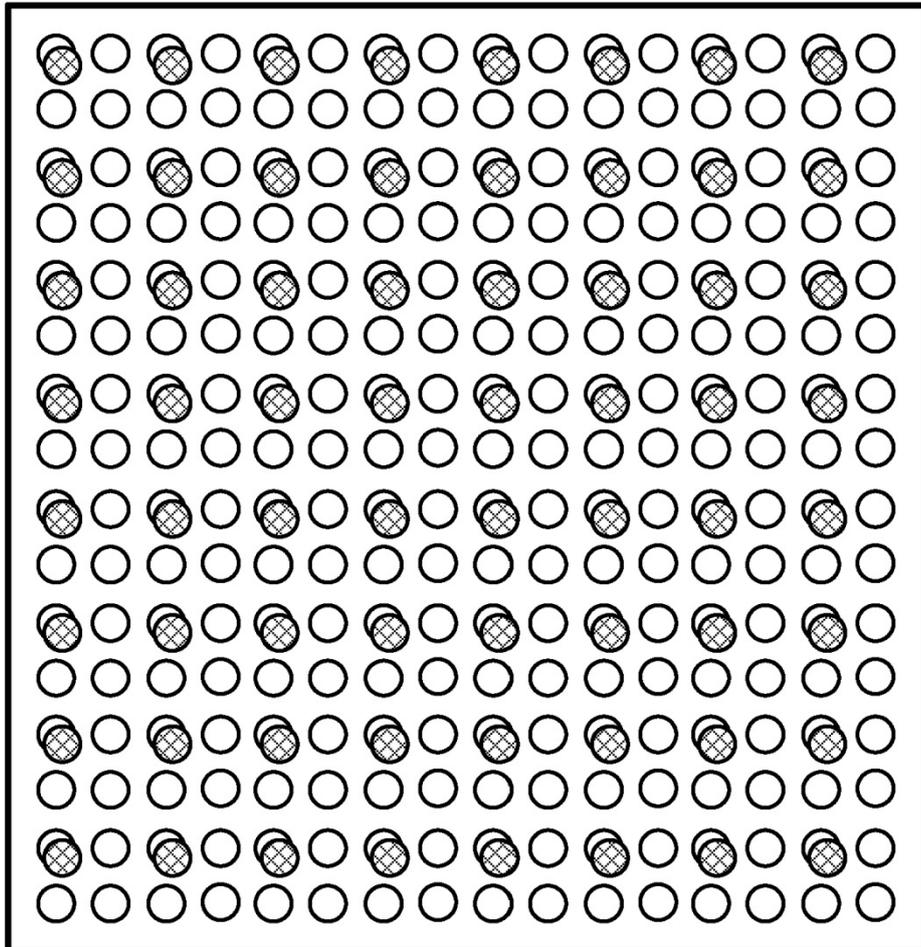
Formato de muestra 4:2:2

FIG. 2B



Formato de muestra 4:4:4

