

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 865 389**

51 Int. Cl.:

H04N 19/13 (2014.01)
H04N 19/156 (2014.01)
H04N 19/176 (2014.01)
H04N 19/44 (2014.01)
H04N 19/60 (2014.01)
H04N 19/91 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **11.04.2014** **PCT/US2014/033780**
87 Fecha y número de publicación internacional: **16.10.2014** **WO14169199**
96 Fecha de presentación y número de la solicitud europea: **11.04.2014** **E 14722509 (8)**
97 Fecha y número de publicación de la concesión europea: **07.04.2021** **EP 2984830**

54 Título: **Actualización del parámetro de rice para la codificación de nivel de coeficiente en el procedimiento de codificación de vídeo**

30 Prioridad:

12.04.2013 US 201361811632 P
07.06.2013 US 201361832652 P
12.07.2013 US 201361845850 P
14.03.2014 US 201461953647 P
10.04.2014 US 201414250282

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
15.10.2021

73 Titular/es:

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

72 Inventor/es:

SOLE ROJALS, JOEL;
KARCZEWICZ, MARTA y
JOSHI, RAJAN LAXMAN

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 865 389 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

Actualización del parámetro de rice para la codificación de nivel de coeficiente en el procedimiento de codificación de vídeo

Campo técnico

Esta divulgación se relaciona con la codificación de vídeo, y más particularmente a las técnicas de coeficientes de codificación de transformación.

Antecedentes

Las capacidades de vídeo digital pueden incorporarse en un amplio rango de dispositivos, incluidos televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de vídeo juegos, consolas de vídeo juegos, teléfonos celulares o de radio por satélite, dispositivos de vídeo conferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-TH.263, ITU-TH.264/MPEG-4, Parte 10, Codificación de vídeo Avanzada (AVC), el estándar de codificación de vídeo de alta eficiencia (HEVC) actualmente bajo desarrollo, y extensiones de dichos estándares, para transmitir, recibir y almacenar la información de vídeo digital más eficientemente.

Las técnicas de compresión de vídeo incluyen predicción espacial y/o predicción temporal para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo, es decir, un fotograma de vídeo o una porción de un fotograma de vídeo) puede dividirse en bloques. Los bloques de vídeo en un segmento intracodificado (I) de una imagen se codifican mediante el uso de la predicción espacial con respecto a las muestras de referencia en los bloques vecinos de la misma imagen. Los bloques de vídeo en un segmento intercodificado (P o B) pueden usar la predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden referirse como fotogramas y las imágenes de referencia pueden referirse como fotogramas de referencia.

La predicción espacial o temporal resulta en un bloque predictivo para un bloque a codificar. Los datos residuales representan las diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forma 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 mejor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, lo que resulta en coeficientes de transformación residuales, que luego pueden cuantificarse. Los coeficientes de transformación cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden escanearse en un orden particular para producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación de entropía para lograr una compresión aún mayor.

La contribución de JCTVC de Nguyen y otros titulado "Reduced-complexity entropy coding of transform coefficient levels using a combination of VLC and PIPE" (equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-TSG16 WP3 y ISO/IEC JTC1/SC29/WG11 4ta reunión: Daegu, Corea del Sur del 20-28 de enero de 2011, documento número JCTVC-D336) presenta un procedimiento de complejidad reducida para la codificación de niveles de coeficiente de transformación absoluta para el caso de alta eficiencia. La principal idea subyacente es permitir la mezcla de VLC estructurados y bits codificados PIPE/CABAC.

En el artículo "Adaptive coding of DCT coefficients by Golomb-Rice codes" por Menon (Procesamiento de Imagen, 1998. ICIP 98. Procedimientos. Conferencia Internacional de 1998, volumen 1, páginas 516-520. IEEE, 1998)), se describe la aplicación en el algoritmo de línea base JPEG con pérdida de algunas de las técnicas de codificación de Golomb-Rice que surgieron en JPEG-LS, un estándar para la compresión de imagen sin pérdida. Una reducción en la complejidad que se proporciona por estas técnicas es la eliminación de las tablas de Huffman en la línea de base JPEG, descrita como un cuello de botella en las implementaciones de hardware.

El documento EP 1,653,746 A2 de la Corporación Microsoft se relaciona con la compresión sin pérdidas de datos enteros mediante el uso de una técnica adaptativa hacia atrás. El procedimiento de códec adaptativo de longitud de ejecución y Golomb/Rice (RLGR) cambia entre un modo de codificador Golomb/Rice (G/R) únicamente y mediante el uso del codificador G/R combinado con un codificador de longitud de ejecución. La técnica adaptativa hacia atrás incluye reglas de adaptación que ajustan los parámetros del codificador después de cada símbolo codificado.

Sumario

En general, esta divulgación describe técnicas para actualizar un valor de un parámetro de Rice usado para definir códigos de Golomb para la codificación de nivel de coeficiente. En particular, esta divulgación describe técnicas para actualizar el valor del parámetro de Rice usado para definir códigos de Golomb para codificar los valores absolutos restantes de los niveles de coeficiente para coeficientes donde la codificación aritmética binaria adaptativa al contexto (CABAC) se usa para codificar indicaciones de coeficientes significativos, niveles de coeficiente mayor que 1 y niveles de coeficiente mayor que 2. En algunos ejemplos, las técnicas pueden aplicarse a una actualización del parámetro de Rice para la codificación de nivel de coeficiente en un rango de extensiones para el estándar de codificación de vídeo de alta eficiencia (HEVC). Los códigos de Golomb definidos por el parámetro de Rice pueden usarse para codificar un valor absoluto restante de un nivel de coeficiente para al menos un coeficiente en un grupo de coeficientes (CG), es decir, un bloque de coeficientes. El CG puede residir dentro de un bloque de transformación de una unidad de transformación (TU).

La invención es un proceso de coeficientes de decodificación como se reivindicó en la reivindicación 1, un procedimiento de coeficientes de codificación como se reivindicó en la reivindicación 9, un dispositivo de decodificación de vídeo como se reivindicó en la reivindicación 10, un dispositivo de codificación de vídeo como se reivindicó en la reivindicación 11 y un medio de almacenamiento legible por ordenador como se reivindicó en la reivindicación 12.

Los detalles de uno o más ejemplos se establecen en los dibujos acompañantes y en la descripción de más abajo. Otras características, objetos y ventajas de la invención serán evidentes a partir de la descripción y los dibujos y de las reivindicaciones dependientes.

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede utilizar las técnicas para codificar niveles de coeficiente descritos en esta divulgación.

La Figura 2 es un dibujo conceptual que muestra ejemplos de órdenes de escaneo inversas para la codificación de nivel de coeficiente.

La Figura 3 es un dibujo conceptual que muestra un ejemplo de orden de escaneo diagonal inversa basada en subbloques para la codificación de nivel de coeficiente de los grupos de coeficientes (CG).

La Figura 4 es un dibujo conceptual que muestra un ejemplo de orden de escaneo diagonal inversa para codificar niveles de coeficiente para un CG.

La Figura 5 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas para codificación de niveles de coeficiente descritos en esta divulgación.

La Figura 6 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo que puede implementar las técnicas para decodificación de niveles de coeficiente descritos en esta divulgación.

La Figura 7 es un diagrama de flujo que ilustra una operación de ejemplo de actualización de un valor de un parámetro de Rice durante la codificación de entropía de niveles de coeficiente de acuerdo con las técnicas descritas en esta divulgación.

La Figura 8 es un diagrama de flujo que ilustra una operación de ejemplo de actualización de un valor de un parámetro de Rice durante la decodificación de entropía de niveles de coeficiente de acuerdo con las técnicas descritas en esta divulgación.

La Figura 9 es un diagrama de flujo que ilustra una operación de ejemplo de ajuste de un valor de un parámetro de Rice durante la codificación de entropía de niveles de coeficiente de acuerdo con las técnicas descritas en esta divulgación.

La Figura 10 es un diagrama de flujo que ilustra una operación de ejemplo de ajuste de un valor de un parámetro de Rice durante la decodificación de entropía de niveles de coeficiente de acuerdo con las técnicas descritas en esta divulgación.

Descripción detallada

Esta divulgación describe técnicas para actualizar un valor de un parámetro de Rice usado para definir códigos de Golomb para la codificación de nivel de coeficiente. En particular, esta divulgación describe técnicas para actualizar el valor del parámetro de Rice usado para definir códigos de Golomb para codificar los valores absolutos restantes de los niveles de coeficiente para coeficientes donde la codificación aritmética binaria adaptativa al contexto (CABAC) se usa para codificar indicaciones de coeficientes significativos, niveles de coeficiente mayor que 1 y niveles de coeficiente mayor que 2. En algunos ejemplos, las técnicas pueden aplicarse a una actualización del

parámetro de Rice para la codificación de nivel de coeficiente en un rango de extensiones para el estándar de codificación de vídeo de alta eficiencia (HEVC). El parámetro de Rice es un valor sintonizable usado para seleccionar un conjunto de palabras de código de la familia de códigos de Golomb. Los códigos de Golomb definidos por el parámetro de Rice pueden usarse para codificar un valor absoluto restante de un nivel de coeficiente para al menos un coeficiente en un grupo de coeficientes (CG), es decir, un bloque de coeficientes.

De acuerdo con las técnicas de esta divulgación, el valor del parámetro de Rice se actualiza en base a una selección de un mínimo o de un valor máximo del parámetro de Rice o un incremento de la variable del valor del parámetro de Rice. La cantidad del incremento de la variable se determina en base al valor del parámetro de Rice y un valor absoluto del nivel de coeficiente para el coeficiente actual que se codifica. En algunos ejemplos, el incremento de la variable puede determinarse en base a una versión desplazada a la derecha del valor absoluto del nivel de coeficiente para el coeficiente actual, donde una cantidad del desplazamiento a la derecha se basa en el valor del parámetro de Rice.

El incremento de la variable permite aumentar el valor del parámetro de Rice en cualquier valor entero hasta el valor máximo del parámetro de Rice. Por ejemplo, el valor máximo del parámetro de Rice puede ser igual a 7 y el incremento de la variable puede ser igual a 0, 1, 2, 3, 4, 5 o 6 en función del valor del parámetro de Rice y el valor absoluto del nivel de coeficiente para el coeficiente actual. En otros ejemplos, el valor máximo del parámetro de Rice puede ser igual a 4 o 5. En otro ejemplo, las técnicas de esta divulgación pueden permitir que el parámetro de Rice se disminuya o decremente durante una actualización.

En algunos ejemplos, un codificador de vídeo puede codificar niveles de coeficiente de datos de vídeo residuales en un flujo de bits para su transmisión a un decodificador de vídeo o a un dispositivo de almacenamiento. Tras la recepción del flujo de bits codificado, el decodificador de vídeo puede decodificar los niveles de coeficiente de los datos de vídeo residuales de manera recíproca al codificador de vídeo. En el caso de codificación de vídeo con pérdida, los coeficientes pueden ser coeficientes de transformación cuantificados. En el caso de codificación de vídeo sin pérdida o codificación de vídeo con pérdida con salto o derivación de transformación, los coeficientes pueden tener niveles de coeficiente con valores absolutos grandes. Cuando los coeficientes representan el contenido de la pantalla, que puede incluir gráficos y regiones de texto, el contenido puede no ser bien predicho, lo que resulta en valores absolutos grandes de niveles de coeficiente para los coeficientes.

El esquema de actualización del parámetro de Rice descrito en esta divulgación permite que el parámetro de Rice se adapte rápidamente y eficientemente a grandes valores de coeficientes, que pueden ocurrir en bloques de contenido de la pantalla y bloques con salto o derivación de transformación. De acuerdo con las técnicas, el valor del parámetro de Rice puede actualizarse en base a un incremento de la variable con el fin de adaptarse más rápidamente para acomodar el contenido de la pantalla y/o los coeficientes que no se han transformado o cuantificado, pero que aún manejan la codificación de nivel de coeficiente de los coeficientes de transformación para contenido natural con un rendimiento similar al de un esquema de actualización del parámetro de Rice del HEVC. Por ejemplo, cuando un nivel de coeficiente para un coeficiente actual que se codifica tiene un valor absoluto grande, el incremento de la variable puede ser igual a un valor entero mayor que uno con el fin de adaptarse más rápidamente a los grandes valores de coeficiente. Las técnicas de esta divulgación también proporcionan una mayor eficiencia computacional al reducir o eliminar las operaciones condicionales durante las actualizaciones del parámetro de Rice dentro del bucle de codificación de entropía de nivel de coeficiente.

Además, esta divulgación describe técnicas para ajustar el valor del parámetro de Rice usado para definir códigos de Golomb para la codificación de nivel de coeficiente en base a si se aplica una transformación a un bloque de transformación que incluye el CG. Como se describió en esta divulgación, ajustar el valor del parámetro de Rice puede incluir actualizar el valor del parámetro de Rice para definir los códigos de Golomb para codificar un valor absoluto restante de un nivel de coeficiente para un coeficiente posterior en el CG, o inicializar el valor del parámetro de Rice para definir los códigos de Golomb para codificar un valor absoluto restante de un nivel de coeficiente para al menos un coeficiente en un CG posterior.

De acuerdo con las técnicas, si la transformación no se aplica al bloque de transformación, es decir, en el caso de salto o derivación de transformación, el esquema de actualización del parámetro de Rice descrito en esta divulgación se usa para actualizar el valor del parámetro de Rice. De cualquier otra manera, si la transformación se aplica al bloque de transformación, el esquema de actualización del parámetro de Rice del HEVC puede usarse para actualizar el valor del parámetro de Rice. En otro ejemplo, de acuerdo con las técnicas, si la transformación no se aplica al bloque de transformación, el valor del parámetro de Rice puede inicializarse para un CG posterior en el bloque de transformación en base al valor del parámetro de Rice después de la decodificación de los valores absolutos restantes de los niveles de coeficiente para los coeficientes en el CG actual. De cualquier otra manera, si la transformación se aplica al bloque de transformación, el valor del parámetro de Rice puede inicializarse para que sea igual a cero para un CG posterior en el bloque de transformación.

La Figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede utilizar las técnicas para codificar niveles de coeficiente descritos en esta divulgación. Como se muestra en la Figura 1, el sistema incluye el dispositivo fuente 12 que transmite vídeo codificado al dispositivo de

destino 14 a través del canal de comunicación 16. Los datos de vídeo codificados también pueden almacenarse en el medio de almacenamiento 34 o en el servidor de archivos 36 y puede accederse a ellos mediante el dispositivo de destino 14 como se desee. Cuando se almacena en un medio de almacenamiento o servidor de archivos, el codificador de vídeo 20 puede proporcionar datos de vídeo codificados a otro dispositivo, como una interfaz de red, un disco compacto (CD), Blu-ray o dispositivo de grabación o instalación de estampación de discos de vídeo digitales (DVD), u otros dispositivos, para almacenar los datos de vídeo codificados en el medio de almacenamiento. Igualmente, un dispositivo separado del decodificador de vídeo 30, como una interfaz de red, lector de CD o DVD, o similar, puede recuperar datos de vídeo codificados de un medio de almacenamiento y proporcionar los datos recuperados al decodificador de vídeo 30.

El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia variedad de dispositivos, que incluyen ordenadores de escritorio, notebook (es decir, ordenador portátil), tabletas, cajas decodificadoras, teléfonos móviles como los llamados teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de vídeo juegos o similares. En muchos casos, estos dispositivos pueden equiparse para la comunicación inalámbrica. Por tanto, el canal de comunicación 16 puede comprender un canal inalámbrico, un canal alámbrico o una combinación de canales inalámbricos y alámbricos adecuados para la transmisión de datos de vídeo codificados. De manera similar, al servidor de archivos 36 puede accederse por el dispositivo de destino 14 a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem por cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos.

Las técnicas para codificar los niveles de coeficiente, de acuerdo con los ejemplos de esta divulgación, pueden aplicarse a la codificación de vídeo para soportar cualquiera de una variedad de aplicaciones multimedia, como la difusión de televisión por aire, las transmisiones de televisión por cable, las transmisiones de televisión por satélite, las transmisiones de vídeo en tiempo real, 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 puede configurarse para soportar transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones como la transmisión de vídeo en tiempo real, la reproducción de vídeo, la difusión de vídeo y/o vídeo telefonía.

En el ejemplo de la Figura 1, el dispositivo fuente 12 incluye una fuente de vídeo 18, un codificador de vídeo 20, un modulador/demodulador 22 y un transmisor 24. En el dispositivo fuente 12, la fuente de vídeo 18 puede incluir una fuente como un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contenga vídeo capturado previamente, una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo, y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como la fuente del vídeo, o una combinación de dichas fuentes. Como un ejemplo, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o teléfonos de vídeo, que pueden proporcionarse, por ejemplo, dentro de teléfonos inteligentes o tabletas. Sin embargo, las técnicas descritas en esta divulgación pueden aplicarse a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o alámbricas, o aplicaciones en las que los datos de vídeo codificados se almacenan en un disco local.

El vídeo capturado, precapturado o generado por ordenador puede codificarse por el codificador de vídeo 20. La información de vídeo codificado puede modularse por el módem 22 de acuerdo con un estándar de comunicación, como un protocolo de comunicación alámbrico o inalámbrico, y transmitirse al dispositivo de destino 14 a través del transmisor 24. El módem 22 puede incluir varios mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de la señal. El transmisor 24 puede incluir circuitos diseñados para transmitir datos, incluidos amplificadores, filtros y, en el caso de la comunicación inalámbrica, una o más antenas.

El vídeo capturado, precapturado o generado por ordenador que se codifica por el codificador de vídeo 20 también puede almacenarse en el medio de almacenamiento 34 o en el servidor de archivos 36 para su consumo posterior. El medio de almacenamiento 34 puede incluir discos Blu-ray, DVD, CD-ROM, memoria flash o cualquier otro medio de almacenamiento digital adecuado para almacenar vídeo codificado. El vídeo codificado almacenado en el medio de almacenamiento 34 puede entonces accederse por el dispositivo de destino 14 para decodificación y reproducción. Aunque no se muestra en la figura 1, en algunos ejemplos, el medio de almacenamiento 34 y/o el servidor de archivos 36 puede almacenar la salida del transmisor 24.

El servidor de archivos 36 puede ser cualquier tipo de servidor capaz de almacenar vídeo codificado y transmitir ese vídeo codificado al dispositivo de destino 14. Ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS), una unidad de disco local o cualquier otro tipo de dispositivo capaz de almacenar datos de vídeo codificados y transmitirlos a un dispositivo de destino. La transmisión de datos de vídeo codificados desde el servidor de archivos 36 puede ser una transmisión en tiempo real, una transmisión de descarga o una combinación de ambas. Al servidor de archivos 36 puede accederse por el dispositivo de destino 14 a través de cualquier conexión de datos estándar, incluida una conexión a internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión

alámbrica (por ejemplo, DSL, módem por cable, Ethernet, USB, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos.

El dispositivo de destino 14, en el ejemplo de la Figura 1, incluye el receptor 26, el módem 28, el decodificador de vídeo 30 y el dispositivo de visualización 32. El receptor 26 del dispositivo de destino 14 recibe información por el canal 16, y el módem 28 demodula la información para producir un flujo de bits demodulado para el decodificador de vídeo 30. La información comunicada por el canal 16 puede incluir una variedad de información de sintaxis generada por el codificador de vídeo 20 para su uso por el decodificador de vídeo 30 en la decodificación de datos de vídeo. Tal sintaxis también puede incluirse con los datos de vídeo codificados almacenados en el medio de almacenamiento 34 o el servidor de archivos 36. Cada uno de los codificadores de vídeo 20 y decodificadores de vídeo 30 pueden formar parte de un respectivo codificador-decodificador (CODEC) que es capaz de codificación o decodificación de datos de vídeo.

El dispositivo de visualización 32 puede integrarse con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también puede configurarse como interfaz con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico emisor de luz (OLED), u otro tipo de dispositivo de visualización.