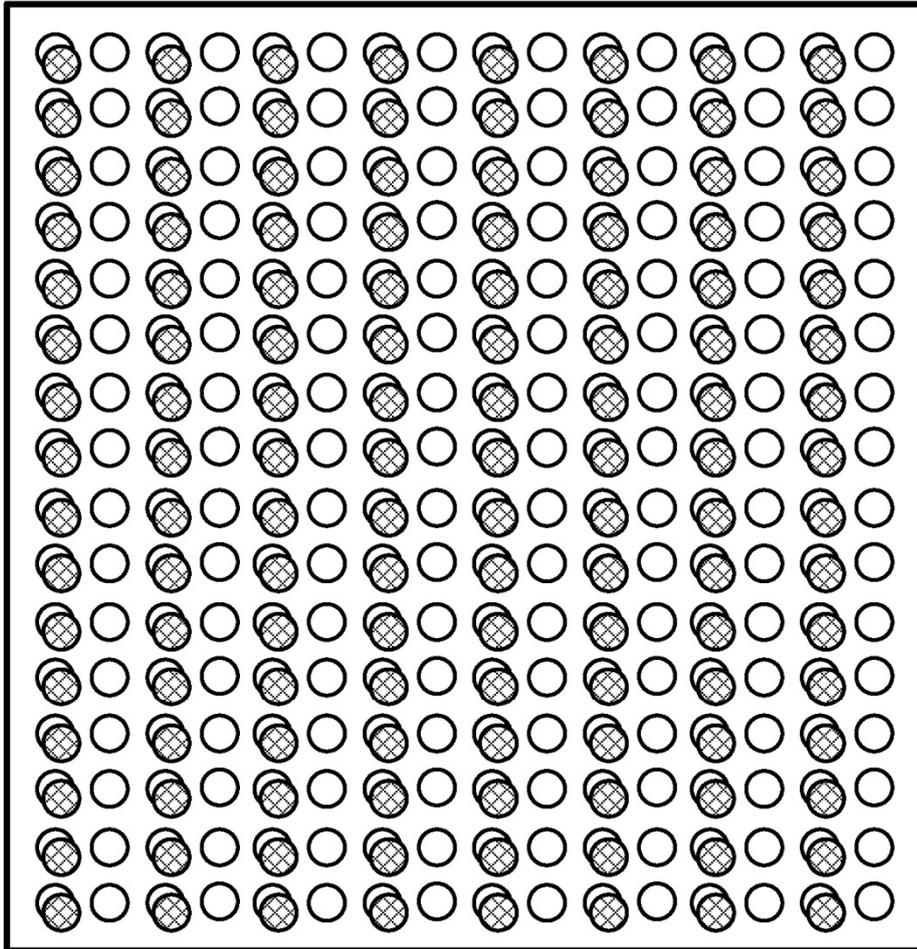
FIG. 2C



CU 16x16 con formato de muestra 4:2:0

- Muestra de luma
- ⊗ Muestras de croma