En el ejemplo de la Figura 1, el canal de comunicación 16 puede comprender cualquier medio de comunicación inalámbrico o alámbrico, como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas, o cualquier combinación de medios inalámbricos y alámbricos. El canal de comunicación 16 puede formar parte de una red basada en paquetes, como una red de área local, una red de área amplia o una red global como Internet. El canal de comunicación 16 generalmente representa cualquier medio de comunicación adecuado, o colección de diferentes medios de comunicación, para transmitir datos de vídeo desde el dispositivo fuente 12 al dispositivo de destino 14, que incluye cualquier combinación adecuada de medios alámbricos o inalámbricos. El canal de comunicación 16 puede incluir enrutadores, conmutadores, estaciones base, o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 12 al dispositivo de destino 14.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con un estándar de compresión de vídeo, como el estándar de codificación de vídeo de alta eficiencia (HEVC) desarrollado por el equipo de colaboración conjunta en codificación de vídeo (JCT-VC) del grupo de expertos en codificación de vídeo de ITU-T (VCEG) e ISO/IEC del grupo de expertos de imagen en movimiento (MPEG). Un borrador del estándar HEVC en el documento JCTVC-L1003v34, Bross y otros, "High efficiency video coding (HEVC) text specification draft 10," equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12a reunión: Ginebra, Suiza, del 14-23 de enero de 2013, disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip.

Aunque no se muestra en la Figura 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden integrar cada uno con un codificador y un decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación de ambos, audio y vídeo en un flujo de datos común o flujos de datos separados. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo multiplexor ITU H.223 u otros protocolos como el protocolo de datagrama de usuario (UDP).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores adecuados, como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matriz de puertas lógicas programables en campo (FPGA), lógica discreta, software, hardware, microprograma o cualquiera de sus combinaciones. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador y ejecutar las instrucciones en hardware mediante el uso de uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno de los codificadores de vídeo 20 y el decodificador de vídeo 30 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/descodificador combinado (CODEC) en un dispositivo respectivo.

El codificador de vídeo 20 puede implementar cualquiera o todas las técnicas de esta divulgación para la codificación de niveles de coeficiente en un procedimiento de codificación de vídeo. Igualmente, el decodificador de vídeo 30 puede implementar cualquiera o todas estas técnicas para la decodificación de niveles de coeficiente en un proceso de decodificación de vídeo. Un codificador de vídeo, como se describió en esta divulgación, puede referirse a un codificador de vídeo o un decodificador de vídeo. De manera similar, una unidad de codificación de vídeo puede referirse a un codificador de vídeo o un decodificador de vídeo. Igualmente, la codificación de vídeo puede referirse a la codificación o decodificación de vídeo.

Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo para codificar y decodificar información de vídeo digital más eficientemente. La compresión de vídeo puede aplicar técnicas de predicción espacial (dentro del cuadro) y/o de predicción temporal (entre cuadros) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. El estándar HEVC descrito anteriormente es en base a un modelo evolutivo de un dispositivo de codificación de vídeo que se refiere como modelo de prueba HEVC (HM). El HM presupone varias capacidades adicionales de los dispositivos de codificación de vídeo con relación 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 intrapredicción, el HEVC HM puede proporcionar hasta treinta y tres modos de codificación intrapredicción. En la siguiente sección se analizan con más detalles ciertos aspectos de la HM.

Para la codificación de vídeo de acuerdo con el estándar HEVC, un fotograma de vídeo puede particionarse en unidades de codificación. Una unidad de codificación (CU) generalmente se refiere a una región de imagen que sirve como unidad básica a la que se aplican diversas herramientas de codificación para la compresión de vídeo. Una CU generalmente tiene un componente de luminancia, denotado como Y, y dos componentes de crominancia, denotados como U y V. En función del formato de muestreo de vídeo, el tamaño de los componentes U y V, en términos de número de muestras, puede ser el mismo o diferente al tamaño del componente Y.

Una CU es típicamente cuadrada y puede considerarse similar a un llamado macrobloque, por ejemplo, bajo otros estándares de codificación de vídeo como ITU-T H.264. La codificación de acuerdo con algunos de los aspectos actualmente propuestos del estándar en desarrollo HEVC se describirá en la presente solicitud con propósitos ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otro procedimiento de codificación de vídeo, como los definidos de acuerdo con H.264 u otro procedimiento de codificación de vídeo estándar o propietario.

De acuerdo con el HM, una CU puede incluir una o más unidades de predicción (PU) y/o una o más unidades de transformación (TU). Las sintaxis de datos dentro de un flujo de bits pueden definir una unidad de codificación más grande (LCU), que es una CU más grande en términos de número de píxeles. En general, una CU tiene un propósito similar a un macrobloque de H.264, excepto que una CU no tiene una distinción de tamaño. Por lo tanto, una CU puede dividirse en sub-CU. En general, las referencias en esta divulgación a una CU pueden referirse a una unidad de codificación más grande de una imagen o una sub-CU de una LCU. Una LCU puede dividirse en sub-CU, y cada sub-CU puede dividirse, además, en sub-CU. Las sintaxis de datos para un flujo de bits pueden definir un número máximo de veces que puede dividirse una LCU, que se refiere como profundidad de CU. En consecuencia, un flujo de bits también puede definir una unidad de codificación más pequeña (SCU). Esta divulgación también usa el término "bloque" o "porción" para referirse a cualquiera de una CU, PU o TU. En general, "porción" puede referirse a cualquier subconjunto de un fotograma de vídeo.

Una LCU puede asociarse con una estructura de datos de árbol cuaternario. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, donde un nodo raíz corresponde a la LCU. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU. Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar sintaxis de datos para la correspondiente CU. Por ejemplo, un nodo en el árbol cuaternario puede incluir una bandera de división, que indica si la CU correspondiente al nodo se divide en sub-CU. Los elementos de sintaxis para una CU pueden definirse recursivamente y pueden depender de si la CU se divide en sub-CU. Si una CU no se divide más, se refiere a ella como una CU hoja. En esta divulgación, 4 sub-CU de una CU hoja también puede referirse como CU hojas, aunque no hay una división explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 16x16 no se divide más, las cuatro 8x8 sub-CU también puede referirse como CU hoja, aunque la CU de 16x16 nunca se dividió.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa toda o una porción de la CU correspondiente y puede incluir datos para recuperar una muestra de referencia para la PU. Por ejemplo, cuando la PU se codifica intermodo, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), un fotograma de referencia al que apunta el vector de movimiento, y/o una lista de referencia (por ejemplo, lista 0 o lista 1) para el vector de movimiento. Los datos para la CU hoja que define las PU también pueden describir, por ejemplo, la partición de la CU en una o más PU. Los modos de particionamiento pueden diferir en función de si la CU no se codifica, se codifica en modo de intrapredicción o se codifica en modo interpredicción. Para la intracodificación, una PU puede tratarse igual que una unidad de transformación de hoja descrita más abajo.

El estándar emergente HEVC permite transformaciones de acuerdo con unidades de transformación (TU), que pueden ser diferentes para diferentes CU. Las TU se dimensionan típicamente en base al tamaño de las PU dentro de una CU dada definida por una LCU que se particiona, aunque esto puede no ser siempre el caso. Las TU son típicamente del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas mediante el uso de una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT pueden referirse como unidades de transformación (TU). Los valores de diferencia de píxeles asociados con las TU pueden transformarse

para producir coeficientes de transformación, que pueden cuantificarse. Una TU incluye un bloque de transformación luma y dos bloques de transformación croma. Como tal, cualquier procedimiento de codificación descrito más abajo que se aplica a una TU puede, en realidad, aplicarse a los bloques de transformación luma y croma.

5 En general, una PU se refiere a datos relacionados con el procedimiento de predicción. Por ejemplo, cuando la PU se codifica intramodo, la PU puede incluir datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU se codifica intermodo, la PU puede incluir datos que definen un vector de movimiento para la PU.

10 En general, una TU se usa para los procedimientos de transformación y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Tras la predicción, el codificador de vídeo 20 puede calcular los valores residuales del bloque de vídeo identificado por el nodo de codificación de acuerdo con la PU. El nodo de codificación se actualiza entonces para hacer referencia a los valores residuales en lugar del bloque de vídeo original. Los valores residuales comprenden valores de diferencia de píxeles que pueden transformarse en coeficientes de transformación, cuantificarse y escanearse mediante el uso de las transformaciones y otra información de transformación especificada en las TU para producir coeficientes de transformación serializados para la codificación de entropía. El nodo de codificación puede actualizarse de nuevo para hacer referencia a estos coeficientes de transformación serializados. Esta divulgación usa típicamente el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término "bloque de vídeo" para referirse a un bloque de árbol, es decir, LCU o un CU, que incluye un nodo de codificación y PU y TU.

Una secuencia de vídeo típicamente incluye una serie de fotogramas de vídeo o imágenes. Un grupo de imágenes (GOP) generalmente comprende una serie de una o más de las imágenes de vídeo. Un GOP puede incluir sintaxis de datos en un encabezado del GOP, un encabezado de una o más de las imágenes, o en otro lugar, que describe un número de imágenes incluidas en el GOP. Cada segmento de una imagen puede incluir sintaxis de datos de segmento que describen un modo de codificación para el segmento respectivo. El codificador de vídeo 20 típicamente opera en bloques de vídeo dentro de segmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño de acuerdo con un estándar de codificación específico.

Para codificar un bloque (por ejemplo, una unidad de predicción de datos de vídeo), primero se deriva un predictor para el bloque. El predictor, se refiere también como un bloque predictivo, que puede derivarse o mediante intrapredicción (I) (es decir, predicción espacial) o interpredicción (P o B) (es decir, predicción temporal). Por lo tanto, algunas unidades de predicción pueden intracodificarse (I) mediante el uso de predicción espacial con respecto a muestras de referencia en bloques de referencia vecinos en el mismo fotograma (o segmento), y otras unidades de predicción pueden intercodificarse unidireccionalmente (P) o intercodificarse bidireccionalmente (B) con respecto a bloques de muestras de referencia en otros fotogramas (o segmentos) previamente codificados. En cada caso, las muestras de referencia pueden usarse para formar un bloque predictivo para un bloque a codificar.

Tras la identificación de un bloque predictivo, se determina la diferencia entre los píxeles en el bloque de datos de vídeo original y los píxeles en su bloque predictivo. Esta diferencia puede referirse como datos residuales de predicción e indica las diferencias de píxeles entre los valores de píxeles en el bloque a codificar y los valores de píxeles en el bloque predictivo seleccionado para representar el bloque codificado. Para lograr una mejor compresión, los datos residuales de predicción pueden transformarse, por ejemplo, mediante el uso de una transformación de coseno discreta (DCT), una transformación de seno discreta (DST), una transformación de entero, una transformación de Karhunen-Loeve (KL) u otra transformación para producir coeficientes de transformación.

50 Los datos residuales en un bloque de transformación, como una TU, pueden disponerse en una matriz bidimensional (2D) de valores de diferencia de píxeles que residen en el dominio espacial de píxeles. Una transformación convierte los valores de píxeles residuales en una matriz bidimensional de coeficientes de transformación en un dominio de transformación, como un dominio de frecuencia. Para una mayor compresión, los coeficientes de transformación pueden cuantificarse antes de la codificación de entropía. En algunos ejemplos, como la codificación sin pérdida o la codificación con pérdida con salto o derivación de la transformación, ambos procedimientos de transformación y cuantificación pueden saltarse por los coeficientes.

Un codificador de entropía aplica codificación de entropía, como codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación de entropía de particiones de intervalo de probabilidad (PIPE), o similares, a los coeficientes. En algunos ejemplos, el codificador de vídeo 20 puede utilizar una orden de escaneo predefinido para escanear los coeficientes para producir un vector serializado que puede codificarse por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneo adaptativo. Después de escanear los coeficientes para formar un vector unidimensional, el codificador de vídeo 20 puede codificar la entropía en el vector unidimensional, el codificador de vídeo 20 también puede codificar la entropía de elementos de sintaxis asociados con los datos de vídeo codificados para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

Esta divulgación se relaciona con técnicas para codificadores de entropía de codificación aritmética binaria adaptativa al contexto (CABAC) u otros codificadores de entropía, como codificación de entropía de particiones de intervalo de probabilidad (PIPE) o codificadores relacionados. La codificación aritmética es una forma de codificación de entropía usada en muchos algoritmos de compresión que tienen una alta eficiencia de codificación, porque es capaz de asignar símbolos a palabras de código de longitud no entera. Un ejemplo de un algoritmo de codificación aritmética es la codificación aritmética binaria adaptativa al contexto (CABAC).

En general, codificación de entropía de los símbolos de datos mediante el uso de CABAC implica una o más de las siguientes etapas:

- (1) Binarización: Si un símbolo a codificar tiene un valor no binario, es asignado a una secuencia de los llamados "contenedores". Cada contenedor puede tener un valor de "0" o "1".
- (2) Asignación de contexto: Cada contenedor (en modo normal) se asigna a un contexto. Un modelo de contexto determina cómo se calcula un contexto para un contenedor dado en base a la información disponible para el contenedor, como los valores de los símbolos previamente codificados o el número de contenedor.
- (3) Codificación de contenedor: Los contenedores se codifican con un codificador aritmético. Para codificar un contenedor, el codificador aritmético requiere como entrada una probabilidad del valor del contenedor, es decir, una probabilidad de que el valor del contenedor sea igual a "0" y una probabilidad de que el valor del contenedor sea igual a "1". La probabilidad estimada de cada contexto se representa por un valor entero llamado "estado de contexto". Cada contexto tiene un estado y, por tanto, el estado (es decir, la probabilidad estimada) es el mismo para los contenedores asignados a un contexto y difiere entre los contextos.
- (4) Actualización de estado: El estado de probabilidad para un contexto seleccionado se actualiza en base al valor codificado real del contenedor (por ejemplo, si el valor del contenedor era "1", la probabilidad de "1" aumenta).

En el caso de codificación de entropía de símbolos de datos mediante el uso de CABAC en modo de derivación, un símbolo a codificar se binariza en una secuencia de contenedores y se codifica aritméticamente con un modelo fijo de igual probabilidad (por ejemplo, con un código Exponencial-Golomb o un código Golomb-Rice). El modo de derivación no requiere asignación de contexto o actualización del estado de probabilidad. Por ejemplo, esta divulgación describe técnicas para la codificación de derivación de los valores absolutos restantes de los niveles de coeficiente para los coeficientes mediante el uso de códigos de Golomb definidos por un parámetro de Rice. Se debe señalar que la codificación de entropía de particiones de intervalo de probabilidad (PIPE) utiliza principios similares a los de la codificación aritmética y, por lo tanto, también puede utilizar las técnicas de esta divulgación.

CABAC en H.264/AVC y HEVC usa estados, y cada estado está implícitamente relacionado con una probabilidad. Existen variantes de CABAC, en las que se usa directamente la probabilidad de un símbolo ("0" o "1"), es decir, la probabilidad o una versión entera de la probabilidad es el estado. Por ejemplo, estas variantes de CABAC se describen en "Description of video coding technology proposal by France Telecom, NTT, NTT DOCOMO, Panasonic and Technicolor," JCTVC-A114, 1ra reunión del JCT-VC, Dresde, Alemania, abril de 2010, que se refiere como "JCTVC-A114" en adelante, y A. Alshin y E. Alshina, "Multi-parameter probability update for CABAC," JCTVC-F254, 6ta reunión del JCT-VC, Turín, Italia, julio de 2011, que se refiere como "JCTVC-F254" en adelante.

Para codificar la entropía de un bloque de coeficientes, ya sea transformado y cuantificado o ni transformado ni cuantificado, generalmente se realiza un procedimiento de escaneo para que la matriz bidimensional (2D) de coeficientes en un bloque se reorganice, de acuerdo con un orden de escaneo particular, en una matriz ordenada, unidimensional (1D), es decir, un vector, de coeficientes. A continuación, se aplica la codificación de entropía al vector de coeficientes. El escaneo de los coeficientes en una unidad de transformación serializa la matriz 2D de coeficientes para el codificador de entropía. Puede generarse un mapa de importancia para indicar las posiciones de los coeficientes significativos (es decir, distintos de cero). El escaneo puede aplicarse a niveles de escaneo de coeficientes significativos (es decir, distintos de cero) y/o para codificar los signos de los coeficientes significativos.

En el estándar HEVC, la información de posición de la transformación significativa (por ejemplo, el mapa de importancia) se codifica primero para un bloque de transformación para indicar la ubicación del último coeficiente distinto de cero en la orden de escaneo. El mapa de importancia y la información de nivel (es decir, los valores absolutos y los signos de los coeficientes) se codifican para cada coeficiente en una orden de escaneo inversa.

La Figura 2 es un dibujo conceptual que muestra ejemplos de órdenes de escaneo inversas para la codificación de nivel de coeficiente. El estándar H.264 define un escaneo en zigzag. El estándar HEVC define tres escaneos diferentes: un escaneo diagonal de subbloque, un escaneo horizontal de subbloque y un escaneo vertical de subbloque. La Figura 2 ilustra un patrón de escaneo en zigzag inverso 29, un patrón de escaneo vertical inverso 31, un patrón de escaneo horizontal inverso 33 y un patrón de escaneo diagonal inverso 35 aplicados cada uno a un subbloque de 8x8 de un bloque de transformación. Tenga en cuenta que cada patrón de escaneo diagonal inverso 35, patrón de escaneo de zigzag inverso 29, patrón de escaneo vertical inverso 31 y patrón de escaneo horizontal inverso 33 proceden de coeficientes de frecuencia más altos en la esquina inferior derecha del bloque de transformación a coeficientes de frecuencia más bajos en la esquina superior izquierda del bloque de transformación.

El patrón de escaneo diagonal de subbloque 35, el patrón de escaneo horizontal de subbloque 33 y el patrón de escaneo vertical de subbloque 31 pueden aplicarse para bloques de transformación de 4×4 y 8×8 . El patrón de escaneo diagonal 35 de subbloque también puede aplicarse para bloques de transformación de 16×16 y 32×32 en el estándar HEVC. En algunos ejemplos, el patrón de escaneo diagonal de subbloque 35 también puede aplicarse a una TU de 8×8 . En un escaneo basado en subbloques, se escanea un subbloque de 4×4 de un bloque de transformación más grande antes de pasar a otro subbloque de 4×4 dentro del bloque de transformación más grande. En otros ejemplos, un "subbloque" puede consistir de un número de coeficientes escaneados consecutivamente de acuerdo con la orden de escaneo usada. Por ejemplo, el "subbloque" puede constar de 16 coeficientes escaneados consecutivamente a lo largo de una orden de escaneo diagonal.

La Figura 3 es un dibujo conceptual que muestra un ejemplo de orden de escaneo diagonal inversa basada en subbloques para la codificación de nivel de coeficiente de los grupos de coeficientes (CG). La Figura 3 ilustra un bloque de transformación 8×8 que consta de cuatro subbloques de 4×4 (37A, 37B, 37C, 37D). Como se muestra en la Figura 3, los coeficientes en el subbloque 37D se escanean antes de escanear los coeficientes en el subbloque 37C. El escaneo procede entonces del subbloque 37C al subbloque 37B, y finalmente al subbloque 37A. La Figura 3 representa una orden de escaneo diagonal inversa en cada subbloque. En otros ejemplos, puede usarse cualquier orden de escaneo (por ejemplo, horizontal, vertical, zigzag, etc.). En algunos ejemplos, pueden usarse órdenes de escaneo hacia adelante dentro de cada subbloque.

En el estándar HEVC, los coeficientes pueden agruparse en un fragmento o subconjunto. El mapa de importancia y la información de nivel (es decir, el valor absoluto y el signo) de los coeficientes se codifican para cada subconjunto. El subconjunto de coeficientes puede referirse en esta divulgación como grupo de coeficiente (CG). Un CG puede definirse como coeficientes consecutivos 16 en una orden de escaneo para un bloque de transformación, que puede corresponder a un subbloque de 4×4 . En un ejemplo, un subconjunto consta de coeficientes consecutivos 16 a lo largo de una orden de escaneo (por ejemplo, una orden de escaneo diagonal, horizontal o vertical hacia adelante o inversa) para un bloque de transformación de 4×4 y un bloque de transformación de 8×8 . Para bloques de transformación de 16×16 y 32×32 , un subbloque de coeficientes de 4×4 dentro del bloque de transformación más grande se trata como un subconjunto. En el ejemplo de la Figura 3, cada uno de los subbloques 37 puede ser un CG.