FIG. 3



CU 16x16 con formato de muestra 4:2:2

- Muestra de luma
- ⊗ Muestras de croma

FIG. 4

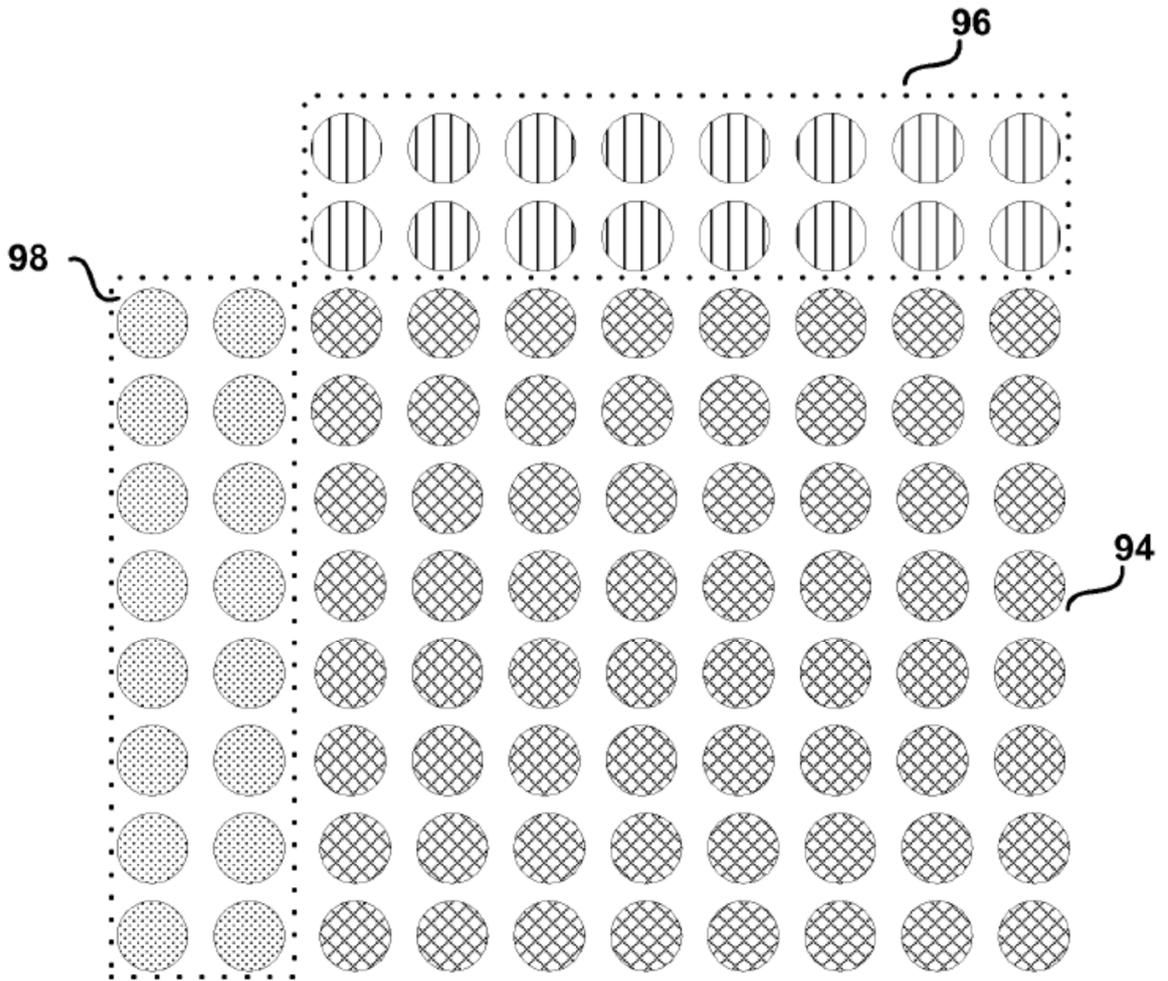


FIG. 5

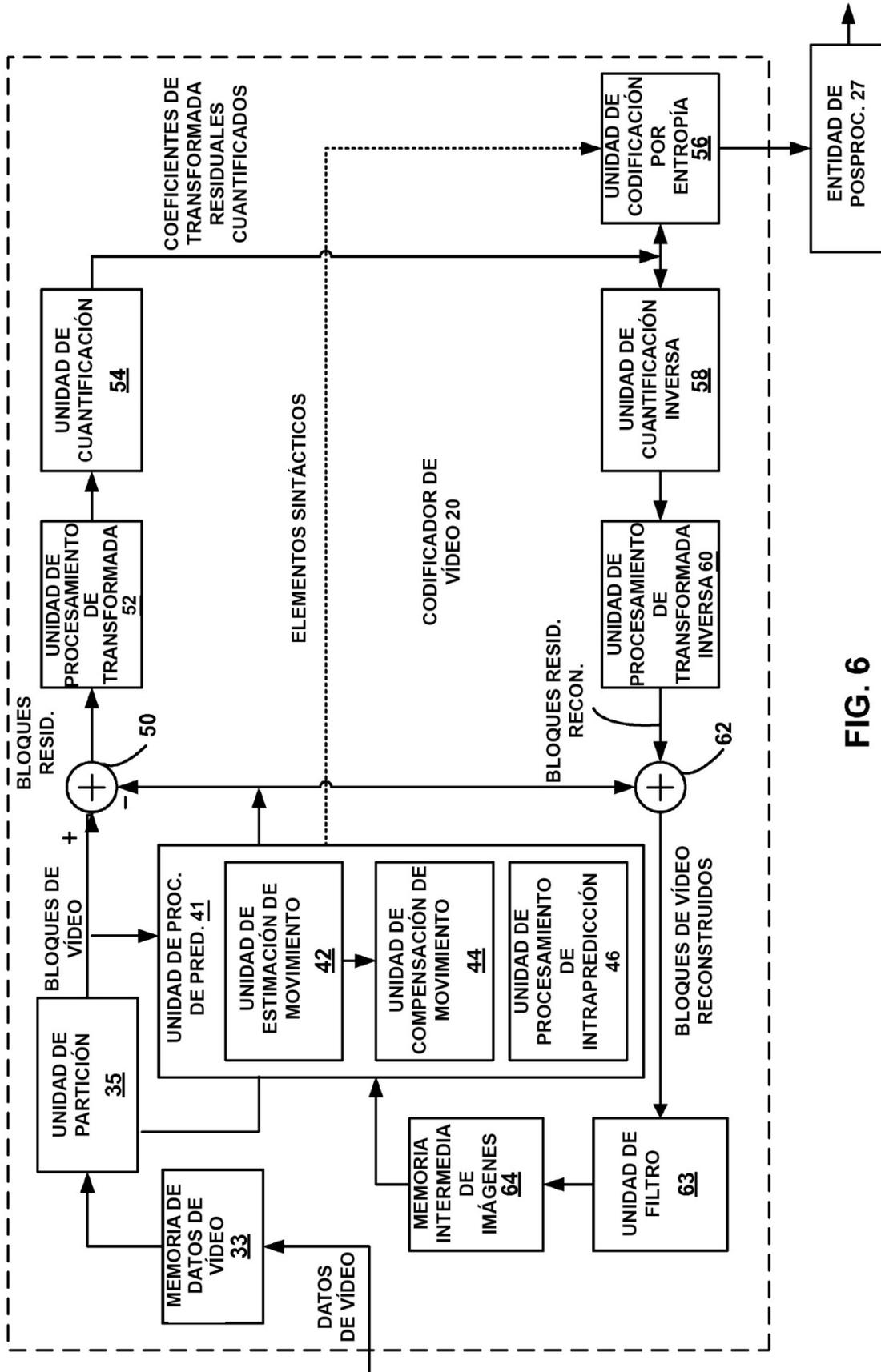


FIG. 6

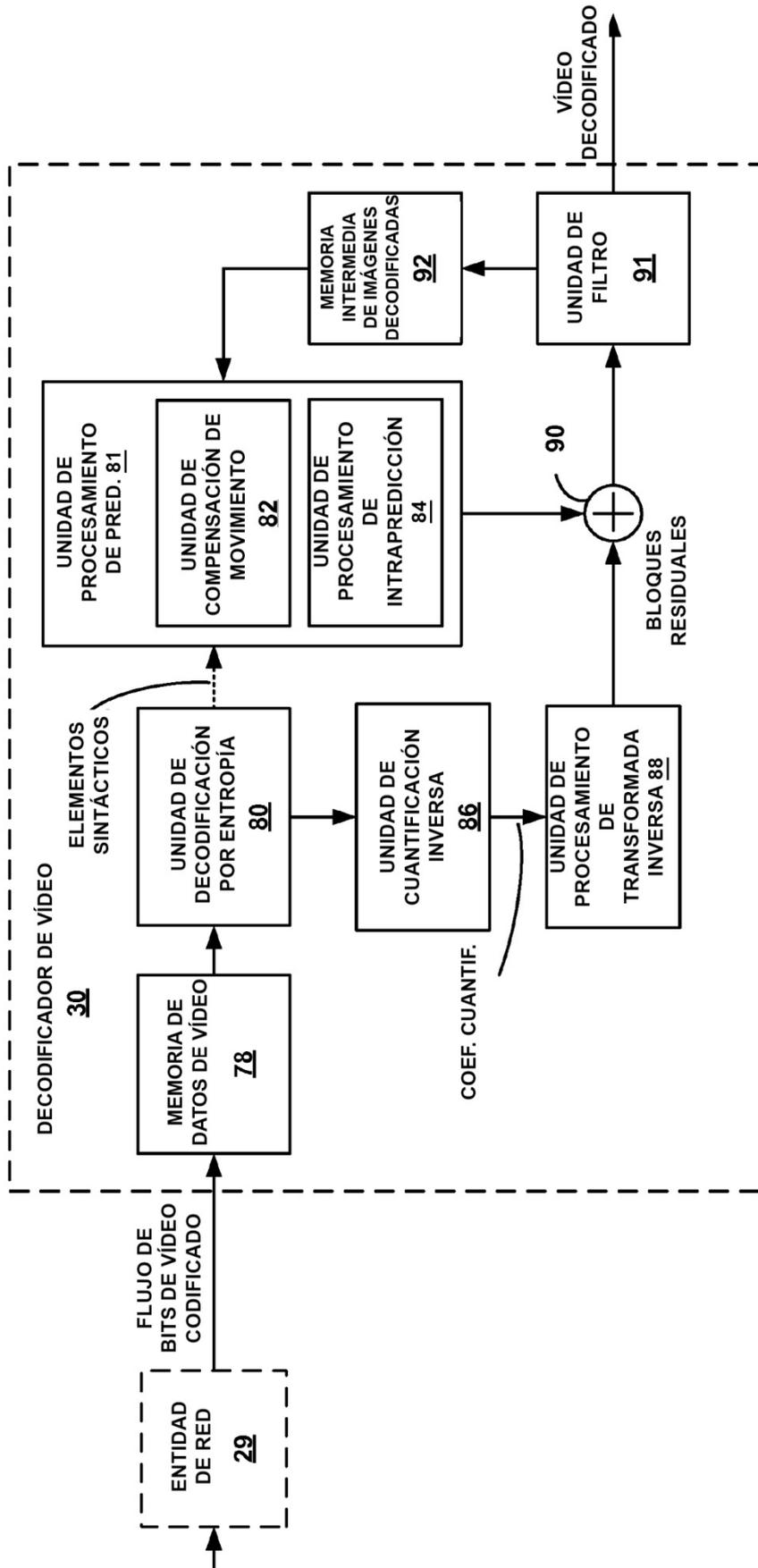


FIG. 7

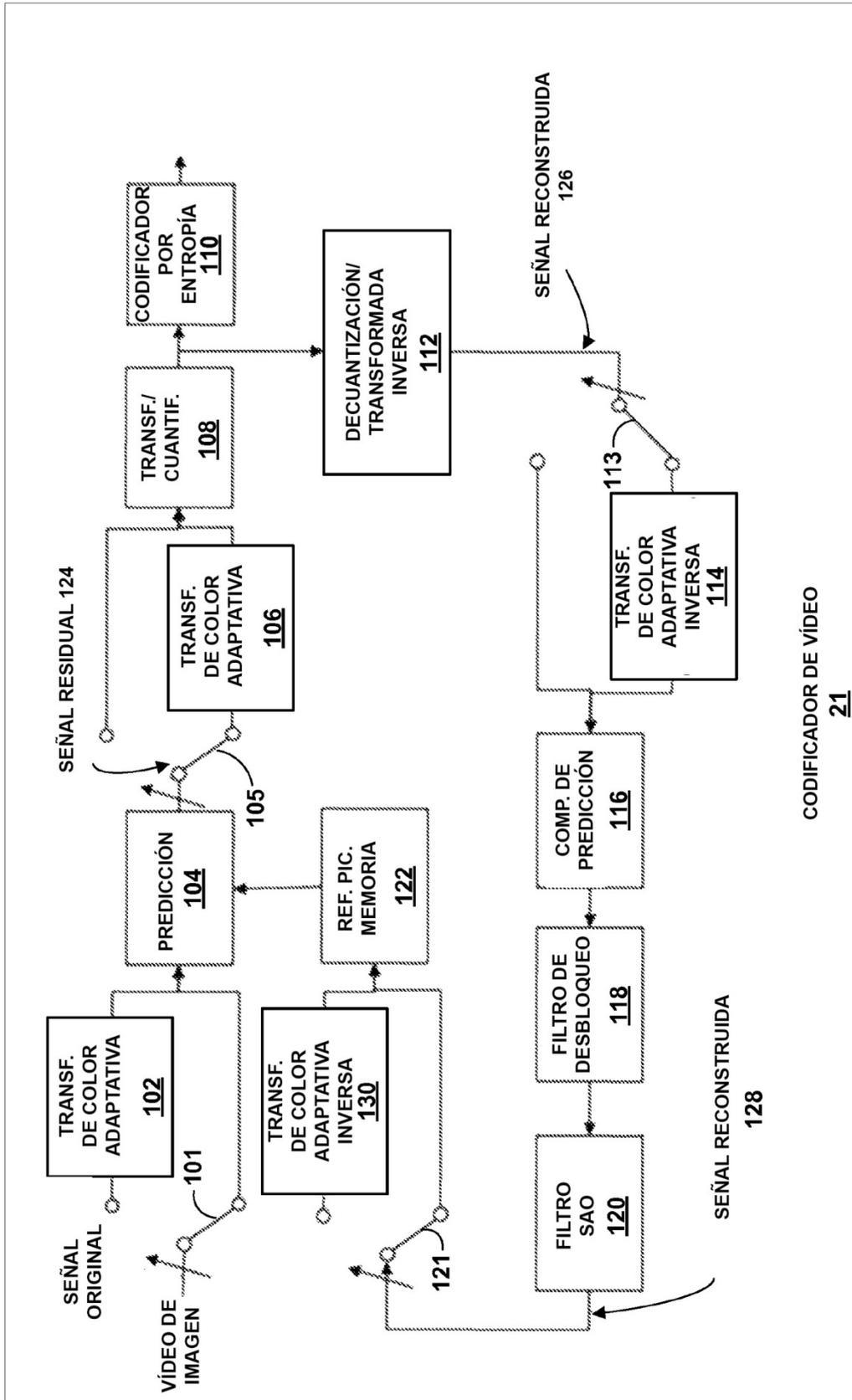
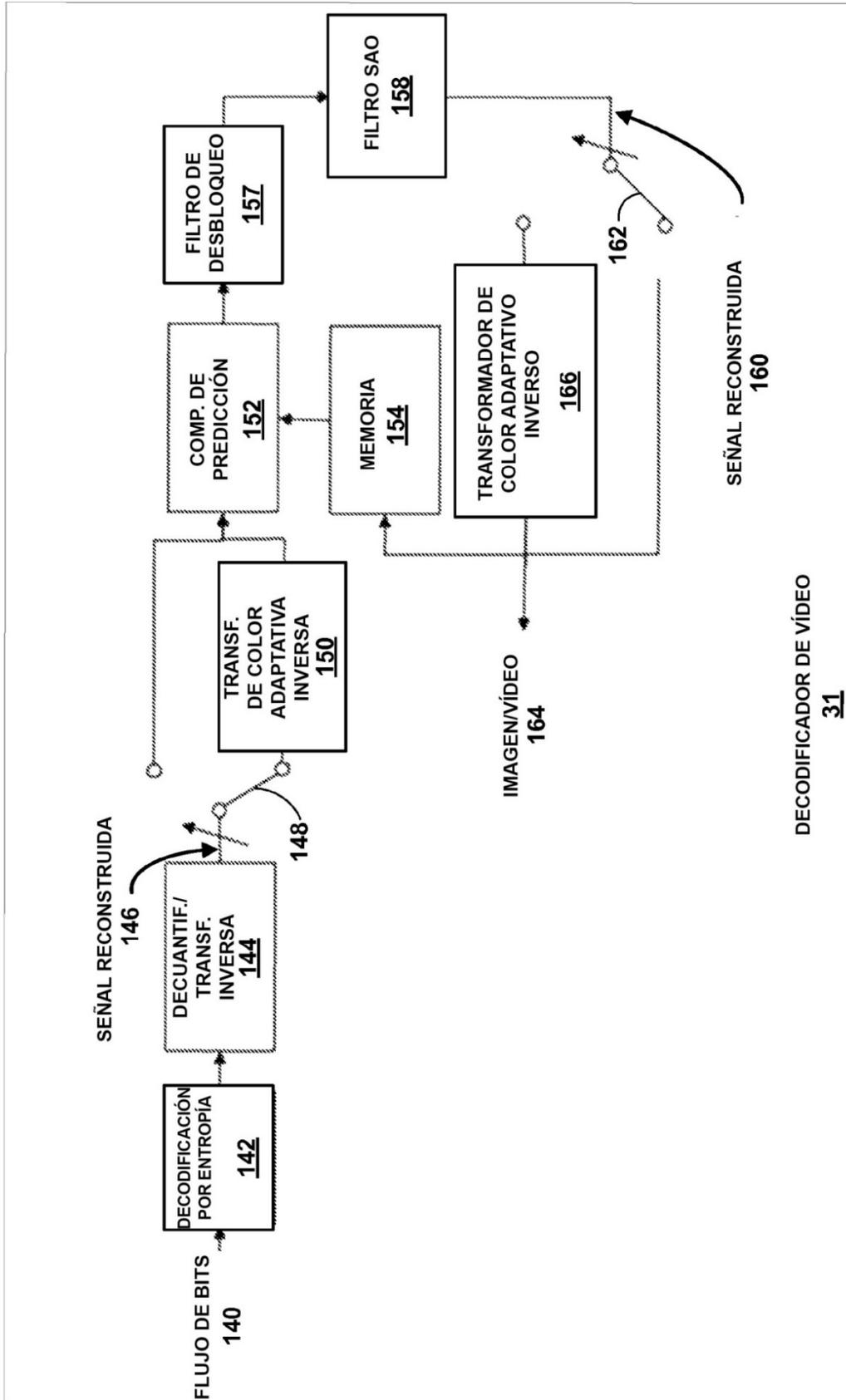