Los símbolos que se describen más abajo se codifican para representar la información a nivel de coeficiente dentro de un CG. En un ejemplo, todos los símbolos se codifican en una orden de escaneo inversa. Cada uno de los símbolos puede codificarse en un escaneo separado del CG de acuerdo con la orden de escaneo inversa. Los siguientes símbolos pueden referirse como "banderas". Se debe señalar que cualquiera de las "banderas" descritas en esta divulgación no necesita limitarse a un símbolo binario, sino que también puede implementarse como elementos de sintaxis de múltiples bits.

El coeficiente significativo bandera, que se refiere también como sigMapFlag, indica la importancia de cada coeficiente en un subconjunto. Un coeficiente con un valor absoluto mayor que cero se considera significativo. Como un ejemplo, un valor de sigMapFlag de 0 indica que el coeficiente no es significativo, es decir, no mayor que cero, mientras que un valor de 1 indica que el coeficiente es significativo, es decir, mayor que cero. Esta bandera puede generalmente referirse como bandera de importancia. El coeficiente sign_flag, que se refiere también como signFlag, indica la información de signo para cualquier coeficiente distinto de cero (es decir, coeficientes con sigMapFlag como 1). Por ejemplo, un cero para esta bandera indica un signo positivo, mientras que un 1 indica un signo negativo.

El coeficiente abs_level_greater1_flag, que se refiere también como el gr1Flag, indica si el valor absoluto del coeficiente es más grande que uno para cualquier coeficiente distinto de cero (es decir, coeficientes con sigMapFlag como 1 o donde el sigMapFlag se deriva implícitamente como 1). Como un ejemplo, un valor de gr1Flag de 0 indica que el coeficiente no tiene un valor absoluto mayor que uno, mientras que un valor de 1 para el gr1Flag indica que el coeficiente tiene un valor absoluto mayor que uno. Esta bandera puede generalmente referirse como una bandera mayor que uno.

El coeff_abs_level_greater2_flag, que se refiere también como el gr2Flag, indica si el valor absoluto del coeficiente es más grande que dos para cualquier coeficiente con un valor absoluto más grande que uno (es decir, coeficientes con gr1Flag como 1). Como un ejemplo, un valor de gr2Flag de 0 indica que el coeficiente no tiene un valor absoluto mayor que dos, mientras que un valor de 1 para el gr2Flag indica que el coeficiente tiene un valor absoluto mayor que dos. Esta bandera puede generalmente referirse como una bandera mayor que dos. El sigMapFlag, gr1Flag y gr2Flag pueden codificarse cada uno mediante el uso de CABAC.

El elemento de sintaxis del coeficiente abs_level_remaining, que se refiere también como elemento de sintaxis levelRem, indica un valor absoluto restante del nivel de coeficiente para cualquier coeficiente con valores absolutos mayores que los valores codificados por las banderas anteriores. Generalmente, para el elemento de sintaxis levelRem, se codifica el valor absoluto del nivel de coeficiente menos tres (es decir, abs (nivel) - 3) para cada coeficiente que tiene un valor absoluto más grande que dos (es decir, coeficientes con gr2Flag como 1). En algunos ejemplos, cuando se alcanza un número máximo de gr1Flag y/o gr2Flag para el CG actual, el elemento de sintaxis

levelRem puede usarse para codificar niveles de coeficiente con valores absolutos menores o iguales a dos. El elemento de sintaxis levelRem puede codificarse mediante el uso de códigos de Golomb definidos por un valor de un parámetro de Rice.

La Figura 4 es un dibujo conceptual que muestra un ejemplo de orden de escaneo diagonal inversa para codificar niveles de coeficiente para un CG 39. El CG 39 puede ser un bloque de transformación de 4×4 o puede ser un subbloque de 4×4 en un bloque de transformación de 8×8 , 16×16 o 32×32 . Los símbolos codificados para los coeficientes mostrados en la Figura 4, escaneados en una orden de escaneo inversa, se resumen en la Tabla 1. En la Tabla 1, la posición de escaneo se refiere a la posición del coeficiente a lo largo del patrón de escaneo diagonal inverso del CG 39 mostrado en la Figura 4. La posición de escaneo 15 es el primer coeficiente escaneado y se ubica en la esquina inferior derecha del CG 39. El coeficiente en la posición de escaneo 15 tiene un valor absoluto de 0. La posición de escaneo 0 es el último coeficiente escaneado y se ubica en la esquina superior izquierda del CG 39. El coeficiente cuantificado en la posición de escaneo 0 tiene un valor absoluto de 10. En el caso de un bloque de transformación de 4×4 o el último subbloque de 4×4 de un bloque de transformación más grande, no es necesario codificar los primeros cuatro sigMapFlags, ya que se conoce la posición del último coeficiente distinto de cero. Es decir, la codificación del sigMapFlag puede comenzar en el último coeficiente distinto de cero (en este ejemplo, el coeficiente en la posición de escaneo 11).

Tabla 1. Símbolos codificados para los coeficientes de un grupo de coeficiente

Posición de escaneo	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Coeficientes	0	0	0	0	1	-1	0	2	0	3	2	-1	0	5	-7	10
sigMapFlag	0	0	0	0	1	1	0	1	0	1	1	1	0	1	1	1
gr1Flag					0	0		1		1	1	0		1	1	1
gr2Flag								0		1	0			1	1	1
signFlag					0	1		0		0	0	1		0	1	0
levelRem										0				2	4	7

Entre estos símbolos, los contenedores de sigMapFlag, gr1Flag y gr2Flag se codifican con modelos de contexto adaptativo, por ejemplo, mediante el uso de CABAC. El signFlag y los contenedores que se binarizan de levelRem se codifican a través del modo de derivación con un modelo fijo de igual probabilidad (por ejemplo, con un código Exponencial-Golomb o un código Golomb-Rice).

Como se ha descrito anteriormente, el elemento de sintaxis coeff_abs_level_remaining (es decir, levelRem) en el estándar HEVC indica el valor absoluto restante de un nivel de coeficiente para un coeficiente si el valor es más grande que el codificado en pasadas de escaneo anteriores para los coeficientes de codificación. Este elemento de sintaxis se codifica en modo de derivación para aumentar el rendimiento. El estándar HEVC emplea la codificación Golomb-Rice para valores pequeños y cambia a la codificación Exponencial-Golomb (Exp-Golomb) para valores más grandes. El punto de transición entre los códigos de Golomb-Rice y los códigos Exp-Golomb es cuando la longitud del código unario es igual a 4. Un parámetro de Rice es un valor sintonizable para seleccionar el conjunto de palabras de código de la familia de códigos de Golomb.

Por ejemplo, los códigos Golomb-Rice son un subconjunto de los códigos de Golomb y representan un valor $n \geq 0$, dado un parámetro de Rice sintonizable m , como un cociente $q = \text{suelo}(n/m)$ y un resto $r = n - q \times m$, donde m es una potencia de 2. El cociente q es el prefijo y tiene una representación de código unario. El resto r es el sufijo y tiene una representación de longitud fija. En los códigos Exp-Golomb, la estructura del código se forma de manera similar por un prefijo unario seguido de un sufijo de longitud fija, pero el número de palabras de código en la parte del sufijo se duplica después de cada bit en el código unario. Los códigos Exp-Golomb, por lo tanto, tienen un crecimiento más lento de la longitud de la palabra de código. En general, un valor más grande del parámetro de Rice resulta en un crecimiento más lento de los códigos, lo que permite una mayor eficiencia cuando se codifican grandes valores de coeficientes. Pueden encontrarse detalles adicionales sobre los parámetros de Rice en Transform Coefficient Coding in HEVC, por J. Sole, R. Joshi, M. Karczewicz, N. Nguyen, T. Ji, G. Clare, F. Henry, A. Duenas, IEEE Transactions on Circuits and Systems for video Transmission (edición especial en HEVC), diciembre de 2012.

En el estándar HEVC, el parámetro de Rice se establece igual a un valor de 0 al comienzo de cada grupo de coeficientes (CG) y se actualiza condicionalmente durante la codificación del CG en función del valor del parámetro de Rice y el valor absoluto del nivel de coeficiente para un coeficiente actual que se codifica como sigue:

Si $\text{absCoeffLevel} > 3 * 2^{\text{cRiceParam}}$, entonces $\text{cRiceParam} = \min(\text{cRiceParam} + 1, 4)$

De cualquier otra manera, $\text{cRiceParam} = \text{cRiceParam}$,

donde `cRiceParam` es el parámetro de Rice, `absCoeffLevel` es el valor absoluto del nivel de coeficiente para el coeficiente actual y `min()` es una función que selecciona un valor mínimo. El esquema de actualización del parámetro de Rice del HEVC permite que el procedimiento de binarización se adapte gradualmente a las estadísticas de coeficientes cuando se observan valores absolutos grandes en la distribución. Como puede verse en la ecuación anterior, el esquema de actualización del parámetro de Rice del HEVC actualiza el valor del parámetro de Rice en un incremento fijo de 1 o 0.

Un ejemplo del procedimiento de actualización del parámetro de Rice del HEVC en código C se muestra más abajo:

```
10     if (absCoeff > 3 * (1 << cRiceParam))
    {
        } cRiceParam = min<UInt>(cRiceParam + 1,4);
```

donde `cRiceParam` es el parámetro de Rice, `absCoeff` es el valor absoluto del nivel de coeficiente para el coeficiente actual, `min()` es una función que selecciona un valor mínimo y `<<` es la operación de desplazamiento de bits a la izquierda.

En el estándar HEVC, el procedimiento de actualización del parámetro de Rice del HEVC se describe como sigue.

La variable `cRiceParam` se deriva de `cLastAbsLevel` y `cLastRiceParam` como:

$$cRiceParam = \text{Min}(cLastRiceParam + (cLastAbsLevel > (3 * (1 << cLastRiceParam)) ? 1 : 0), 4)$$

En la ecuación anterior, `cRiceParam` es el valor actualizado del parámetro de Rice, `Min()` es una función que selecciona un valor mínimo, `cLastRiceParam` es el valor actual del parámetro de Rice y `cLastAbsLevel` es el valor absoluto del nivel de coeficiente para el coeficiente codificado actual. Se refieren a los parámetros mediante el uso del término "Último" porque el valor actualizado del parámetro de Rice puede usarse para definir los códigos de Golomb para codificar coeficientes posteriores en un CG. La "?" es la abreviatura de una declaración condicional "Si". Por ejemplo, puede considerarse que la ecuación anterior establece: si `cLastAbsLevel` es mayor que $(3 * (1 << cLastRiceParam))$, entonces `cLastRiceParam` se incrementa en 1, de cualquier otra manera `cLastRiceParam` no se cambia.

Técnicas propuestas en JCTVC-M0316, M. Budagavi, "AHG8: Coefficient level `cRiceParam` updates for screen content coding," equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 13ra Reunión: Incheon, Corea del Sur, del 18-26 de abril de 2013 (en adelante, JCTVC-M0316), describe un procedimiento de actualización más rápido para el parámetro de Rice. Las técnicas de JCTVC-M0316 apuntan a regiones de texto y gráficos, que tienen bordes afilados que a veces no son predichos bien por las herramientas de intrapredicción del HEVC. Como resultado del error de predicción, los valores de los coeficientes pueden tener altos aumentos de amplitud.

Las contribuciones de JCTVC-M0316 presentan dos modificaciones al procedimiento de actualización del parámetro de Rice para codificar el elemento de sintaxis coeficientes `abs_level_remaining` con el fin de mejorar la eficiencia de codificación sin pérdida en el estándar HEVC cuando se codifica vídeo de contenido de la pantalla. Las dos modificaciones son: (1) una actualización rápida para el valor del parámetro de Rice y (2) un aumento en el valor máximo del parámetro de Rice de 4 a 5.

Un ejemplo del código C modificado para la modificación (1) de JCTVC-M0316 es como sigue.

```
50     if (absCoeff > 25 * (1 << cRiceParam)) {
        } cRiceParam = 4;
        else
        if(absCoeff > 3 * (1 << cRiceParam))
    {
        } cRiceParam = min<UInt>(cRiceParam + 1,4);
```

Un ejemplo del estándar HEVC modificado para la modificación (1) es como sigue.

```
60     if (cLastAbsLevel > (25 * (1 << cLastRiceParam))) cRiceParam = 4
        else cRiceParam = Min(cLastRiceParam + (cLastAbsLevel >
            (3 * (1 << cLastRiceParam)) ? 1 : 0), 4)
```

Para la combinación de las modificaciones (1) y (2) de JCTVC-M0316, un ejemplo del código C modificado es como sigue.

```

if(absCoeff> 25*(1<<cRiceParam)) {
    } cRiceParam = 4;
else
if(absCoeff > 3*(1<<cRiceParam))
5   {
    cRiceParam = min<UInt>(cRiceParam + 1,4);
    }
if(cRiceParam == 4) {
10  } cRiceParam = 5;

```

Un ejemplo del estándar HEVC modificado para la combinación de las modificaciones (1) y (2) es el siguiente.

```

if (cLastAbsLevel > (25 * (1 « cLastRiceParam) )) cRiceParam = 4
else cRiceParam = Min(cLastRiceParam + (cLastAbsLevel >
15  (3 * (1 << cLastRiceParam) ) ? 1 : 0), 4)
if (cRiceParam == 4) cRiceParam = 5

```

Las técnicas de actualización rápida propuestas en JCTVC-M0316 muestran una buena eficiencia para algunos contenidos de la pantalla. Las técnicas de JCTVC-M0316 se ajustan mejor a algunas estadísticas de vídeo que el esquema de actualización del parámetro de Rice del HEVC, ya que el parámetro de Rice va directamente a su valor máximo cuando el valor absoluto del nivel de coeficiente para el coeficiente actual está por encima de un umbral. El procedimiento de adaptación rápida en JCTVC-M0316 básicamente establece el parámetro de Rice en un valor máximo cuando se observa un coeficiente grande. En el estándar HEVC, el parámetro de Rice (como se muestra en las ecuaciones anteriores) sólo aumenta en 1 o 0 en la codificación del valor absoluto restante para el nivel de coeficiente de cada coeficiente. Por lo tanto, en el esquema de actualización del parámetro de Rice del HEVC, el parámetro de Rice puede no adaptarse tan rápido a la codificación de grandes valores de coeficientes.

Un posible inconveniente del procedimiento propuesto en JCTVC-M0316 es que incorpora verificaciones condicionales en el procedimiento de actualización del parámetro de Rice. Tales verificaciones pueden hacer que el codificador de vídeo 20 y el decodificador de vídeo 30 sean más lentos. Además, las técnicas de JCTVC-M0316 requieren una multiplicación adicional para calcularse. Dichos cálculos adicionales tienen que llevarse a cabo dentro del bucle de codificación de entropía de coeficientes, que es un bucle crítico en la codificación de vídeo y puede ser un cuello de botella.

En vista de estos inconvenientes, esta divulgación propone técnicas para un esquema de actualización del parámetro de Rice rápido y eficiente en base a un incremento de la variable de acuerdo con los valores de nivel de coeficiente o contenido de vídeo. Las técnicas de esta divulgación no requieren la introducción de condiciones adicionales en el bucle crítico, y en algunos casos, pueden incluso eliminar algunas condiciones existentes en el esquema de actualización del parámetro de Rice del HEVC. Esta divulgación introduce una actualización rápida sin lógica adicional que puede ser más flexible porque el incremento de la variable permite que el valor del parámetro de Rice aumente en cualquier cantidad hasta alcanzar el valor máximo del parámetro de Rice.

En un primer ejemplo de esta divulgación, se describen técnicas para señalar parámetros para la actualización del parámetro de Rice. Las técnicas actuales en el estándar HEVC para el procedimiento de actualización del parámetro de Rice se repiten más abajo como:

```

if(absCoeff> 3*(1<<cRiceParam))
{
50  } cRiceParam = min<UInt>(cRiceParam + 1,4);

```

donde el cRiceParam es igual a 0 al comienzo de un CG.

Las técnicas de esta divulgación pueden expresarse con cuatro nuevos parámetros.

- INIT_PARAM: el valor inicial del parámetro de Rice al inicio del CG
- INC_PARAM: el valor de incremento del parámetro de Rice
- MAX_PARAM: el valor máximo del parámetro de Rice
- THRESH_ABS_COEFF: el valor de umbral usado en el esquema de actualización para comparar con el valor absoluto del nivel de coeficiente

De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 puede configurarse para señalar uno o más de estos parámetros en el flujo de bits, por ejemplo, en el conjunto de parámetro de secuencia (SPS), conjunto de parámetro de imagen (PPS), encabezado del segmento, LCU, CU, TU o nivel de bloque de transformación. En este caso, el codificador de vídeo 20 puede configurarse para elegir el mejor parámetro para el contenido que se codifica y señalar la selección al decodificador de vídeo 30.

Un ejemplo de la expresión en código C de la función de actualización del parámetro de Rice de esta divulgación, incluidos los parámetros señalados, es como sigue.

```

5      cRiceParam = INIT_PARAM
      if(absCoeff > THRESH_ABS_COEFF*(1<<cRiceParam))
      {
      } cRiceParam = min<UInt>(cRiceParam + INC_PARAM, MAX_PARAM);

```

10 En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden almacenar valores por defecto de los parámetros para el esquema de actualización del parámetro de Rice de esta divulgación. Para la codificación estándar HEVC de coeficientes de transformación cuantificados para contenido natural, los valores de los parámetros, que podrían ser los valores por defecto, pueden ser como sigue.

- INIT_PARAM = 0
- 15 • INC_PARAM = 1
- MAX_PARAM = 4
- THRESH_ABS_COEFF = 3

20 Para la codificación de los coeficientes, que no se han transformado o cuantificado, para el contenido de la pantalla o para la codificación sin pérdida, los parámetros por defecto pueden ser diferentes. Por ejemplo, los parámetros por defecto pueden ser como sigue.

- INIT_PARAM = 2
- INC_PARAM = 1
- 25 • MAX_PARAM = 5
- THRESH_ABS_COEFF = 3

Los valores de los parámetros anteriores son simplemente ejemplos. En otros ejemplos, pueden almacenarse otros valores por defecto en el codificador de vídeo 20 y el decodificador de vídeo 30. Por ejemplo, MAX_PARAM puede ser igual a un valor de 7. Como otro ejemplo, INC_PARAM puede ser igual a cualquier valor entero mayor que 1 hasta el valor máximo.

35 En otro ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para usar un conjunto de valores cuando el residual se codifica de manera con pérdida, y otro conjunto de valores cuando el residual se codifica de cualquier manera sin pérdida (es decir, cuando la bandera de derivación de cuantificación está activada). En otro ejemplo, se podría usar un conjunto de valores para la codificación con pérdida en el modo de salto de transformación o de derivación de transformación que es diferente de los valores del HEVC para la codificación con pérdida. En otro ejemplo, el conjunto de valores por defecto podría depender del tipo de transformación (por ejemplo, DCT, DST, salto o similar).

40 Como se describió anteriormente, el esquema de actualización del parámetro de Rice del HEVC aumenta el parámetro de Rice a lo máximo en 1 en cada actualización (es decir, en cada coeficiente que se codificará en modo de derivación mediante el uso de códigos de Golomb definidos por el parámetro de Rice). El procedimiento de adaptación rápida en JCTVC-M0316 básicamente establece el parámetro de Rice en un valor máximo cuando se observa un coeficiente grande.

45 En otro ejemplo, esta divulgación presenta una técnica de actualización rápida que puede utilizarse sin lógica adicional, o incluso reduciendo el número de verificaciones en el esquema de actualización del parámetro de Rice del HEVC. Esta divulgación propone el siguiente procedimiento para una actualización rápida y eficiente del valor del parámetro de Rice que es flexible porque puede aumentar el valor del parámetro de Rice en cualquier cantidad hasta alcanzar el valor máximo.

De acuerdo con este ejemplo de la divulgación, las líneas de código C para el esquema de actualización del parámetro de Rice del HEVC, se repiten más abajo:

```

55      if(absCoeff > 3*(1<<cRiceParam))
      {
      } cRiceParam = min<UInt>(cRiceParam + 1,4);

```

60 sería reemplazado por

```

      cRiceParam = min<UInt>((cRiceParam + (absCoeff >>
      (PARAM1 * (2<<cRiceParam))))), MAX_RICE_PARAM);

```

65

donde $(\text{absCoeff} \gg (\text{PARAM1} * (2 \ll \text{cRiceParam})))$ define un incremento de la variable del valor del parámetro de Rice que se determina en base al valor del parámetro de Rice (es decir, cRiceParam) y un valor absoluto de un nivel de coeficiente para el coeficiente que se codifica (es decir, absCoeff). En este ejemplo, el incremento de la variable se determina en base a una versión desplazada a la derecha del valor absoluto del nivel de coeficiente para el coeficiente que se codifica (es decir, absCoeff), donde una cantidad del desplazamiento a la derecha se basa en el valor del parámetro de Rice (es decir, cRiceParam).

En la representación anterior del esquema de actualización del parámetro de Rice de esta divulgación, se elimina una verificación del esquema de actualización del parámetro de Rice del HEVC y se eliminan dos o tres verificaciones con respecto al esquema de actualización del parámetro de Rice introducido en JCTVC-M0316. Esencialmente, cRiceParam se actualiza en base a una selección de un mínimo o del valor máximo del parámetro de Rice (es decir, MAX_RICE_PARAM) o un incremento de la variable del valor del parámetro de Rice (es decir, cRiceParam) que se determina en base al valor absoluto del coeficiente que se codifica (es decir, absCoeff) dividido (es decir, desplazado a la derecha, " \gg ") por una función del parámetro de Rice (es decir, cRiceParam) multiplicado (es decir, desplazado a la izquierda, " \ll ") por un valor constante del parámetro (PARAM1).

El estándar HEVC se definiría como sigue.

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + (\text{cLastAbsLevel} \gg (\text{PARAM1} * (2 \ll \text{cLastRiceParam}))), \text{MAX_RICE_PARAM})$$

En un ejemplo de la divulgación, PARAM1 se establece en 1 y MAX_RICE_PARAM se establece en 4. Como se describió anteriormente en el primer ejemplo de esta divulgación, estos parámetros pueden señalizarse por el codificador de vídeo 20 al decodificador de vídeo 30, o pueden derivarse por el decodificador de vídeo 30 sin señalización. Por ejemplo, MAX_RICE_PARAM puede establecerse igual a 4 para codificación con pérdida y establecerse igual a 5 para codificación sin pérdida (es decir, cuando la bandera de derivación del cu_transquant está activada).

En un ejemplo, mediante el uso de los valores de ejemplo dados anteriormente, la fórmula de código C para el esquema de actualización del parámetro de Rice de esta divulgación sería como sigue.

$$\text{cRiceParam} = \text{min}(\text{UInt}((\text{cRiceParam} + (\text{absCoeff} \gg (2 \ll \text{cRiceParam}))), 4);$$

De manera equivalente, el estándar HEVC se definiría como sigue.

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + (\text{cLastAbsLevel} \gg (2 \ll \text{cLastRiceParam})), 4)$$

Se debe señalar que la actualización del parámetro de Rice como se presenta en JCTVC-M0316 solo aumenta el valor del parámetro de Rice en 1 o establece el parámetro de Rice igual a su valor máximo de 4 o 5 en función de una condición. Las técnicas de esta divulgación pueden aumentar la cantidad de incremento de la variable en cualquier actualización por cualquier valor entero, es decir, 1, 2, 3 o más, hasta el valor máximo del parámetro de Rice. En algunos ejemplos, el valor máximo del parámetro de Rice puede ser un valor entero mayor que o igual a 4. En un ejemplo específico, el valor máximo del parámetro de Rice es igual a 7. La cantidad del incremento de la variable puede depender del valor absoluto del coeficiente que se codifica. En un ejemplo, en base al nivel de coeficiente para el coeficiente decodificado que tiene un valor absoluto grande, puede determinarse que el incremento de la variable tiene un valor entero mayor que uno. Como tal, las técnicas de esta divulgación permiten una mayor capacidad de adaptación.

Además, como se indicó anteriormente, las funciones que representan el procedimiento de actualización del HEVC y el procedimiento de actualización de JCTVC-M0316, el "?" es equivalente a una declaración condicional "Si". Las técnicas descritas en esta divulgación no requieren la realización de ningunas operaciones condicionales. Las operaciones condicionales pueden requerir mucho tiempo y procesamiento en el bucle crítico de la codificación de la entropía del coeficiente, y la eliminación de la declaración condicional "Si" puede mejorar la eficiencia.

Además, se debe señalar que esta capacidad de aumentar el parámetro de Rice en diferentes cantidades también se puede lograr por tablas de consulta. Por ejemplo, se puede usar una tabla de consulta en la que las entradas a la tabla de consulta son el valor del parámetro de Rice y el valor absoluto del nivel de coeficiente para el coeficiente que se decodifica, y la salida de la tabla de consulta es la cantidad del incremento de la variable que se aplica al valor del parámetro de Rice, que puede ser diferente del incremento convencional de 0 y 1.

Puede usarse una combinación de cualquiera de los ejemplos dados anteriormente. Por ejemplo, los parámetros en cualquiera de los ejemplos dados anteriormente pueden señalizarse por el codificador de vídeo 20 al decodificador

de vídeo 30, o derivados por el decodificador de vídeo 30 sin señalización, como se describió anteriormente. Las técnicas de esta divulgación pueden particularmente adecuarse para datos que están codificados sin pérdida o codificados con transformaciones diferentes a las del estándar HEVC, debido a las diferentes estadísticas de coeficientes observadas en esas situaciones. Como un ejemplo específico, la ecuación de actualización del parámetro de Rice $cRiceParam = \text{Min}(cLastRiceParam + (cLastAbsLevel \gg (2 \ll cLastRiceParam)), 4)$ puede usarse para CG para los que no se aplica una transformación, es decir, en el caso de codificación sin pérdida o codificación con pérdida con salto o derivación de transformación; de cualquier otra manera, se aplica el procedimiento de actualización del parámetro de Rice del HEVC. Otra situación en la que las técnicas de esta divulgación pueden proporcionar más ganancias es para la codificación de vídeo con una tasa de bits muy alta (es decir, alta calidad). Las técnicas de la divulgación, por lo tanto, pueden aplicarse en función de la tasa de bits objetivo, la proporción máxima de señal a ruido (PSNR) objetivo o la etapa de cuantificación o parámetro de cuantificación (QP) en el que se codifican los datos de vídeo residuales.

Las técnicas de esta divulgación incluyen variantes del esquema de actualización del parámetro de Rice descrito anteriormente, cuya fórmula de código C se repite más abajo.

$$cRiceParam = \text{min}(\text{UInt})((cRiceParam + (\text{absCoeff} \gg (PARAM1 * (2 \ll cRiceParam)))), \text{MAX_RICE_PARAM});$$

En un ejemplo de las técnicas de esta divulgación, la fórmula básica anterior puede ampliarse para obtener diferentes valores de umbral para el valor absoluto del coeficiente para el que se aumenta el parámetro de Rice. Los valores de los parámetros pueden sumarse al valor absoluto del coeficiente que se codifica (es decir, absCoeff), o sumar o multiplicar el valor por el cual se realiza la operación de desplazamiento a la derecha del valor absoluto. Un ejemplo de la fórmula del código C se da como:

$$cRiceParam = \text{min}(\text{UInt})((cRiceParam + ((\text{absCoeff} + P1) \gg (P2 + P3 * (P4 \ll cRiceParam)))), \text{MAX_RICE_PARAM});$$

donde los valores de los parámetros son $P1$, $P2$, $P3$ y $P4$, que son valores enteros, y donde $((\text{absCoeff} + P1) \gg (P2 + P3 * (P4 \ll cRiceParam)))$ define un incremento de la variable del valor del parámetro de Rice que se determina en base al valor del parámetro de Rice (es decir, $cRiceParam$) y un valor absoluto de un nivel de coeficiente para el coeficiente que se codifica (es decir, absCoeff). En este ejemplo, el incremento de la variable se determina en base a una versión desplazada a la derecha del valor absoluto del nivel de coeficiente para el coeficiente que se codifica (es decir, absCoeff), donde una cantidad del desplazamiento a la derecha se basa en el valor del parámetro de Rice (es decir, $cRiceParam$). El parámetro MAX_RICE_PARAM establece el valor máximo que puede alcanzar el parámetro de Rice.

Como se muestra en el ejemplo anterior, $P1$ puede sumarse al valor absoluto del nivel de coeficiente para el coeficiente que se codifica (es decir, absCoeff), $P2$ puede sumarse a un valor por el cual se realiza la operación de desplazamiento a la derecha del valor absoluto (es decir, $(P4 \ll cRiceParam)$), $P3$ puede multiplicar un valor por el que se realiza la operación de desplazamiento a la derecha del valor absoluto (es decir, $(P4 \ll cRiceParam)$). El valor del parámetro $P4$ y el valor del parámetro de Rice (es decir, $cRiceParam$) pueden determinar el valor por el cual se realiza la operación de desplazamiento a la derecha del valor absoluto.

De manera equivalente, el estándar HEVC puede definirse para este último caso como sigue.

$$cRiceParam = \text{Min}(cLastRiceParam + ((cLastAbsLevel + P1) \gg (P2 + P3 * (P4 \ll cLastRiceParam))), \text{MAX_RICE_PARAM})$$

En una implementación de esta fórmula, los valores de los parámetros se establecen en $P1 = 0$, $P2 = 1$, $P3 = 1$, $P4 = 1$ y $\text{MAX_RICE_PARAM} = 4$. El texto HEVC para este caso específico es como sigue.

$$cRiceParam = \text{Min}(cLastRiceParam + (cLastAbsLevel \gg (1 + (1 \ll cLastRiceParam))), 4)$$

En el ejemplo donde MAX_RICE_PARAM es igual a 5, los resultados de rendimiento pueden mejorarse para la codificación sin pérdida del contenido de la pantalla. Para un rendimiento aún mayor en el caso de codificación sin pérdida del contenido de la pantalla, MAX_RICE_PARAM puede establecerse en 7.

En otro ejemplo, al agregar una condición, el impacto en la cantidad del desplazamiento a la derecha aplicado al valor absoluto por el valor del parámetro de Rice puede limitarse como sigue.

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + ((\text{cLastAbsLevel} + \text{P1}) \gg (\text{P2} + \text{P3} * (\text{P4} \ll (\text{cLastRiceParam} > \text{P5} ? \text{P5} : \text{cLastRiceParam})))), \text{MAX_RICE_PARAM})$$

En el ejemplo anterior, el esquema de actualización del parámetro de Rice se modifica para incluir la condición ($\text{cLastRiceParam} > \text{P5} ? \text{P5} : \text{cLastRiceParam}$), en la que la cantidad de desplazamiento a la derecha por el parámetro de Rice se limita al valor de P5 cuando el valor del parámetro de Rice es mayor que P5. En una implementación, los valores de los parámetros se establecen en $\text{P1} = 0$, $\text{P2} = 1$, $\text{P3} = 1$, $\text{P4} = 1$, $\text{P5} = 0$ y $\text{MAX_RICE_PARAM} = 4$. El texto HEVC para este caso específico es como sigue.

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + (\text{cLastAbsLevel} \gg (2 \ll (\text{cLastRiceParam} > 0 ? 1 : 0))), 4)$$

En este caso, la cantidad de desplazamiento a la derecha aplicada al valor absoluto es la misma, es decir, igual a 1, para cualquier parámetro de Rice que sea mayor que o igual a 1 (es decir, mayor que 0).

Pueden usarse diferentes procedimientos para actualizar el valor del parámetro de Rice bajo diferentes condiciones. Los procedimientos de actualización del parámetro de Rice descritos en esta divulgación pueden aplicarse en función de si se aplica una transformación a un bloque de transformación que incluye el CG que se codifica. Un dispositivo de codificación de vídeo, como un codificador de vídeo 20 o un decodificador de vídeo 30, puede determinar si la transformación se aplica al bloque de transformación en base a uno o más de los siguientes: un estado de codificación sin pérdida, un estado de derivación de la transformación, un estado de salto de la transformación, o un perfil que se establece en un conjunto de parámetro, como el SPS. Un perfil que se establece y que usa el esquema de actualización del parámetro de Rice de esta divulgación puede ser un perfil para el rango de extensiones 4: 4: 4 del HEVC o un perfil dedicado al contenido de la pantalla, pero otros perfiles pueden usar el esquema de actualización del parámetro de Rice del HEVC.

En un ejemplo, la condición en base a si se aplica una transformación a un bloque de transformación que incluye el CG puede presentarse como sigue.

Si la bandera de salto de transformación es igual a 1 o la bandera de derivación de cu_transquant es igual a 1, la variable cRiceParam se deriva de cLastAbsLevel y cLastRiceParam como:

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + (\text{cLastAbsLevel} \gg (1 + (1 \ll \text{cLastRiceParam}))), 4)$$

De cualquier otra manera, la variable cRiceParam se deriva de cLastAbsLevel y cLastRiceParam como:

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + (\text{cLastAbsLevel} \gg (3 * (1 \ll \text{cLastRiceParam})) ? 1 : 0), 4)$$

En consecuencia, puede aplicarse una variante del esquema de actualización del parámetro de Rice de esta divulgación cuando no se aplica la transformación, es decir, en el caso de codificación sin pérdida o codificación con pérdida con salto de transformación, y puede aplicarse el esquema de actualización del parámetro de Rice del HEVC cuando se aplica la transformación, es decir, en el caso de codificación con pérdida de coeficientes de transformación cuantificados.

Las técnicas de esta divulgación incluyen variantes adicionales del esquema de actualización del parámetro de Rice descrito anteriormente, cuya fórmula de código C se repite más abajo.

$$\text{cRiceParam} = \text{min}(\text{UInt}((\text{cRiceParam} + (\text{absCoeff} \gg (\text{PARAM1} * (2 \ll \text{cRiceParam})))), \text{MAX_RICE_PARAM});$$

Las técnicas de esta divulgación describen además procedimientos para disminuir el valor del parámetro de Rice y para establecer o inicializar el parámetro de Rice al comienzo de cada CG.

En un ejemplo de las técnicas de esta divulgación, la fórmula básica anterior puede extenderse para obtener diferentes valores de umbral para el valor absoluto del coeficiente para el cual se incrementa el parámetro de Rice, y para eliminar una operación de multiplicación (es decir, una operación de desplazamiento de bits a la izquierda, " \ll "). Un ejemplo de la fórmula del código C se da como:

$$\text{cRiceParam} = \min((\text{cRiceParam} + ((\text{absCoeff} + \text{P1}) \gg (\text{P2} + \text{P3} * \text{cRiceParam}))), \text{MAX_RICE_PARAM});$$

5 donde los valores de los parámetros son P1, P2 y P3, que son valores enteros, y donde $((\text{absCoeff} + \text{P1}) \gg (\text{P2} + \text{P3} * \text{cRiceParam}))$ define un incremento de la variable del valor del parámetro de Rice que se determina en base al valor del parámetro de Rice (es decir, cRiceParam) y un valor absoluto de un nivel de coeficiente para el coeficiente que se codifica (es decir, absCoeff). En este ejemplo, el incremento de la variable se determina en base a una versión desplazada a la derecha del valor absoluto del nivel de coeficiente para el coeficiente que se codifica (es decir, absCoeff), donde una cantidad del desplazamiento a la derecha se basa en el valor del parámetro de Rice (es decir, cRiceParam). El parámetro MAX_RICE_PARAM establece el valor máximo que puede alcanzar el parámetro de Rice.

15 Como se muestra en el ejemplo anterior, P1 puede sumarse al valor absoluto del nivel de coeficiente para el coeficiente que se codifica (es decir, absCoeff), P2 puede sumarse al valor del parámetro de Rice (es decir, cRiceParam) por el que se realiza la operación de desplazamiento a la derecha del valor absoluto, P3 puede multiplicar el valor del parámetro de Rice (es decir, cRiceParam) por el que se realiza la operación de desplazamiento a la derecha del valor absoluto.

20 El estándar HEVC se definiría como sigue.

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + ((\text{cLastAbsLevel} + \text{P1}) \gg (\text{P2} + \text{P3} * \text{cLastRiceParam})), \text{MAX_RICE_PARAM})$$

25 En una implementación del procedimiento de actualización del ejemplo anterior, los valores de los parámetros se establecen en P1 = 0, P2 = 2, P3 = 1 y MAX_RICE_PARAM = 4. Estos parámetros pueden señalizarse por el codificador de vídeo 20 al decodificador de vídeo 30, o los parámetros pueden derivarse por el decodificador de vídeo 30 sin señalización. En algunos ejemplos, MAX_RICE_PARAM puede establecerse igual a 4 para codificación con pérdida y establecerse igual a 5 para codificación sin pérdida. En algunos otros ejemplos, MAX_RICE_PARAM puede establecerse igual a 7.

35 Un ejemplo de la fórmula del código C para el esquema de actualización del parámetro de Rice de esta divulgación mediante el uso de los valores por defecto anteriores es como sigue.

$$\text{cRiceParam} = \min((\text{cRiceParam} + (\text{absCoeff} \gg (2 + \text{cRiceParam}))), 4);$$

40 De manera equivalente, el estándar HEVC se definiría para este último caso como sigue.

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + (\text{cLastAbsLevel} \gg (2 + \text{cLastRiceParam})), 4)$$

45 En otro ejemplo, puede limitarse la cantidad en una cantidad del incremento de la variable en cada actualización del valor del parámetro de Rice. Por ejemplo, puede imponerse una restricción adicional en la que el valor del parámetro de Rice puede aumentar a lo máximo un valor de incremento máximo (es decir, MAX_INCREMENT) en cada actualización (es decir, para cada nivel de coeficiente que se codifica mediante el uso de códigos de Golomb definidos por el parámetro de Rice). Una fórmula de ejemplo para limitar el esquema de actualización del parámetro de Rice puede ser como sigue.

$$\begin{aligned} \text{increment} &= \text{Min}((\text{cLastAbsLevel} \gg (2 + \text{cLastRiceParam})), \text{MAX_INCREMENT}) \\ \text{cRiceParam} &= \text{Min}(\text{cLastRiceParam} + \text{increment}, \text{MAX_RICE_PARAM}) \end{aligned}$$

55 En otros ejemplos, en lugar de limitar la cantidad del incremento de la variable en cada actualización del valor del parámetro de Rice a MAX_INCREMENT, se introduce un factor de amortiguación por el cual se escala el incremento de la variable. Un ejemplo del esquema de actualización del parámetro de Rice que se amortigua puede ser como sigue.

$$\text{increment} = \lfloor \text{damp_fact} * (\text{cLastAbsLevel} \gg (2 + \text{cLastRiceParam})) \rfloor$$

$$\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + \text{increment}, \text{MAX_RICE_PARAM})$$

En algunos ejemplos, el factor de amortiguación (es decir, `damp_fact`) es menor o igual a 1.

Este concepto de limitar el incremento máximo de `cRiceParam` en una etapa de actualización o amortiguación de la cantidad de incremento puede aplicarse a cualquiera de los ejemplos del esquema de actualización del parámetro de Rice descrito en esta divulgación. El mismo concepto también puede aplicarse para limitar o amortiguar una disminución del valor del parámetro de Rice en los casos en que se permite que el parámetro de Rice disminuya, como se describe con más detalle más abajo. Las técnicas descritas anteriormente pueden aplicarse a CG en todos los bloques o solo CG en bloques para los que no se aplica una transformación, es decir, para codificación sin pérdida o codificación con pérdida con salto o derivación de transformación.

Es posible que las técnicas de actualización del parámetro de Rice descritas anteriormente no permitan que el parámetro de Rice disminuya con respecto a un CG. En cambio, la función de actualización se ha descrito como una función creciente. Las técnicas de esta divulgación incluyen además un procedimiento de actualización del parámetro de Rice que permite que el valor del parámetro de Rice disminuya en algunos casos. En términos generales, el esquema de actualización del parámetro de Rice que permite tanto incrementar como decrementar el parámetro de Rice puede expresarse como:

$$cRiceParam = F_1(cRiceParam, absCoeff)$$

donde F_1 es una función que bajo algunos valores de entrada del parámetro de Rice (es decir, `cRiceParam`) y el valor absoluto del nivel de coeficiente para el coeficiente que se codifica (es decir, `absCoeff`) puede disminuirse de manera que el valor de salida de `cRiceParam` es menor que el valor de entrada del `cRiceParam`.