FIG. 8



DECODIFICADOR DE VÍDEO 31

FIG. 9

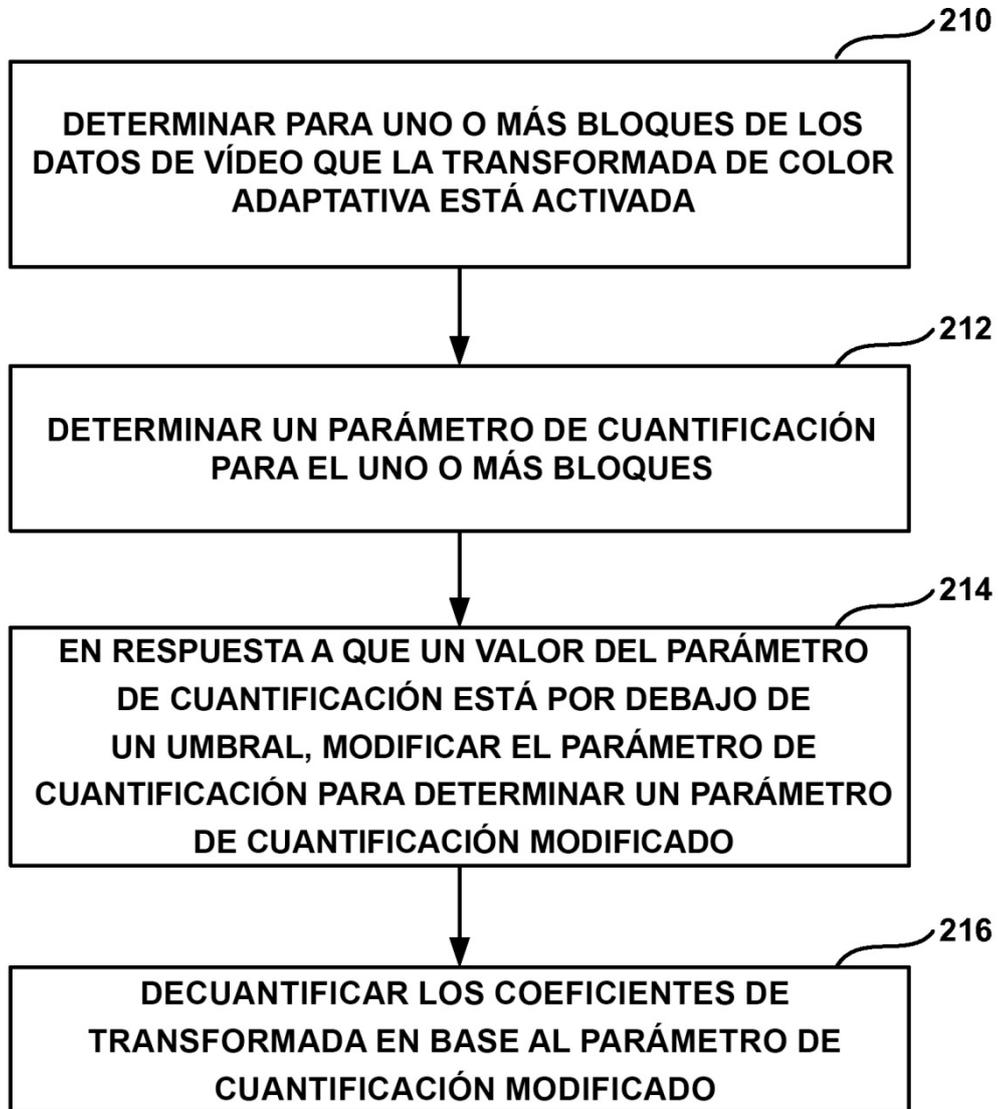


FIG. 10