En un ejemplo más específico, el esquema de actualización puede describirse como:

```
If ( F2 ( cRiceParam, absCoeff ) )
    cRiceParam = cRiceParam + F3( cRiceParam, absCoeff );
else
    cRiceParam = cRiceParam - F4( cRiceParam, absCoeff );
```

donde se prueba una condición en la función F_2 , y funciones F_3 y F_4 toman como entradas `cRiceParam` y `absCoeff`, y potencialmente otras entradas como estadísticas de CG anteriores) y valores de salida del parámetro de Rice que son mayores que o iguales a 0.

En una implementación del esquema de actualización del ejemplo anterior, las funciones simples F_2 , F_3 y F_4 se definen cada una como sigue:

```
If (absCoeff >= (P1*cRiceParam+P2))
    cRiceParam = min((cRiceParam+(absCoeff >> (2+cRiceParam))),
        MAX_RICE_PARAM);
else
    cRiceParam = max(cRiceParam - 1, 0);
```

donde P_1 y P_2 son valores de parámetro. En un ejemplo, los valores de parámetro pueden establecerse como $P_1 = 1$, $P_2 = 0$ y $MAX_RICE_PARAM = 4$ para proporcionar buenas ganancias de rendimiento de codificación. En otros ejemplos, particularmente para la codificación sin pérdida, el valor máximo del parámetro de Rice puede establecerse igual a un valor entero que es mayor que 4, como 5 o 7.

En el ejemplo anterior, puede realizarse una operación condicional para determinar si el valor absoluto del nivel de coeficiente para el coeficiente actual que se codifica (es decir, `absCoeff`) es mayor que o igual al valor del parámetro de Rice (es decir, `cRiceParam`). Si el valor absoluto del nivel de coeficiente para el coeficiente actual es mayor que o igual al valor del parámetro de Rice, entonces el valor del parámetro de Rice puede actualizarse en base a la selección del mínimo o del valor máximo del parámetro de rice o el incremento de la variable del valor del parámetro de Rice, como se describió en detalle anteriormente. Si el valor absoluto del nivel de coeficiente para el coeficiente actual es menor que el valor del parámetro de Rice, entonces el valor del parámetro de Rice puede actualizarse en base a una selección de un máximo de 0 o cero o una disminución del valor del parámetro de Rice.

El ejemplo anterior del esquema de actualización del parámetro de Rice que permite tanto incrementar como disminuir el parámetro de Rice puede incluir una operación condicional adicional (es decir, una condición `if/else`) para decidir si aumentar o disminuir el parámetro de Rice. Sin embargo, la técnica es en base al esquema de actualización del parámetro de Rice de esta divulgación en el que una condición se elimina con respecto al esquema de actualización del parámetro de Rice del HEVC de manera que el número total de condiciones permanece igual que en el esquema del HEVC. En algunos ejemplos, otros procedimientos de actualización del parámetro de Rice pueden modificarse para incrementar o disminuir el parámetro de Rice.

Adicionalmente, en el caso de codificación con pérdida con salto de transformación, la disminución del valor del parámetro de Rice solo puede aplicarse a los bloques de salto de transformación como sigue:

```

5      If (absCoeff >= cRiceParam || transform_skip_flag)
        cRiceParam = min((cRiceParam+(absCoeff>>(2 + cRiceParam))), 4);
      else
        cRiceParam = cRiceParam - 1;

```

10 donde transform_skip_flag señala si la transformación se ha saltado para el CG o el bloque de transformación del CG.

En los ejemplos anteriores, el parámetro de Rice puede disminuir solo por 1. En otros ejemplos, la función F4 puede elegirse de modo que el parámetro de Rice pueda disminuir por más de 1 si el valor actual del parámetro de Rice es grande y el valor absoluto del coeficiente actual es pequeño.

15 Las técnicas de esta divulgación se han centrado hasta ahora en la actualización de un valor del parámetro de Rice mientras se codifica un CG o un bloque de coeficientes. El siguiente ejemplo describe técnicas para inicializar un valor del parámetro de Rice al comienzo de un CG. Convencionalmente, en el estándar HEVC, el parámetro de Rice se restablece a un valor de 0 después de codificar los valores absolutos restantes de los coeficientes en un CG (por ejemplo, un bloque de transformación 4 × 4 o un subbloque de 4 × 4 en un bloque de transformación). En el caso de la codificación de niveles de coeficiente para el contenido de la pantalla o en el caso de la codificación sin pérdida o la codificación con pérdida con salto o derivación de transformación, la inicialización del valor del parámetro de Rice a cero puede no ser óptimo. Las técnicas de esta divulgación pueden inicializar o establecer el valor del parámetro de Rice en un valor distinto de cero al comienzo de cada CG de un bloque de transformación.

25 Como un ejemplo del esquema de inicialización del parámetro de Rice de esta divulgación, el parámetro de Rice no se restablece después de codificar un CG. De acuerdo con las técnicas de esta divulgación, el valor del parámetro de Rice para un CG posterior se establece en cambio en el mismo valor obtenido para el parámetro de Rice al final de la codificación del CG actual. Como en el esquema de inicialización del HEVC, el valor puede establecerse igual a 0 al comienzo del CG posterior. Sin embargo, a diferencia del esquema del HEVC, no es necesario establecer el valor del parámetro de Rice en 0.

30 Como otro ejemplo del esquema de inicialización del parámetro de Rice de esta divulgación, el valor del parámetro de Rice para un CG posterior se establece en un valor que se basa en el valor del parámetro de Rice después de codificar el CG actual. En un ejemplo específico, el parámetro de Rice al comienzo de cada CG se establece como sigue.

$$\text{cRiceParam} = \max (0, \text{cRiceParam} - 1);$$

40 En el ejemplo anterior, el valor del parámetro de Rice se inicializa para el CG posterior en base a una selección de un máximo de cero o una disminución en el valor del parámetro de Rice después de codificar el CG actual. En algunos otros ejemplos, el valor de inicialización del parámetro de Rice puede tener un límite, como en el siguiente ejemplo.

$$\text{cRiceParam} = \min (2, \max (0, \text{cRiceParam} - 1));$$

50 En este ejemplo, el valor de inicialización del parámetro de Rice está limitado a no ser mayor que 2.

En algunos ejemplos, pueden usarse valores distintos de 1 para reducir el valor actual del parámetro de Rice (por ejemplo, el valor n), que incluye restar 2 (es decir, n es igual a 2) en lugar que el 1 del valor actual del parámetro de Rice. Por ejemplo, el valor de la disminución para inicializar el parámetro de Rice puede ser variable en base a si el CG posterior se incluye en un bloque de transformación para el que se aplica una transformación. Un ejemplo de fórmula puede ser como sigue.

```

55      if(transform_skip_flag)
        cRiceParam = max(0, cRiceParam - 1);
      else
60      cRiceParam = max(0, cRiceParam - 2);

```

Los casos de ejemplo del esquema de inicialización del parámetro de Rice descrito en esta divulgación pueden aplicarse a CG incluidos en todos los bloques de transformación o pueden aplicarse a CG en base a si se aplica una transformación al bloque de transformación. Por ejemplo, en el caso de un bloque de transformación con salto de transformación o derivación de transformación, el valor del parámetro de Rice puede no restablecerse a 0 para un

CG posterior en el bloque, pero para bloques de transformación para los que se ha aplicado la transformación, el parámetro de Rice puede restablecerse a 0 para un CG posterior en el bloque.

Por ejemplo, en el caso de codificación con pérdida con salto de transformación, la disminución en el valor del parámetro de Rice para la inicialización para un CG posterior puede aplicarse sólo para CG en bloques de salto de transformación. Como ejemplo, la fórmula puede ser como sigue.

```

if(transform_skip_flag)
    cRiceParam = max(0, cRiceParam - 1);
else
    cRiceParam = 0

```

En general, esta divulgación describe técnicas para ajustar el valor del parámetro de Rice usado para definir los códigos de Golomb para la codificación de nivel de coeficiente en base a si se aplica una transformación a un bloque de transformación que incluye el CG. De acuerdo con esta divulgación, ajustar el valor del parámetro de Rice puede incluir actualizar el valor del parámetro de Rice para definir los códigos de Golomb para codificar un valor absoluto restante de un nivel de coeficiente para un coeficiente posterior en el CG, o inicializar el valor del parámetro de Rice para definir los códigos de Golomb para codificar un valor absoluto restante de un nivel de coeficiente para al menos un coeficiente en un CG posterior.

Por ejemplo, si no se aplica una transformación a un bloque de transformación, es decir, en el caso de salto o derivación de transformación, el esquema de actualización del parámetro de Rice descrito en esta divulgación puede usarse para actualizar el valor del parámetro de Rice para un coeficiente posterior en el CG. De cualquier otra manera, si la transformación se aplica al bloque de transformación, el esquema de actualización del parámetro de Rice del HEVC puede usarse para actualizar el valor del parámetro de Rice para un coeficiente posterior en el CG. En un ejemplo adicional, si la transformación no se aplica al bloque de transformación, el valor del parámetro de Rice puede inicializarse para un CG posterior en el bloque de transformación en base al valor del parámetro de Rice después de codificar el CG actual. De cualquier otra manera, si la transformación se aplica al bloque de transformación, el valor del parámetro de Rice puede inicializarse para que sea igual a cero para un CG posterior en el bloque de transformación.

Aunque los procedimientos de ejemplo para ajustar un valor del parámetro de Rice, por ejemplo, los esquemas de inicialización y actualización del parámetro de Rice se describen por separado anteriormente, las técnicas de esta divulgación no están limitadas. En general, pueden ser posibles varias combinaciones de las técnicas de ejemplo descritas anteriormente. Los procedimientos de ejemplo descritos anteriormente también pueden implementarse por separado. Además, todos los procedimientos descritos anteriormente o sus combinaciones pueden aplicarse a CG dentro de todos los bloques de transformación o solo los bloques de transformación para los que se salte o se derive una transformación.

La Figura 5 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas para la codificación de niveles de coeficiente descritos en esta divulgación. El codificador de vídeo 20 se describirá en el contexto de la codificación del HEVC con propósitos ilustrativos, pero sin limitación de esta divulgación en cuanto a otros estándares o procedimientos de codificación que pueden requerir el escaneo de coeficientes de transformación. El codificador de vídeo 20 puede realizar intracodificación e intercodificación de CU dentro de los fotogramas de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en los datos de vídeo dentro de un fotograma de vídeo dado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal entre un fotograma actual y los fotogramas previamente codificados de una secuencia de vídeo. El modo Intra (modo I) puede referirse a cualquiera de varios modos de compresión de vídeo basados en el espacio. EL modo Inter, como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de varios modos de compresión de vídeo basados en el tiempo.

Como se muestra en la Figura 5, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de un fotograma de vídeo a codificarse. En el ejemplo de la Figura 5, el codificador de vídeo 20 incluye una unidad de selección de modo 30, una memoria de datos de vídeo 41, una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de procesamiento de intrapredicción 46, una memoria intermedia de imagen decodificada (DPB) 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantificación 54 y una unidad de codificación de entropía 56. La unidad de procesamiento de transformación 52 ilustrada en la Figura. 5 es la unidad que aplica la transformación actual o combinaciones de transformación a un bloque de datos residuales, y no debe confundirse con un bloque de coeficientes de transformación, que también puede referirse como unidad de transformación (TU) de una CU. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye una unidad de cuantificación inversa 58, una unidad de procesamiento de transformación inversa 60, y un sumador 62. También puede incluirse un filtro de desbloqueo (no mostrado en la Figura 5) para filtrar los límites de los bloques para eliminar los artefactos de bloqueo del vídeo reconstruido. Si lo desea, el filtro de desbloqueo filtraría típicamente la salida del sumador 62.

La memoria de datos de vídeo 41 puede almacenar datos de vídeo para codificarse por los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 41 pueden obtenerse, por ejemplo, de la fuente de vídeo 18. La memoria intermedia de imagen decodificada 64 puede ser una memoria de imagen de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de vídeo 41 y la memoria intermedia de imagen decodificada 64 se pueden formar por cualquiera de una variedad de dispositivos de memoria, como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM), u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 41 y la memoria intermedia de imagen decodificada 64 pueden proporcionarse por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria de datos de vídeo 41 puede estar en el chip con otros componentes del codificador de vídeo 20, o fuera del chip con relación a esos componentes.

Durante el procedimiento de codificación, el codificador de vídeo 20 recibe un fotograma o segmento de vídeo a codificar. El fotograma o segmento puede dividirse en múltiples bloques de vídeo, por ejemplo, unidades de codificación más grande (LCU). La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan una codificación interpredictiva del bloque de vídeo que se recibe con relación a uno o más bloques en uno o más fotogramas de referencia para proporcionar compresión temporal. La unidad de procesamiento de intrapredicción 46 puede realizar una codificación intrapredictiva del bloque de vídeo que se recibe con relación a uno o más bloques vecinos en el mismo fotograma o segmento como el bloque a codificar para proporcionar compresión espacial.

La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, en base a los resultados de error (es decir, distorsión) para cada modo, y proporciona el bloque de intrapredicción o interpredicción resultante (por ejemplo, una unidad de predicción (PU)) al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso en una imagen de referencia. El sumador 62 combina el bloque previsto con datos cuantificados inversos, transformados inversos de la unidad de procesamiento de transformación inversa 60 para el bloque que reconstruya el bloque codificado, como se describe con mayor detalle más abajo. Algunos fotogramas de vídeo pueden designarse como fotogramas I, donde todos los bloques de un fotograma I se codifican en un modo de intrapredicción. En algunos casos, la unidad de procesamiento de intrapredicción 46 puede realizar la codificación de intrapredicción de un bloque en un fotograma P o B, por ejemplo, cuando la búsqueda de movimiento realizada por la unidad de estimación de movimiento 42 no resulta en una predicción suficiente del bloque.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse altamente, pero se ilustran por separado por propósitos conceptuales. La estimación de movimiento (o búsqueda de movimiento) es el procedimiento de generar vectores de movimiento, que estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una unidad de predicción en un fotograma actual con relación a una muestra de referencia de un fotograma de referencia. La unidad de estimación de movimiento 42 calcula un vector de movimiento para una unidad de predicción de un fotograma intercodificado al comparar la unidad de predicción con muestras de referencia de una imagen de referencia almacenada en la memoria intermedia de imagen decodificada 64. Una muestra de referencia puede ser un bloque que se encuentra que coincide estrechamente con la porción de la CU, incluida la PU que se codifica en términos de diferencia de píxel, que puede determinarse por la suma de la diferencia absoluta (SAD), la suma de la diferencia al cuadrado (SSD), u otras métricas de diferencia. La muestra de referencia puede aparecer en cualquier lugar dentro de un fotograma de referencia o segmento de referencia, y no necesariamente en un límite de bloque (por ejemplo, unidad de codificación) del fotograma o segmento de referencia. En algunos ejemplos, la muestra de referencia puede ocurrir en una posición de píxel fraccional.

La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación de entropía 56 y la unidad de compensación de movimiento 44. La porción del fotograma de referencia identificada por un vector de movimiento puede referirse como muestra de referencia. La unidad de compensación de movimiento 44 puede calcular un valor de predicción para una unidad de predicción de una CU actual, por ejemplo, al recuperar la muestra de referencia identificada por un vector de movimiento para la PU.

La unidad de procesamiento de intrapredicción 46 puede intraprededir el bloque que se recibe, como una alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44. La unidad de procesamiento de intrapredicción 46 puede predecir el bloque que se recibe con relación a los bloques vecinos previamente codificados, por ejemplo, los bloques de arriba, de arriba y a la derecha, de arriba y a la izquierda, o a la izquierda del bloque actual, al asumir una orden de codificación de izquierda a derecha, de arriba a abajo para los bloques. La unidad de procesamiento de intrapredicción 46 puede configurarse con una variedad de diferentes modos de intrapredicción. Por ejemplo, la unidad de procesamiento de intrapredicción 46 puede configurarse con un cierto número de modos de predicción direccional, por ejemplo, treinta y tres modos de predicción direccional, en base al tamaño de la CU que se codifica.

La unidad de procesamiento de intrapredicción 46 puede seleccionar un modo de intrapredicción, por ejemplo, calculando valores de error para varios modos de intrapredicción y seleccionando un modo que produzca el valor de error más bajo. Los modos de predicción direccional pueden incluir funciones para combinar valores de píxeles espacialmente vecinos y aplicar los valores combinados a una o más posiciones de píxeles en una PU. Una vez que se han calculado los valores para todas las posiciones de los píxeles en la PU, la unidad de procesamiento de intrapredicción 46 puede calcular un valor de error para el modo de predicción en base a las diferencias de píxeles entre la PU y el bloque que se recibe a codificarse. La unidad de procesamiento de intrapredicción 46 puede continuar probando modos de intrapredicción hasta que se descubra un modo de intrapredicción que produzca un valor de error aceptable. La unidad de procesamiento de intrapredicción 46 puede entonces enviar la PU al sumador 50.

El codificador de vídeo 20 forma un bloque residual al restar los datos de predicción calculados por la unidad de compensación de movimiento 44 o la unidad de procesamiento de intrapredicción 46 del bloque de vídeo original que se codifica. El sumador 50 representa el componente o componentes que realizan esta operación de resta. El bloque residual puede corresponder a una matriz bidimensional de valores de diferencia de píxeles, donde el número de valores en el bloque residual es el mismo como el número de píxeles en la PU correspondiente al bloque residual. Los valores del bloque residual pueden corresponder a las diferencias, es decir, al error, entre los valores de los píxeles cúbicos en la PU y en el bloque original a codificar. Las diferencias pueden ser diferencias de croma o luma en función del tipo de bloque que se codifica.

La unidad de procesamiento de transformación 52 puede formar una o más unidades de transformación (TU) a partir del bloque residual. La unidad de procesamiento de transformación 52 selecciona una transformación de entre una pluralidad de transformaciones. La transformación puede seleccionarse en base a una o más características de codificación, como el tamaño de bloque, el modo de codificación o similares. La unidad de procesamiento de transformación 52 aplica entonces la transformación seleccionada a la TU, lo que produce un bloque de vídeo que comprende una matriz bidimensional de coeficientes de transformación. La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 puede entonces cuantificar los coeficientes de transformación.

La unidad de codificación de entropía 56 puede entonces realizar un escaneo de los coeficientes en la matriz de acuerdo con un modo de escaneo. En el caso de codificación con pérdida, los coeficientes pueden ser coeficientes de transformación cuantificados. En el caso de codificación sin pérdida o codificación con pérdida con salto o derivación de transformación, los coeficientes pueden ser coeficientes que no se han transformado o cuantificado. Esta divulgación describe que la unidad de codificación de entropía 56 realiza el escaneo. Sin embargo, debe entenderse que, en otros ejemplos, otras unidades de procesamiento, como la unidad de cuantificación 54, podrían realizar el escaneo.

Una vez que los coeficientes de transformación se escanean en la matriz unidimensional, la unidad de codificación de entropía 56 puede aplicar codificación de entropía como CABAC, codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación de entropía de particiones de intervalo de probabilidad (PIPE) u otra metodología de codificación de entropía a los coeficientes. La unidad de codificación de entropía 56 puede configurarse para codificar los coeficientes de acuerdo con las técnicas de esta divulgación. En el ejemplo de CABAC, la unidad de codificación de entropía 56 puede codificar los coeficientes mediante el uso de o modo de codificación regular o modo de derivación. Para realizar CABAC, la unidad de codificación de entropía 56 puede seleccionar un modelo de contexto para aplicar un determinado contexto para codificar los símbolos a transmitir. El contexto puede relacionarse con, por ejemplo, si los valores vecinos son distintos de cero o no. La unidad de codificación de entropía 56 puede seleccionar el modelo de contexto usado para codificar estos símbolos en base a, por ejemplo, una dirección de intrapredicción para los modos de intrapredicción, una posición de escaneo del coeficiente correspondiente a los elementos de sintaxis, tipo de bloque y o tipo de transformación, entre otros factores usados para la selección del modelo de contexto.

La unidad de codificación de entropía 56 codifica niveles de coeficiente de datos de vídeo residuales en un flujo de bits para su transmisión a un decodificador de vídeo o a un dispositivo de almacenamiento. En el caso de codificación de vídeo sin pérdida o codificación de vídeo con pérdida con salto o derivación de transformación, los coeficientes a codificarse pueden tener niveles de coeficiente con valores absolutos grandes. Cuando los coeficientes representan el contenido de la pantalla, que puede incluir gráficos y regiones de texto, el contenido puede no ser bien predicho, lo que resulta en valores absolutos grandes de niveles de coeficiente para los coeficientes a codificarse.

La unidad de codificación de entropía 56 codifica un valor absoluto restante (por ejemplo, coeficiente abs_level remaining o levelRem) de un nivel de coeficiente para al menos un coeficiente en un grupo de coeficientes (CG) en modo de derivación mediante el uso de códigos de Golomb definidos por un parámetro de Rice. De acuerdo con las técnicas de esta divulgación, la unidad de codificación de entropía 56 se configura para actualizar un valor del parámetro de Rice en base a una selección de un mínimo o de un valor máximo del parámetro de Rice o un incremento de la variable del valor del parámetro de Rice. La cantidad del incremento de la variable se determina en base al valor del parámetro de Rice y un valor absoluto del nivel de coeficiente para el coeficiente actual codificado.

El incremento de la variable permite aumentar el valor del parámetro de Rice en cualquier valor entero hasta el valor máximo del parámetro de Rice. Los ejemplos de ecuaciones representan la actualización del parámetro de Rice que se describen con más detalle anteriormente. El esquema de actualización del parámetro de Rice descrito en esta divulgación permite que el parámetro de Rice se adapte rápidamente y eficientemente a grandes valores de coeficientes, que pueden ocurrir en bloques de contenido de la pantalla y bloques con salto o derivación de transformación.

En algunos ejemplos, la unidad de codificación de entropía 56 puede configurarse además para ajustar el valor del parámetro de Rice usado para definir códigos de Golomb para la codificación de nivel de coeficiente en base a si se aplica una transformación a un bloque de transformación que incluye el CG. De acuerdo con las técnicas descritas en esta divulgación, cuando la transformación no se aplica al bloque de transformación, es decir, en el caso de salto o derivación de la transformación, la unidad de codificación de entropía 56 puede realizar la actualización del parámetro de Rice y los esquemas de inicialización del parámetro de Rice que son flexibles para adaptarse a valores absolutos grandes de niveles de coeficiente para los coeficientes del CG. Por otro lado, cuando la transformación se aplica al bloque de transformación, la unidad de codificación de entropía 56 puede aplicar los esquemas de actualización e inicialización del parámetro de Rice del HEVC, que se optimizan para los pequeños valores de coeficientes de los coeficientes transformados.

Después de la codificación de entropía por la unidad de codificación de entropía 56, el vídeo codificado resultante puede transmitirse a otro dispositivo, como el decodificador de vídeo 30, o archivar para su posterior transmisión o recuperación. En algunos casos, la unidad de codificación de entropía 56 u otra unidad del codificador de vídeo 20 puede configurarse para realizar otras funciones de codificación, además de la codificación de entropía. Por ejemplo, la unidad de codificación de entropía 56 puede configurarse para determinar valores de patrón de bloque codificado (CBP) para CU y PU. Además, en algunos casos, la unidad de codificación de entropía 56 puede realizar la codificación de longitud de ejecución de coeficientes.

La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia de la memoria intermedia de imagen decodificada 64. 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 sub-enteros para su uso en la estimación de movimiento. 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 vídeo reconstruido para su almacenamiento en la memoria intermedia de imagen decodificada 64. El bloque de vídeo reconstruido puede usarse por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en un fotograma de vídeo posterior.

La Figura 6 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede implementar las técnicas para la decodificación de niveles de coeficiente descritos en esta divulgación. En el ejemplo de la Figura 6, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 70, una memoria de datos de vídeo 71, una unidad de compensación de movimiento 72, una unidad de procesamiento de intrapredicción 74, una unidad de cuantificación inversa 76, una unidad de transformación inversa 78, una memoria intermedia de imagen decodificada (DPB) 82 y un sumador 80. El decodificador de vídeo 30 puede, en algunos ejemplos, realizar un pase de decodificación generalmente recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 (ver Figura 5).

La memoria de datos de vídeo 71 puede almacenar datos de vídeo, como un flujo de bits de vídeo codificado, para decodificarse por los componentes del decodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 71 pueden obtenerse, por ejemplo, de un medio legible por ordenador 16, por ejemplo, de una fuente de vídeo local, como una cámara, a través de la comunicación de datos de vídeo por red inalámbrica o alámbrica, o por acceder a medios de almacenamiento de datos físicos. La memoria de datos de vídeo 71 puede formar una memoria intermedia de imagen codificada (CPB) que almacena datos de vídeo codificado de un flujo de bits de vídeo codificado. La memoria intermedia de imagen decodificada 82 puede ser una memoria de imagen de referencia que almacena datos de vídeo de referencia para su uso en la decodificación de datos de vídeo por el decodificador de vídeo 30, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de vídeo 71 y la memoria intermedia de imagen decodificada 82 se pueden formar por cualquiera de una variedad de dispositivos de memoria, como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM), u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 71 y la memoria intermedia de imagen decodificada 82 pueden proporcionarse por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria de datos de vídeo 71 puede estar en el chip con otros componentes del decodificador de vídeo 30, o fuera del chip con relación a esos componentes.

La unidad de decodificación de entropía 70 realiza un proceso de decodificación de entropía en el flujo de bits codificado para recuperar una matriz unidimensional de coeficientes de datos de vídeo residuales. El proceso de decodificación de entropía usado depende de la codificación de entropía usada por el codificador de vídeo 20 (por ejemplo, CABAC). El proceso de codificación de entropía usado por el codificador puede señalizarse en el flujo de bits codificado o puede ser un procedimiento predeterminado. La unidad de decodificación de entropía 70 puede configurarse para decodificar los coeficientes de acuerdo con las técnicas de esta divulgación. En el ejemplo de CABAC, la unidad de decodificación de entropía 70 puede decodificar los coeficientes mediante el uso de o modo de codificación regular o modo de derivación.

En algunos ejemplos, la unidad de decodificación de entropía 70 puede escanear los valores que se reciben mediante el uso de un escaneo que refleja el modo de escaneo usado por la unidad de codificación de entropía 56 del codificador de vídeo 20. Aunque el escaneo de coeficientes puede realizarse en la unidad de cuantificación inversa 76, el escaneo se describirá con propósitos ilustrativos como se realiza por la unidad de decodificación de entropía 70. Además, aunque se muestran como unidades funcionales separadas para facilitar la ilustración, la estructura y funcionalidad de la unidad de decodificación de entropía 70, la unidad de cuantificación inversa 76 y otras unidades del decodificador de vídeo 30 pueden integrarse altamente entre sí.

La unidad de decodificación de entropía 70 decodifica los niveles de coeficiente de los coeficientes de datos de vídeo residuales del flujo de bits de una manera recíproca al codificador de vídeo 20. En el caso de la codificación de vídeo con pérdida, los coeficientes a decodificarse pueden ser coeficientes de transformación cuantificados. En el caso de codificación de vídeo sin pérdida o codificación de vídeo con pérdida con salto o derivación de transformación, los coeficientes a decodificarse pueden tener niveles de coeficiente con valores absolutos grandes. Cuando los coeficientes representan contenido de la pantalla, que puede incluir gráficos y regiones de texto, el contenido puede no ser bien predicho, lo que resulta en valores absolutos grandes de niveles de coeficiente para los coeficientes a decodificarse.

La unidad de decodificación de entropía 70 decodifica un valor absoluto restante de un nivel de coeficiente para al menos un coeficiente en un grupo de coeficientes (CG) mediante el uso de códigos de Golomb definidos por un parámetro de Rice. De acuerdo con las técnicas de esta divulgación, la unidad de decodificación de entropía 70 se configura para actualizar un valor del parámetro de Rice en base a una selección de un mínimo o de un valor máximo del parámetro de Rice o un incremento de la variable del valor del parámetro de Rice. La cantidad del incremento de la variable se determina en base al valor del parámetro de Rice y un valor absoluto del nivel de coeficiente para el coeficiente actual que se decodifica. El incremento de la variable permite aumentar el valor del parámetro de Rice en cualquier valor entero hasta el valor máximo del parámetro de Rice. Los ejemplos de ecuaciones representan la actualización del parámetro de Rice que se describen con más detalle anteriormente. El esquema de actualización del parámetro de Rice descrito en esta divulgación permite que el parámetro de Rice se adapte rápidamente y eficientemente a grandes valores de coeficientes, que pueden ocurrir en bloques de contenido de la pantalla y bloques con salto o derivación de transformación.

En algunos ejemplos, la unidad de decodificación de entropía 70 puede configurarse además para ajustar el valor del parámetro de Rice usado para definir los códigos de Golomb para la codificación de nivel de coeficiente en base a si se aplica una transformación a un bloque de transformación que incluye el CG. De acuerdo con las técnicas descritas en esta divulgación, cuando la transformación no se aplica al bloque de transformación, es decir, en el caso de salto o derivación de la transformación, la unidad de decodificación de entropía 70 puede realizar la actualización del parámetro de Rice y los esquemas de inicialización del parámetro de Rice que son flexibles para adaptarse a valores absolutos grandes de niveles de coeficiente para los coeficientes del CG. Por otro lado, cuando la transformación se aplica al bloque de transformación, la unidad de decodificación de entropía 70 puede aplicar los esquemas de inicialización y actualización del parámetro de Rice del HEVC, que se optimizan para valores de coeficientes pequeños de coeficientes transformados.

La unidad de cuantificación inversa 76 cuantifica inversamente, es decir, descuantifica, coeficientes de transformación cuantificados proporcionados en el flujo de bits y decodificados por la unidad de decodificación de entropía 70. El procedimiento de cuantificación inversa puede incluir un procedimiento convencional, por ejemplo, similar a los procedimientos propuestos por el HEVC o definidos por el estándar de decodificación H.264. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP calculado por el codificador de vídeo 20 por la CU para determinar un grado de cuantificación e, igualmente, un grado de cuantificación inversa que debería aplicarse. La unidad de cuantificación inversa 76 puede cuantificar inversamente los coeficientes de transformación ya sea antes o después de que los coeficientes se conviertan de una matriz unidimensional a una matriz bidimensional.

La unidad de procesamiento de transformación inversa 78 aplica una transformación inversa a los coeficientes de transformación cuantificados inversos. En algunos ejemplos, la unidad de procesamiento de transformación inversa 78 puede determinar una transformación inversa en base a la señalización del codificador de vídeo 20, o infiriendo la transformación a partir de una o más características de codificación como el tamaño de bloque, el modo de codificación o similares. En algunos ejemplos, la unidad de procesamiento de transformación inversa 78 puede determinar una transformación para aplicar al bloque actual en base a una transformación señalizada en el nodo raíz

de un árbol cuaternario para una LCU que incluye el bloque actual. Alternativamente, la transformación puede señalizarse en la raíz de un árbol cuaternario de TU para un nodo hoja de CU en el árbol cuaternario de LCU. En algunos ejemplos, la unidad de procesamiento de transformación inversa 78 puede aplicar una transformación inversa en cascada, en la que la unidad de procesamiento de transformación inversa 78 aplica dos o más transformaciones inversas a los coeficientes de transformación del bloque actual que se decodifica.

La unidad de procesamiento de intrapredicción 74 puede generar datos de predicción para un bloque actual de un fotograma actual en base a un modo de intrapredicción señalado y datos de bloques previamente decodificados del fotograma actual.

La unidad de compensación de movimiento 72 puede recuperar el vector de movimiento, la dirección de predicción de movimiento y el índice de referencia del flujo de bits codificado. La dirección de predicción de referencia indica si el modo de interpolación es unidireccional (por ejemplo, un fotograma P) o bidireccional (un fotograma B). El índice de referencia indica en qué fotograma de referencia se basa el vector de movimiento candidato. En base a la dirección de predicción de movimiento recuperada, el índice de fotograma de referencia y el vector de movimiento, la unidad de compensación de movimiento 72 produce un bloque compensado por movimiento para la porción actual. Estos bloques compensados por movimiento esencialmente recrean el bloque predictivo usado para producir los datos residuales.

La unidad de compensación de movimiento 72 puede producir los bloques compensados por movimiento, posiblemente al realizar una interpolación en base a los filtros de interpolación. Los identificadores para los filtros de interpolación a usarse para la estimación de movimiento con precisión de subpíxeles pueden incluirse en los elementos de sintaxis. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación del bloque de vídeo para calcular valores interpolados para píxeles sub-enteros de un bloque de referencia. La unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo con la información de sintaxis que se recibe y usar los filtros de interpolación para producir bloques predictivos.

Adicionalmente, la unidad de compensación de movimiento 72 y la unidad de procesamiento de intrapredicción 74, en un ejemplo del HEVC, pueden usar alguna de la información de sintaxis (por ejemplo, la proporcionada por un árbol cuaternario) para determinar los tamaños de LCU usados para codificar los fotogramas de la secuencia de vídeo codificada. La unidad de compensación de movimiento 72 y la unidad de procesamiento de intrapredicción 74 también pueden usar información de sintaxis para determinar información dividida que describe cómo se divide cada CU de un fotograma de la secuencia de vídeo codificada (e igualmente, cómo se dividen las sub-CU). La información de sintaxis también puede incluir modos que indiquen cómo se codifica cada división (por ejemplo., intrapredicción o interpolación, y para intrapredicción un modo de codificación intrapredicción), uno o más fotogramas de referencia (y/o listas de referencia que contienen identificadores para los fotogramas de referencia) para cada PU intercodificada, y otra información para decodificar la secuencia de vídeo que se codifica.

El sumador 80 combina los bloques residuales con los correspondientes bloques de predicción generados por la unidad de compensación de movimiento 72 o la unidad de procesamiento de intrapredicción 74 para formar bloques decodificados. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados con el fin de eliminar los artefactos de bloque. Los bloques de vídeo decodificados se almacenan entonces en la memoria intermedia de imagen decodificada 82, que proporciona bloques de referencia para la compensación de movimiento posterior y también produce vídeo decodificado para la presentación en un dispositivo de visualización (como el dispositivo de visualización 32 de la figura 1).

La Figura 7 es un diagrama de flujo que ilustra una operación de ejemplo de actualización de un valor de un parámetro de Rice durante la codificación de entropía de niveles de coeficiente de acuerdo con las técnicas descritas en esta divulgación. El ejemplo de la operación se describe con respecto al codificador de vídeo 20 que incluye la unidad de codificación de entropía 56 de la figura. 5.

La unidad de codificación de entropía 56 recibe coeficientes de datos de vídeo residuales para codificarse en un flujo de bits para su transmisión al decodificador de vídeo 30 o para su almacenamiento en el medio de almacenamiento 34 o en el servidor de archivos 36. Al recibir los coeficientes de datos de vídeo residuales para la codificación, la unidad de codificación de entropía 56 determina un bloque o grupo de coeficientes (CG) de los coeficientes (100). El CG puede ser un subbloque de un bloque de transformación, como un subbloque nxn (por ejemplo, 4x4) como se ilustra en la figura. 4.

Como se describió anteriormente con más detalle, la unidad de codificación de entropía 56 puede realizar uno o más escaneos del CG de acuerdo con una orden de escaneo inverso para codificar niveles de coeficiente de los coeficientes en el CG. Por ejemplo, la unidad de codificación de entropía 56 puede codificar banderas o elementos de sintaxis para indicar si los coeficientes tienen niveles de coeficiente con valores absolutos que son mayores que 1 o mayores que 2 mediante el uso de modelos de contexto en un modo de codificación regular. La unidad de codificación de entropía 56 codifica entonces una bandera o un elemento de sintaxis para indicar un valor absoluto restante de un nivel de coeficiente para al menos uno de los coeficientes en el CG mediante el uso de códigos de

Golomb definidos por un parámetro de Rice (102). Por ejemplo, la unidad de codificación de entropía 56 puede codificar un elemento de sintaxis que indica un valor absoluto restante de un nivel de coeficiente para cualquiera de los coeficientes con niveles de coeficiente mayores que 2 en modo de derivación.

De acuerdo con las técnicas de esta divulgación, la unidad de codificación de entropía 56 actualiza un valor del parámetro de Rice en base a una selección de un mínimo o de un valor máximo del parámetro de Rice o un incremento de la variable del valor del parámetro de Rice (104). Como se describió en detalle anteriormente, una cantidad del incremento de la variable se determina en base al valor del parámetro de Rice y un valor absoluto del nivel de coeficiente para el coeficiente que se codifica. En algunos ejemplos, la cantidad del incremento de la variable puede determinarse en base a una versión desplazada a la derecha del valor absoluto del nivel de coeficiente para el coeficiente actual, donde una cantidad del desplazamiento a la derecha se basa en el valor del parámetro de Rice. En algunos ejemplos, la unidad de codificación de entropía 56 puede usar el valor actualizado del parámetro de Rice para definir los códigos de Golomb para la codificación de un valor absoluto restante de un nivel de coeficiente para un coeficiente posterior en el CG.

La Figura 8 es un diagrama de flujo que ilustra una operación de ejemplo de actualización de un valor de un parámetro de Rice durante la decodificación de entropía de niveles de coeficiente de acuerdo con las técnicas descritas en esta divulgación. El ejemplo de la operación se describe con respecto al decodificador de vídeo 30 que incluye la unidad de decodificación de entropía 70 de la figura. 6.

El decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado desde el codificador de vídeo 20 o desde un dispositivo de almacenamiento como un medio de almacenamiento 34 o un servidor de archivos 36. El flujo de bits de vídeo que se recibe representa coeficientes de datos de vídeo residuales a decodificarse. Al recibir el flujo de bits de vídeo, la unidad de decodificación de entropía 70 determina un bloque o grupo de coeficientes (CG) de coeficientes de datos de vídeo residuales a decodificarse desde el flujo de bits (110). El CG puede ser un subbloque de un bloque de transformación, como un subbloque 4x4 como se ilustra en la figura. 4.

Como se describió anteriormente con más detalle, la unidad de decodificación de entropía 70 puede realizar uno o más escaneos del CG de acuerdo con una orden de escaneo inverso para decodificar los niveles de coeficiente de los coeficientes en el CG. Por ejemplo, la unidad de decodificación de entropía 70 puede decodificar banderas o elementos de sintaxis que indican si los coeficientes tienen niveles de coeficiente con valores absolutos mayores que 1 o mayores que 2 mediante el uso de modelos de contexto en un modo de codificación regular. La unidad de decodificación de entropía 70 también decodifica una bandera o un elemento de sintaxis que indica un valor absoluto restante de un nivel de coeficiente para al menos uno de los coeficientes en el CG mediante el uso de códigos de Golomb definidos por un parámetro de Rice (112). Por ejemplo, la unidad de decodificación de entropía 70 puede decodificar un elemento de sintaxis que indica un valor absoluto restante de un nivel de coeficiente para cualquiera de los coeficientes con niveles de coeficiente mayores que 2 en modo de derivación.

De acuerdo con las técnicas de esta divulgación, la unidad de decodificación de entropía 70 actualiza un valor del parámetro de Rice en base a una selección de un mínimo o de un valor máximo del parámetro de Rice o un incremento variable del valor del parámetro de Rice (114). Como se describió en detalle anteriormente, una cantidad del incremento de la variable se determina en base al valor del parámetro de Rice y un valor absoluto del nivel de coeficiente para el coeficiente que se decodifica. En algunos ejemplos, la cantidad del incremento de la variable puede determinarse en base a una versión desplazada a la derecha del valor absoluto del nivel de coeficiente para el coeficiente actual, donde una cantidad del desplazamiento a la derecha se basa en el valor del parámetro de Rice. En algunos ejemplos, la unidad de decodificación de entropía 70 puede usar el valor actualizado del parámetro de Rice para definir los códigos de Golomb para la decodificación de un valor absoluto restante de un nivel de coeficiente para un coeficiente posterior en el CG.