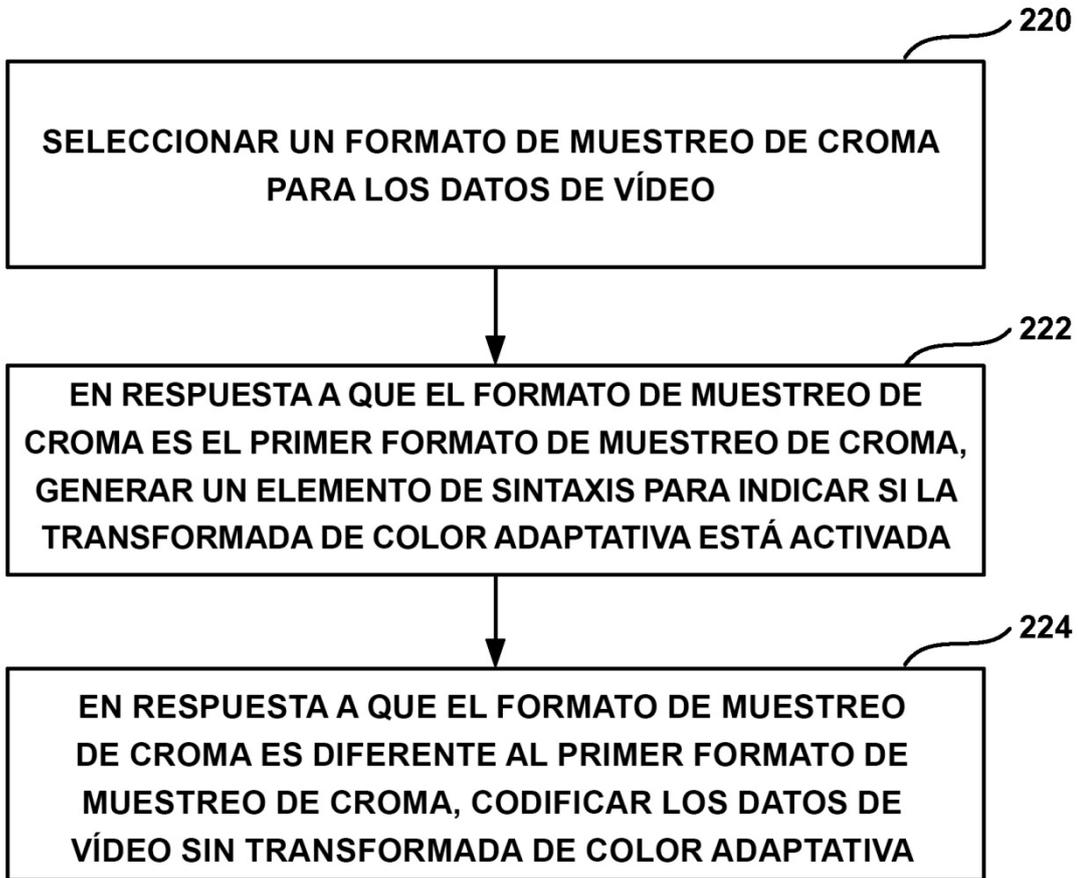


FIG. 11

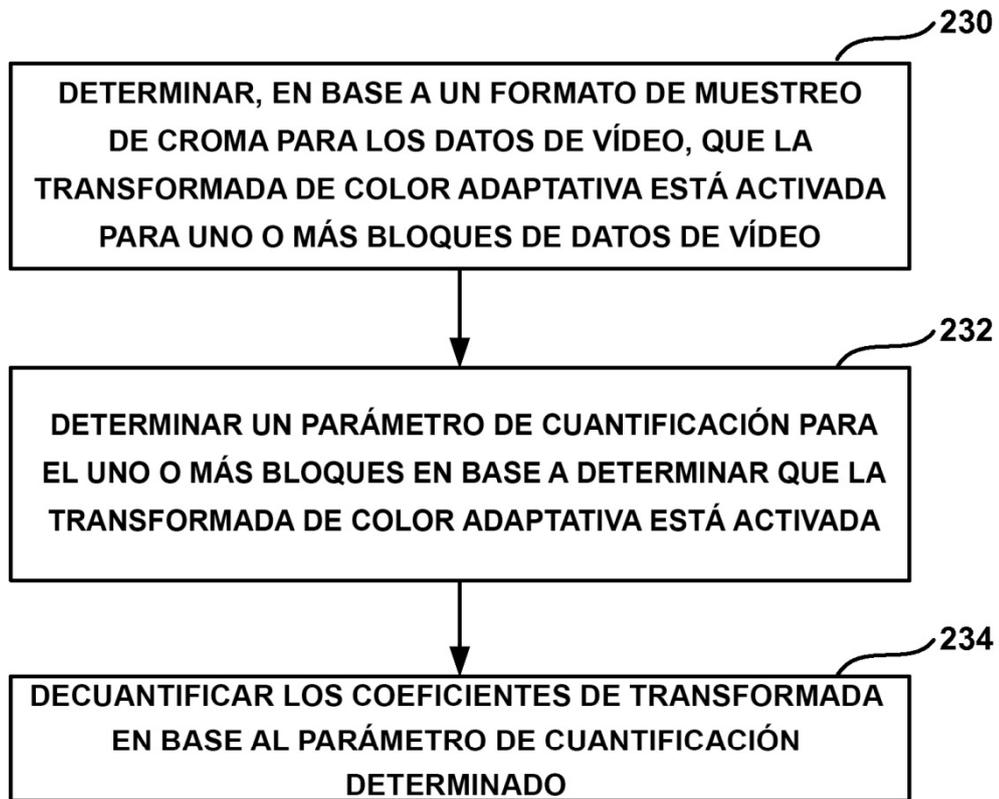


FIG. 12