La Figura 9 es un diagrama de flujo que ilustra una operación de ejemplo de ajuste de un valor de un parámetro de Rice durante la codificación de entropía de niveles de coeficiente de acuerdo con las técnicas descritas en esta divulgación. El ejemplo de la operación se describe con respecto al codificador de vídeo 20 que incluye la unidad de codificación de entropía 56 de la figura. 5.

La unidad de codificación de entropía 56 recibe coeficientes de datos de vídeo residuales para codificarse en un flujo de bits para su transmisión al decodificador de vídeo 30 o para su almacenamiento en el medio de almacenamiento 34 o en el servidor de archivos 36. Al recibir los coeficientes de datos de vídeo residuales para la codificación, la unidad de codificación de entropía 56 determina un bloque o grupo de coeficientes (CG) de los coeficientes (120). El CG puede ser un subbloque de un bloque de transformación, como un subbloque 4x4 como se ilustra en la figura. 4. En el caso de codificación de vídeo con pérdida, los coeficientes a codificarse pueden ser coeficientes de transformación cuantificados. En el caso de codificación de vídeo sin pérdida y codificación de vídeo con pérdida con salto o derivación de transformación, se saltan las operaciones de transformación y cuantificación para los coeficientes a codificarse.

Como se describió anteriormente con más detalle, la unidad de codificación de entropía 56 puede realizar uno o más escaneos del CG de acuerdo con una orden de escaneo inverso para codificar niveles de coeficiente de los

coeficientes en el CG. Por ejemplo, la unidad de codificación de entropía 56 codifica una bandera o un elemento de sintaxis que indica un valor absoluto restante de un nivel de coeficiente para al menos uno de los coeficientes en el CG mediante el uso de códigos de Golomb definidos por un parámetro de Rice (122). De acuerdo con las técnicas de esta divulgación, la unidad de codificación de entropía 56 ajusta un valor del parámetro de Rice en base a si se aplicó una transformación al bloque de transformación que incluye el CG (124). La unidad de codificación de entropía 56 puede determinar si la transformación se aplica al bloque de transformación en base a al menos uno de un estado de codificación sin pérdida, un estado de derivación de transformación, un estado de salto de transformación o un perfil que se establece en un conjunto de parámetro para el bloque de transformación.

En un ejemplo, de acuerdo con las técnicas de esta divulgación, si la transformación no se aplica al bloque de transformación, el esquema de actualización del parámetro de Rice descrito en esta divulgación puede usarse para actualizar el valor del parámetro de Rice para codificar los valores absolutos restantes de los niveles de coeficiente para un coeficiente posterior en el CG. De cualquier otra manera, si la transformación se aplica al bloque de transformación, el esquema de actualización del parámetro de Rice del HEVC puede usarse para actualizar el valor del parámetro de Rice para un coeficiente posterior en el CG. En otro ejemplo, de acuerdo con las técnicas, si la transformación no se aplica al bloque de transformación, el valor del parámetro de Rice puede inicializarse para un CG posterior en el bloque de transformación en base al valor del parámetro de Rice después de la codificación de los valores absolutos restantes de los niveles de coeficiente para los coeficientes en el CG actual. De cualquier otra manera, si la transformación se aplica al bloque de transformación, el valor del parámetro de Rice puede inicializarse para que sea igual a cero para un CG posterior en el bloque de transformación.

La Figura 10 es un diagrama de flujo que ilustra una operación de ejemplo de ajuste de un valor de un parámetro de Rice durante la decodificación de entropía de niveles de coeficiente de acuerdo con las técnicas descritas en esta divulgación. La operación de ejemplo se describe con respecto al decodificador de vídeo 20 que incluye la unidad de decodificación de entropía 70 de la figura. 6.

El decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado desde el codificador de vídeo 20 o desde un dispositivo de almacenamiento como un medio de almacenamiento 34 o un servidor de archivos 36. El flujo de bits de vídeo que se recibe representa coeficientes de datos de vídeo residuales a decodificarse. Al recibir el flujo de bits de vídeo, la unidad de decodificación de entropía 70 determina un bloque o grupo de coeficientes (CG) de coeficientes de datos de vídeo residuales a decodificar desde el flujo de bits (130). El CG puede ser un subbloque de un bloque de transformación, como un subbloque 4x4 como se ilustra en la figura. 4. En el caso de la codificación de vídeo con pérdida, los coeficientes a decodificar pueden ser coeficientes de transformación cuantificados. En el caso de codificación de vídeo sin pérdida y codificación de vídeo con pérdida con salto o derivación de transformación, se saltan las operaciones de transformación y cuantificación para los coeficientes a decodificar.

Como se describió anteriormente con más detalle, la unidad de decodificación de entropía 70 puede realizar uno o más escaneos del CG de acuerdo con una orden de escaneo inverso para decodificar los niveles de coeficiente de los coeficientes en el CG. Por ejemplo, la unidad de decodificación de entropía 70 decodifica una bandera o un elemento de sintaxis que indica un valor absoluto restante de un nivel de coeficiente para al menos uno de los coeficientes en el CG mediante el uso de códigos de Golomb definidos por un parámetro de Rice (132). De acuerdo con las técnicas de esta divulgación, la unidad de decodificación de entropía 70 ajusta un valor del parámetro de Rice en base a si se aplicó una transformación a un bloque de transformación que incluye el CG (134). La unidad de decodificación de entropía 70 puede determinar si la transformación se aplica al bloque de transformación en base a al menos uno de un estado de codificación sin pérdida, un estado de derivación de la transformación, un estado de salto de transformación o un perfil que se establece en un conjunto de parámetro para el bloque de transformación.

En un ejemplo, de acuerdo con las técnicas de esta divulgación, si la transformación no se aplica al bloque de transformación, el esquema de actualización del parámetro de Rice descrito en esta divulgación puede usarse para actualizar el valor del parámetro de Rice para codificar los valores absolutos restantes de los niveles de coeficiente para un coeficiente posterior en el CG. De cualquier otra manera, si la transformación se aplica al bloque de transformación, el esquema de actualización del parámetro de Rice del HEVC puede usarse para actualizar el valor del parámetro de Rice para un coeficiente posterior en el CG. En otro ejemplo, de acuerdo con las técnicas, si la transformación no se aplica al bloque de transformación, el valor del parámetro de Rice puede inicializarse para un CG posterior en el bloque de transformación en base al valor del parámetro de Rice después de la decodificación de los valores absolutos restantes de los niveles de coeficiente para los coeficientes en el CG actual. De cualquier otra manera, si la transformación se aplica al bloque de transformación, el valor del parámetro de Rice puede inicializarse para que sea igual a cero para un CG posterior en el bloque de transformación.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, microprograma o cualquiera de sus combinaciones. Si se implementan en software, las funciones pueden almacenarse en un medio legible por ordenador o transmitirse a través de él, 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 corresponden a un medio tangible, como medios de almacenamiento de datos, o medios de comunicación, que incluyen cualquier medio que facilite transferir 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 generalmente pueden corresponder a (1) medios de almacenamiento legibles por ordenador tangibles que no son transitorios o (2) un medio de comunicación como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder por uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A manera de ejemplo, y no de limitación, tales medios legibles por ordenador pueden comprender RAM, ROM, EPROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnéticos, memoria flash o cualquier otro medio que pueda usarse para almacenar el código del programa deseado en forma de instrucciones o estructuras de datos y al que puede accederse por un ordenador. Además, cualquier conexión se denomina correctamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota mediante el uso de un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. Disco, como se usa en la presente memoria, incluye disco compacto (CD), disco de láser, disco óptico, disco digital versátil (DVD), disquete, y disco de Blu-ray donde hay discos que usualmente reproducen los datos magnéticamente, mientras que otros discos reproducen los datos ópticamente con láseres. Las combinaciones de los medios anteriores pueden incluirse además dentro del ámbito de los medios legibles por ordenador.

Las instrucciones pueden ejecutarse por uno o más procesadores, como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados para aplicaciones específicas (ASIC), matriz de puertas lógicas programable en campo (FPGA) u otro circuito integrado lógico o discreto equivalente. En consecuencia, el término "procesador", como se usa en la presente memoria puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen varios componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de hardware. Más bien, como se describió anteriormente, varias unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describió anteriormente, junto con software y/o microprograma adecuados.

REIVINDICACIONES

1. Un proceso de coeficientes de decodificación en un proceso de decodificación de vídeo, comprendiendo el procedimiento:
 - determinar (110, 130) un bloque actual de coeficientes (38) de datos de vídeo residuales, en el que el bloque de coeficientes comprende un primer grupo de coeficientes, CG (39) y un segundo CG que es posterior al primer CG en el bloque de coeficientes;
 - decodificar, para un coeficiente en el primer CG y mediante el uso de un modelo de contexto adaptativo, una indicación de que un nivel de coeficiente para el coeficiente tiene un valor absoluto mayor que dos;
 - decodificar (112, 132), mediante el uso de códigos de Golomb definidos por un parámetro de Rice, un primer valor para su uso en la determinación del valor absoluto del nivel de coeficiente para el coeficiente en el primer CG;
 - determinar el valor absoluto del nivel de coeficiente para que el coeficiente sea igual a tres más dicho primer valor decodificado; ajustar (114, 134) un valor del parámetro de Rice para el segundo CG en base a si se aplica una transformación a un bloque de transformación que incluye el bloque actual de coeficientes, en el que se usa un primer proceso para ajustar el valor del parámetro de Rice si no se usa una transformación y se usa un segundo proceso para ajustar el valor del parámetro de Rice si se usa una transformación, el segundo proceso es diferente al primer proceso;
 - decodificar un segundo valor para su uso en la determinación del valor absoluto de un nivel de coeficiente para un coeficiente en el segundo CG mediante el uso de códigos de Golomb definidos por el valor ajustado del parámetro de Rice; y
 - determinar el valor absoluto del nivel de coeficiente para que el coeficiente en el segundo CG sea igual a tres más dicho segundo valor decodificado.
2. El procedimiento de la reivindicación 1, en el que el primer proceso de ajustar el valor del parámetro de Rice, usado si la transformación no se aplica al bloque de transformación, comprende actualizar el valor del parámetro de Rice en base a una selección de un mínimo o de un valor máximo del parámetro de Rice o un incremento de la variable del valor del parámetro de Rice, en el que el incremento de la variable se determina en base al valor del parámetro de Rice y un valor absoluto del nivel de coeficiente para el coeficiente que se decodifica.
3. El procedimiento de la reivindicación 2, en el que el incremento de la variable o bien:
 - a) aumenta el valor del parámetro de Rice por cualquier valor entero hasta el valor máximo del parámetro de Rice; o
 - b) se determina en base a una versión desplazada a la derecha del valor absoluto del nivel de coeficiente para el coeficiente que se decodifica, en el que una cantidad del desplazamiento a la derecha se basa en el valor del parámetro de Rice.
4. El procedimiento de la reivindicación 2, en el que una cantidad del incremento de la variable en cada actualización del valor del parámetro de Rice es una de la escalada por un factor de amortiguación o limitada para ser menor o igual a un valor de incremento máximo.
5. El procedimiento de la reivindicación 1, en el que el primer proceso de ajustar el valor del parámetro de Rice, usado si la transformación no se aplica al bloque de transformación, comprende disminuir el valor del parámetro de Rice en base al valor del parámetro de Rice y el valor absoluto del nivel de coeficiente para el coeficiente que se decodifica.
6. El procedimiento de la reivindicación 1, en el que el bloque de coeficientes comprende un primer bloque de coeficientes en el bloque de transformación, y en el que el primer proceso de ajustar el valor del parámetro de Rice, usado si la transformación no se aplica al bloque de transformación, comprende inicializar el valor del parámetro de Rice para un segundo bloque de coeficientes en el bloque de transformación que es posterior al primer bloque de coeficientes en el bloque de transformación en base al valor del parámetro de Rice después de la decodificación del valor absoluto restante del nivel de coeficiente para el al menos un coeficiente en el primer bloque de coeficientes.
7. El procedimiento de la reivindicación 1, en el que el segundo proceso de ajustar el valor del parámetro de Rice, usado si la transformación se aplica al bloque de transformación, comprende inicializar el valor del parámetro de Rice para que sea igual a cero para el segundo bloque de coeficientes en el bloque de transformación.
8. El procedimiento de la reivindicación 1, que comprende además determinar si la transformación se aplica al bloque de transformación en base a al menos uno de un estado de codificación sin pérdida, un estado de derivación de la transformación, un estado de salto de la transformación o un perfil que se establece en un conjunto de parámetro para el bloque de transformación.

9. Un procedimiento de coeficientes de codificación en un procedimiento de codificación de vídeo, comprendiendo el procedimiento

determinar (100, 120) un bloque actual de coeficientes de datos de vídeo residuales, en el que el bloque de coeficientes comprende un primer grupo de coeficientes, CG (39) y un segundo CG que es posterior al primer CG en el bloque de coeficientes;
 codificar, para un coeficiente en el primer CG y mediante el uso de un modelo de contexto adaptativo, una indicación de que un nivel de coeficiente para el coeficiente tiene un valor absoluto mayor que dos;
 codificar (102, 122), mediante el uso de códigos de Golomb definidos por un parámetro de Rice, un primer valor que es igual a menos tres más el valor absoluto del nivel de coeficiente para el coeficiente en el primer CG; y ajustar (104, 124) un valor del parámetro de Rice para el segundo CG en base a si se aplica una transformación a un bloque de transformación que incluye el bloque actual de coeficientes, en el que se usa un primer proceso para ajustar el valor del parámetro de Rice si no se usa una transformación y se usa un segundo proceso para ajustar el valor del parámetro de Rice si se usa una transformación, el segundo proceso es diferente al primer proceso; y
 codificar, para un coeficiente en el segundo CG y mediante el uso de códigos de Golomb definidos por el valor ajustado del parámetro de Rice, un segundo valor que es igual a menos tres más el valor absoluto del nivel de coeficiente para el coeficiente en el segundo CG.

10. Un dispositivo de decodificación de vídeo que comprende:

una memoria configurada para almacenar datos de vídeo; y
 uno o más procesadores configurados para:

determinar un bloque actual de coeficientes de datos de vídeo residuales, en el que el bloque de coeficientes comprende un primer grupo de coeficientes, CG (39) y un segundo CG que es posterior al primer CG en el bloque de coeficientes;
 decodificar, para un coeficiente en el primer CG y mediante el uso de un modelo de contexto adaptativo, una indicación de que un nivel de coeficiente para el coeficiente tiene un valor absoluto mayor que dos;
 decodificar, mediante el uso de códigos de Golomb definidos por un parámetro de Rice, un primer valor para su uso en la determinación del valor absoluto del nivel de coeficiente para el coeficiente en el primer CG;
 determinar el valor absoluto del nivel de coeficiente para que el coeficiente sea igual a menos tres más dicho primer valor decodificado; y ajustar un valor del parámetro de Rice para el segundo CG en base a si se aplica una transformación a un bloque de transformación que incluye el bloque actual de coeficientes, en el que se usa un primer proceso para ajustar el valor del parámetro de Rice si no se usa una transformación y se usa un segundo proceso para ajustar el valor del parámetro de Rice si se usa una transformación, el segundo proceso es diferente al primer proceso;
 decodificar un segundo valor para su uso en la determinación del valor absoluto de un nivel de coeficiente para un coeficiente en el segundo CG mediante el uso de códigos de Golomb definidos por el valor ajustado del parámetro de Rice; y
 determinar el valor absoluto del nivel de coeficiente para que el coeficiente en el segundo CG sea igual a tres más dicho segundo valor decodificado.

11. Un dispositivo de codificación de vídeo que comprende:

una memoria configurada para almacenar datos de vídeo; y
 uno o más procesadores configurados para:

determinar un bloque actual de coeficientes de datos de vídeo residuales, en el que el bloque de coeficientes comprende un primer grupo de coeficientes, CG (39) y un segundo CG que es posterior al primer CG en el bloque de coeficientes;
 codificar, para un coeficiente en el primer CG y mediante el uso de un modelo de contexto adaptativo, una indicación de que un nivel de coeficiente para el coeficiente tiene un valor absoluto mayor que dos;
 codificar, mediante el uso de códigos de Golomb definidos por un parámetro de Rice, un primer valor que sea igual a menos tres más el valor absoluto del nivel de coeficiente para el coeficiente en el primer CG; y ajustar un valor del parámetro de Rice para el segundo CG en base a si se aplica una transformación a un bloque de transformación que incluye el bloque actual de coeficientes, en el que se usa un primer proceso para ajustar el valor del parámetro de Rice si no se usa una transformación y se usa un segundo proceso para ajustar el valor del parámetro de Rice si se usa una transformación, el segundo proceso es diferente al primer proceso; y
 codificar, para un coeficiente en el segundo CG y mediante el uso de códigos de Golomb definidos por el valor ajustado del parámetro de Rice, un segundo valor que sea igual a menos tres más el valor absoluto del nivel de coeficiente para el coeficiente en el segundo CG.

12. Un medio de almacenamiento legible por ordenador que comprende instrucciones que cuando se ejecutan por uno o más procesadores de un dispositivo de codificación de vídeo hacen que los procesadores lleven a cabo un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 9.