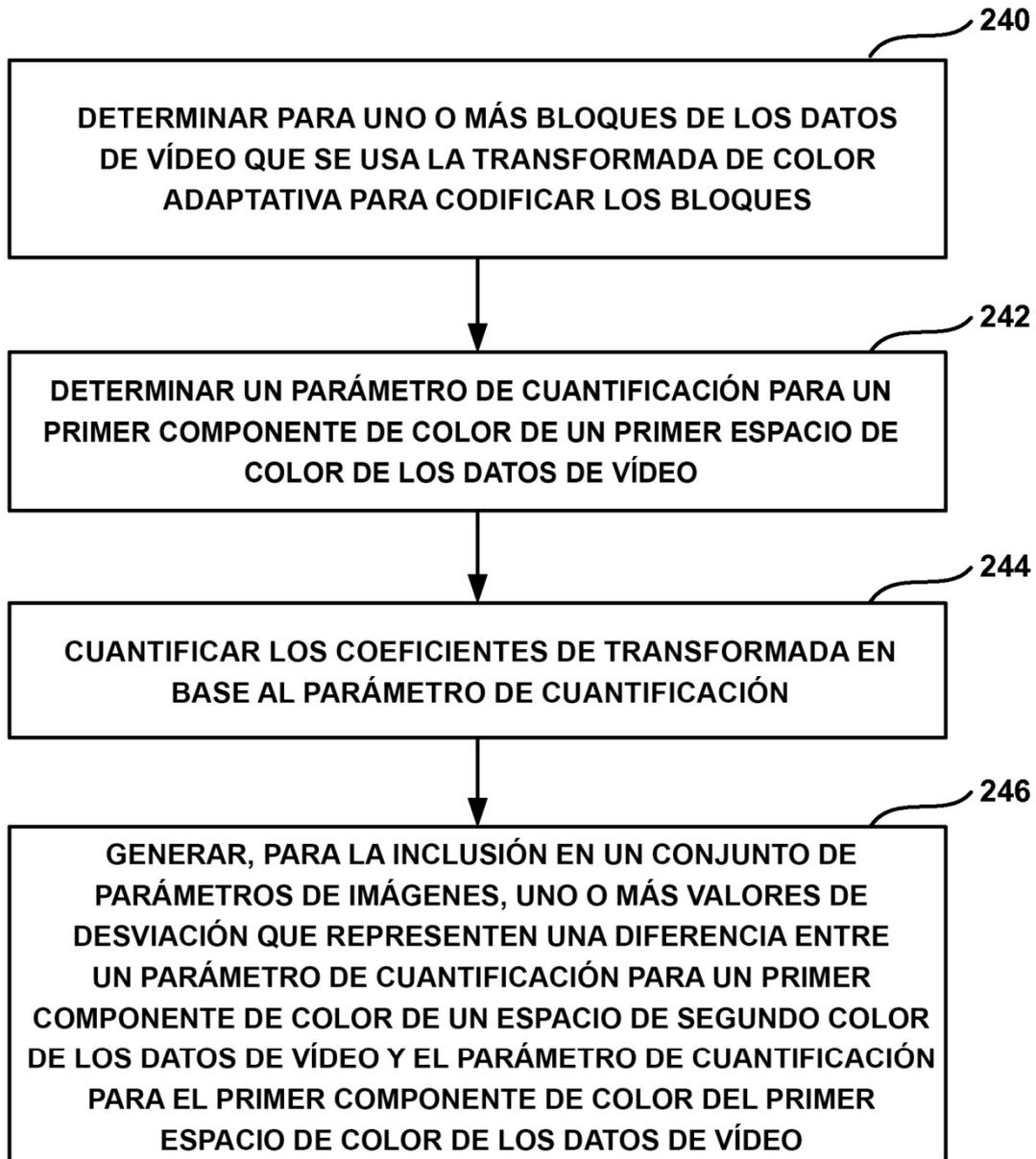


FIG. 13

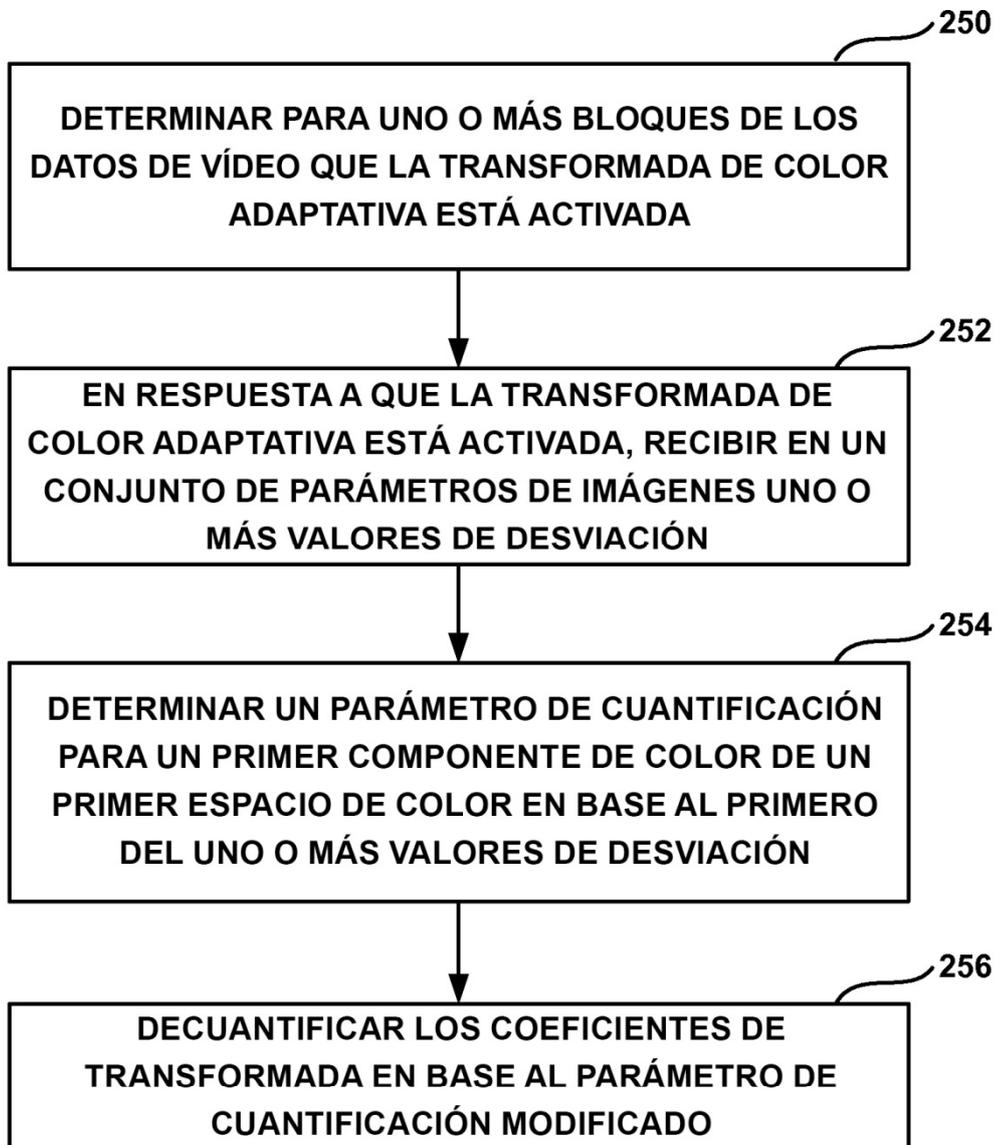


FIG. 14