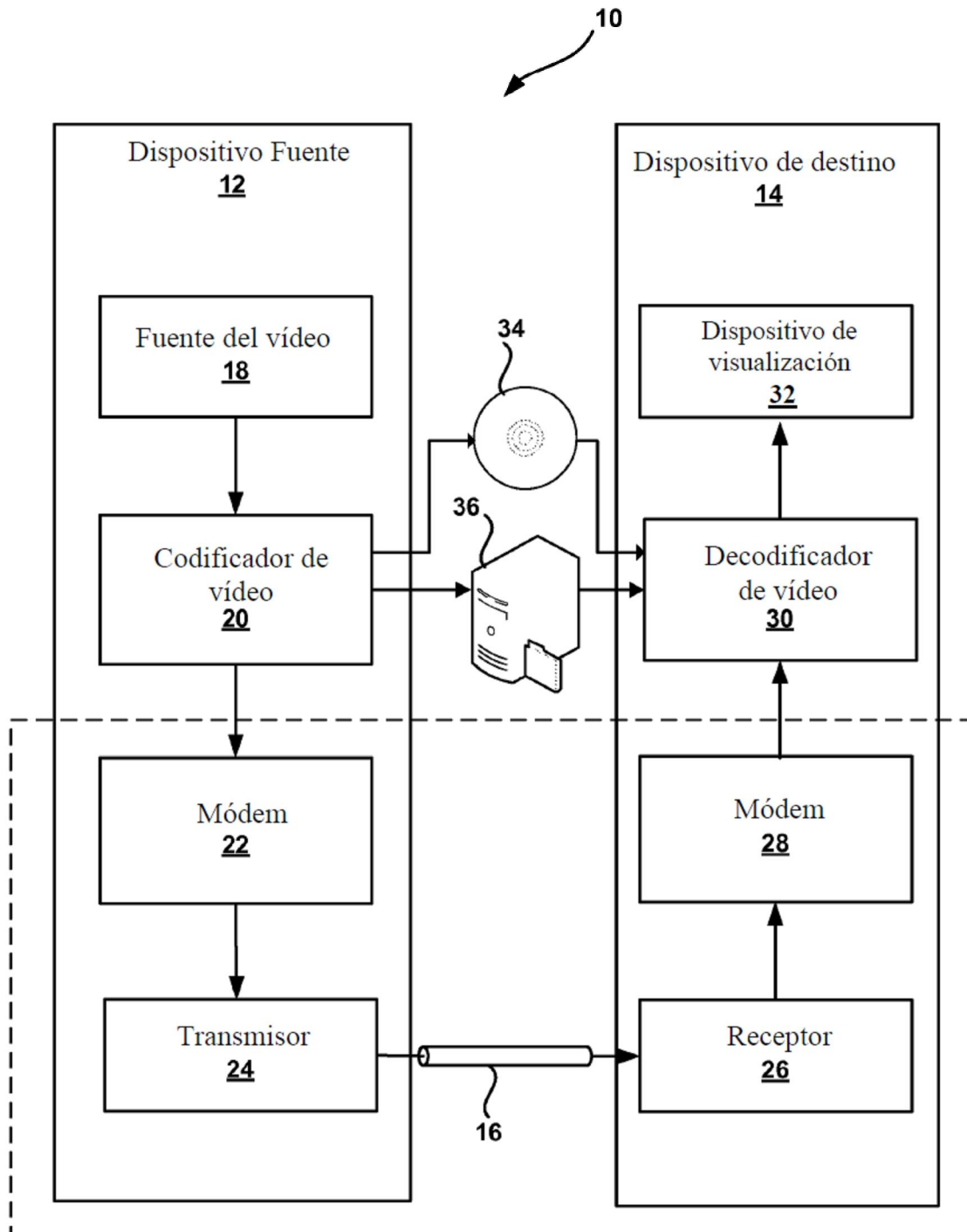


Figura 1

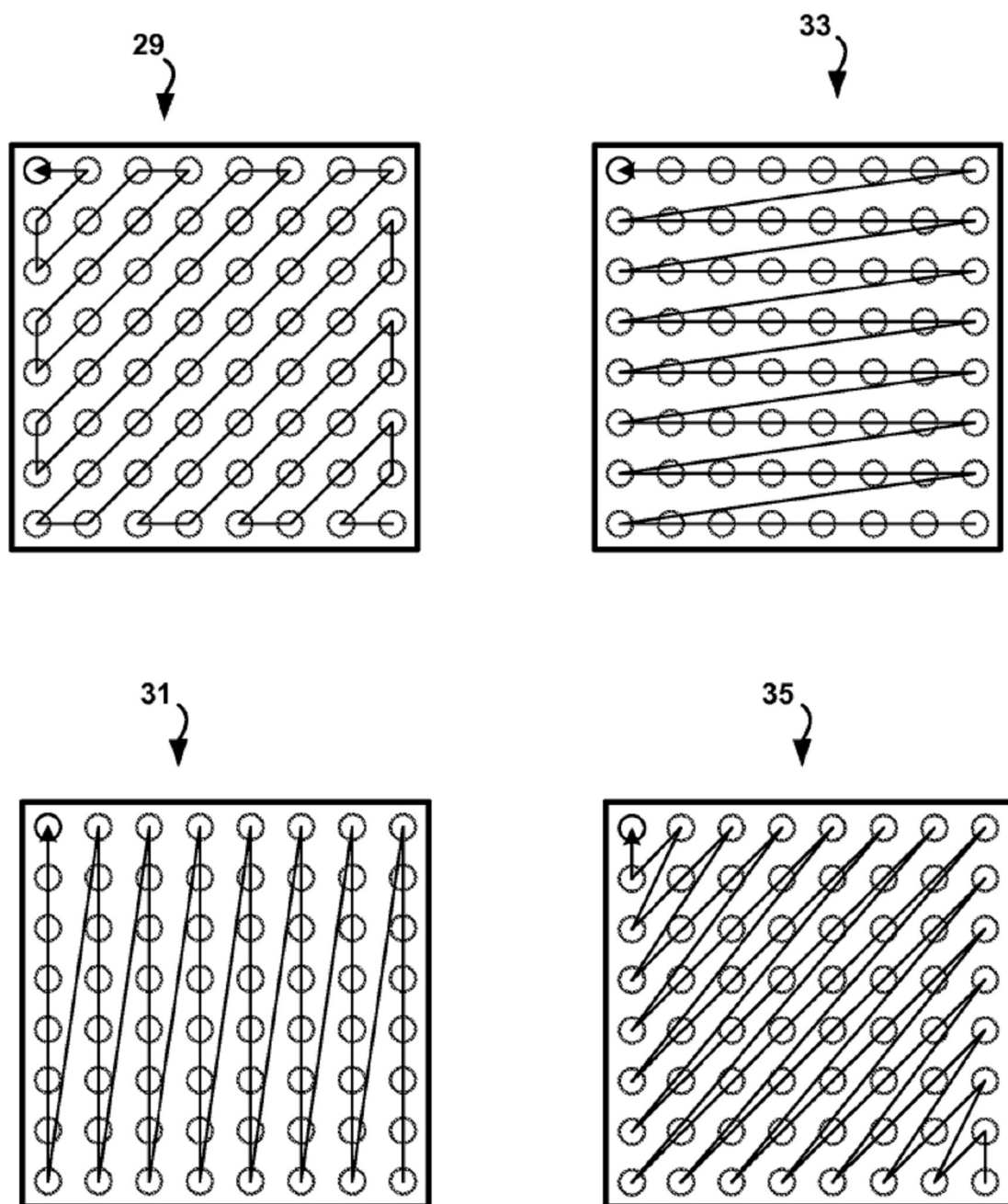


Figura 2

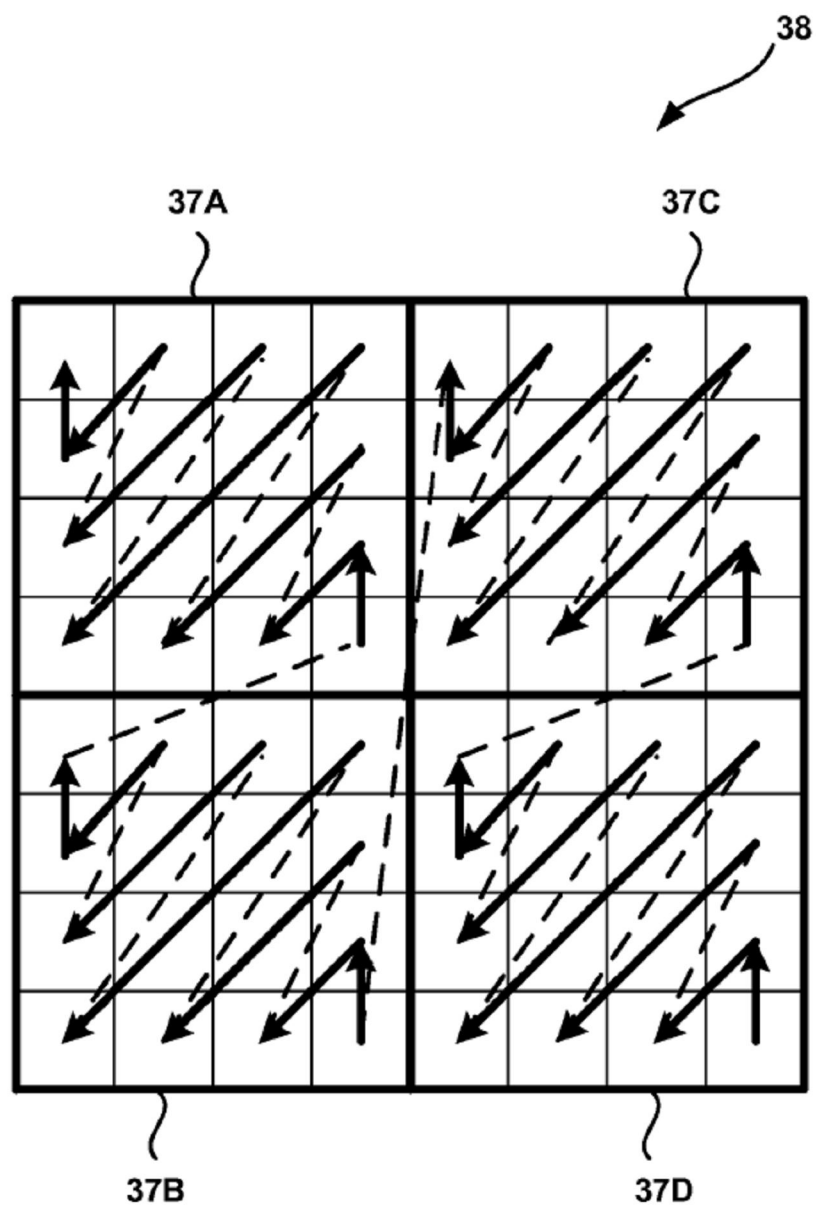


Figura 3

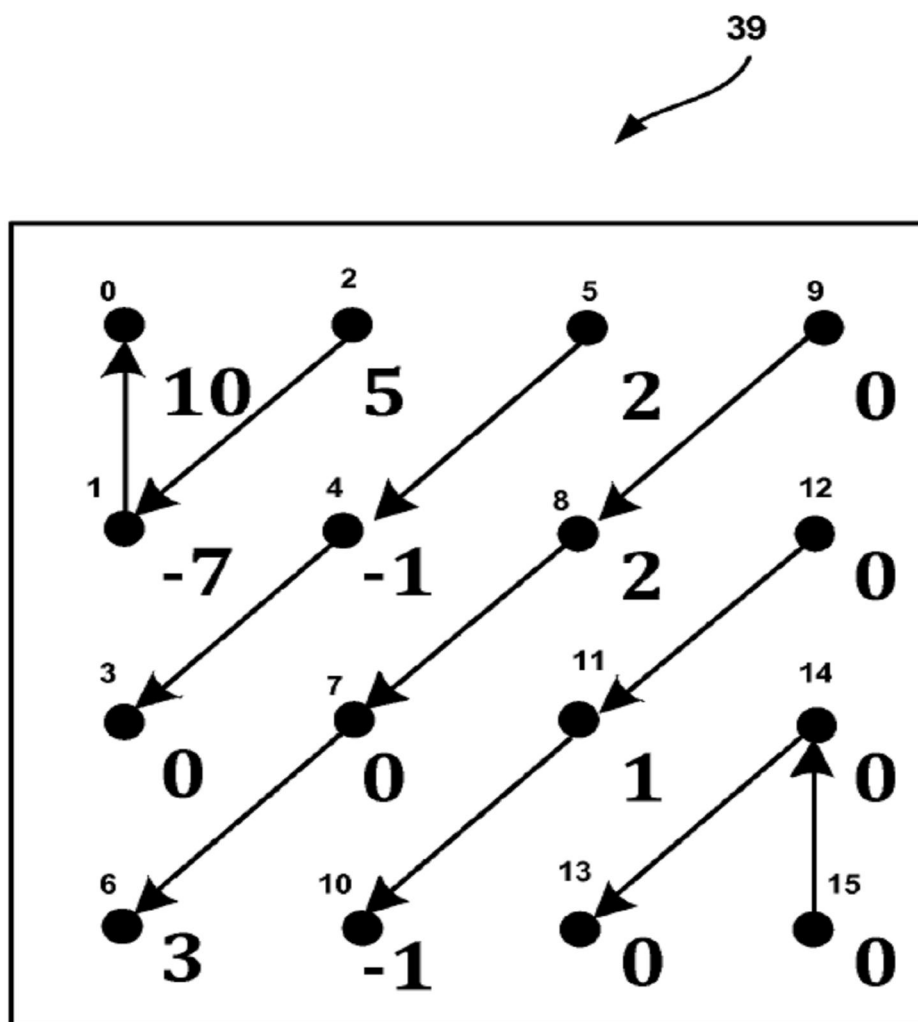


Figura 4

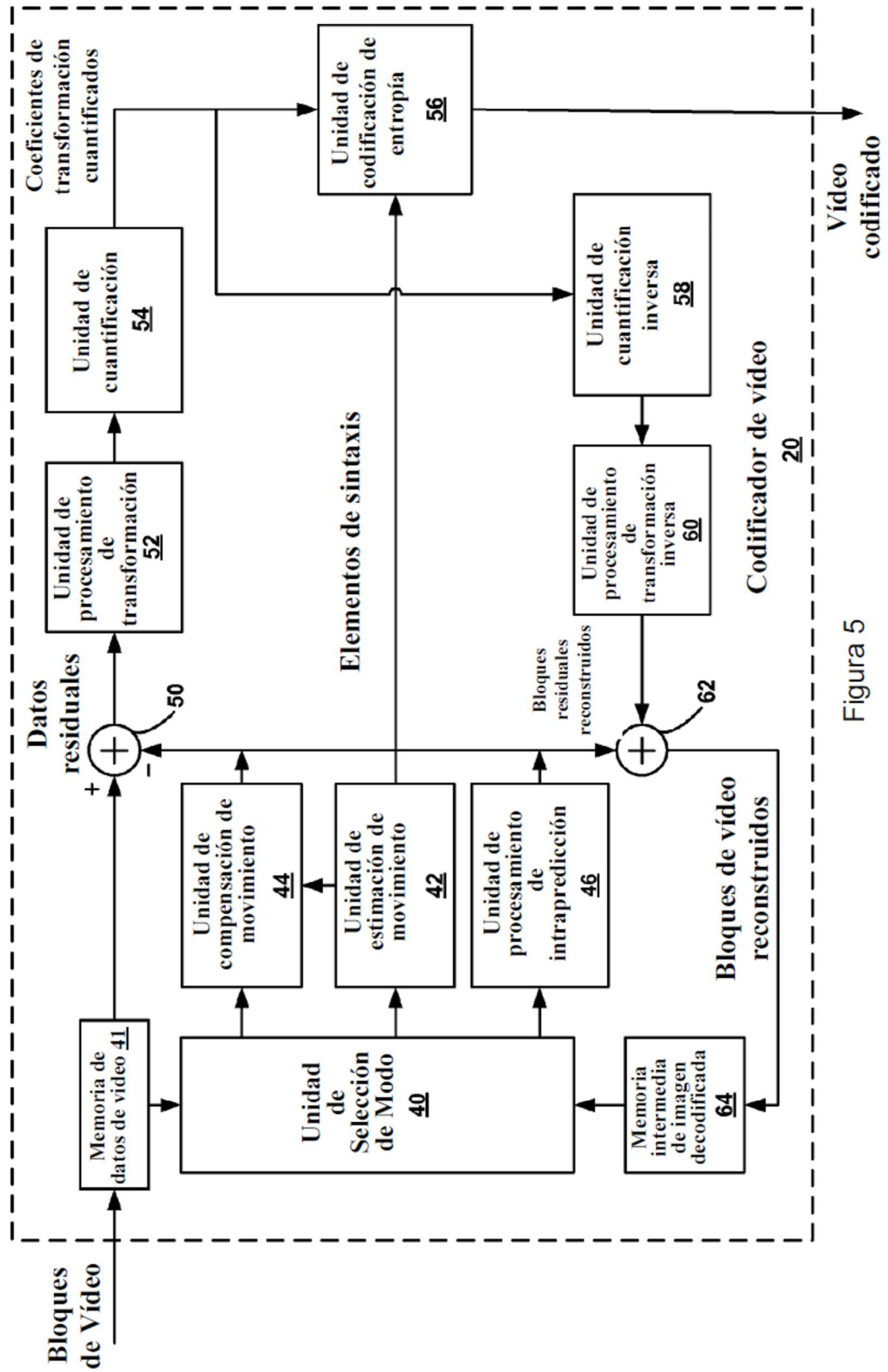


Figura 5

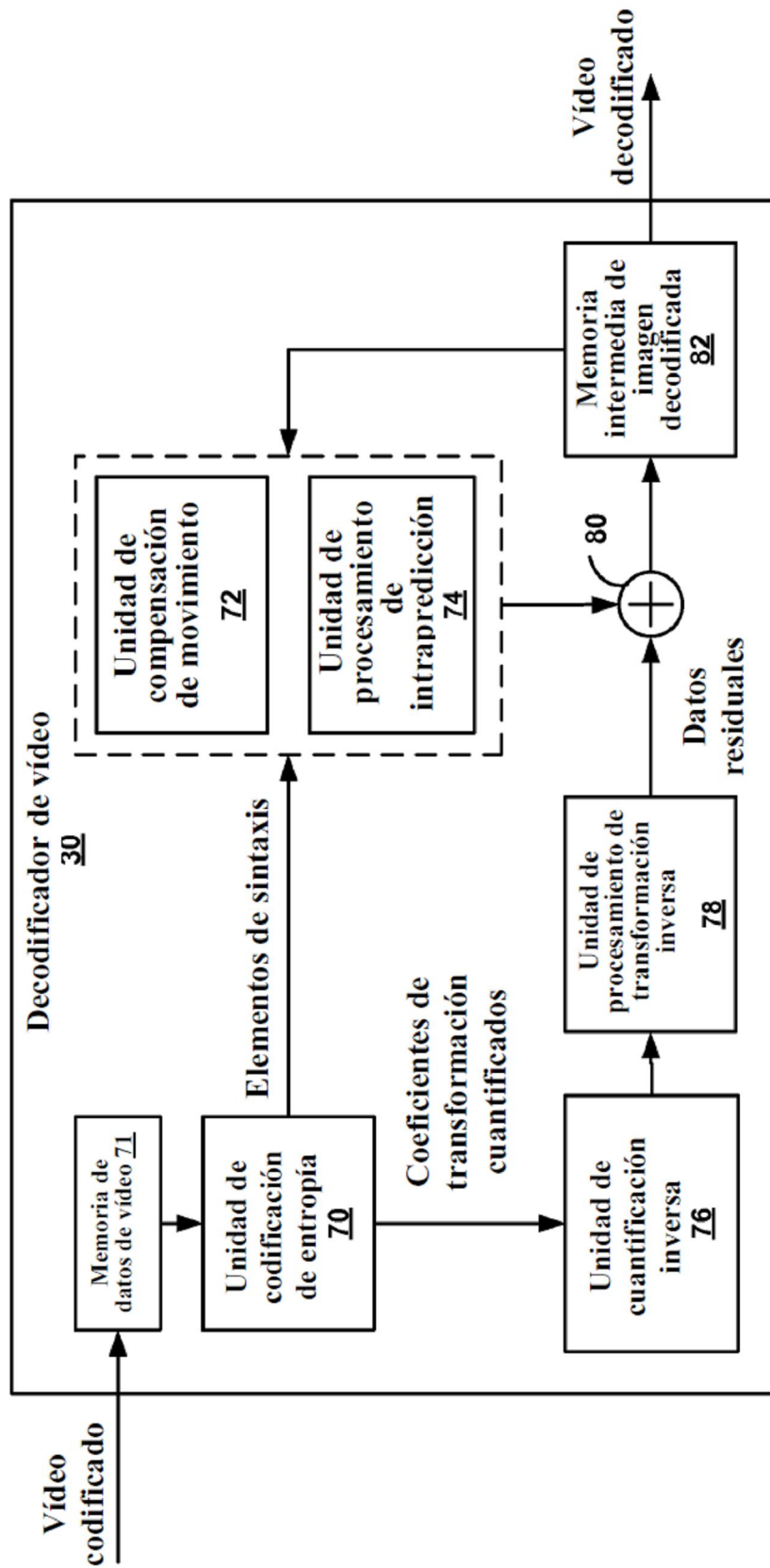


Figura 6

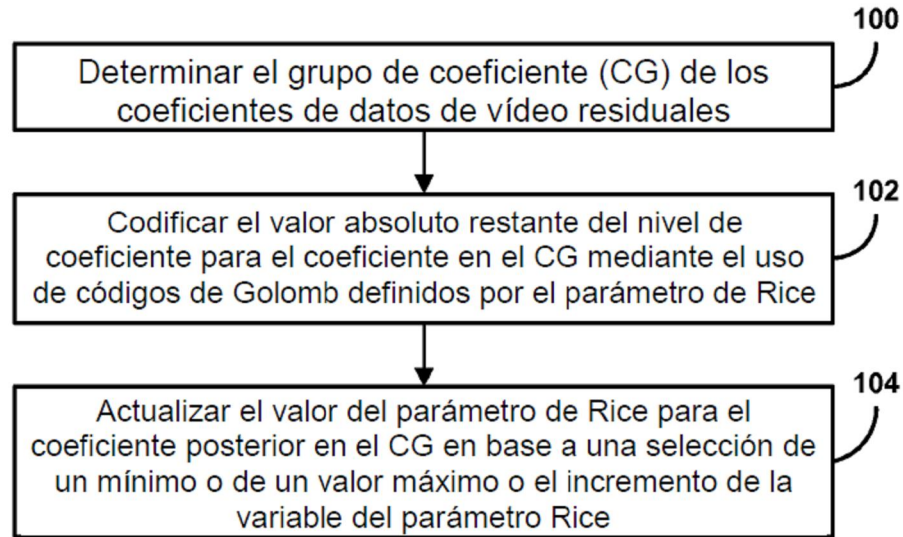


Figura 7

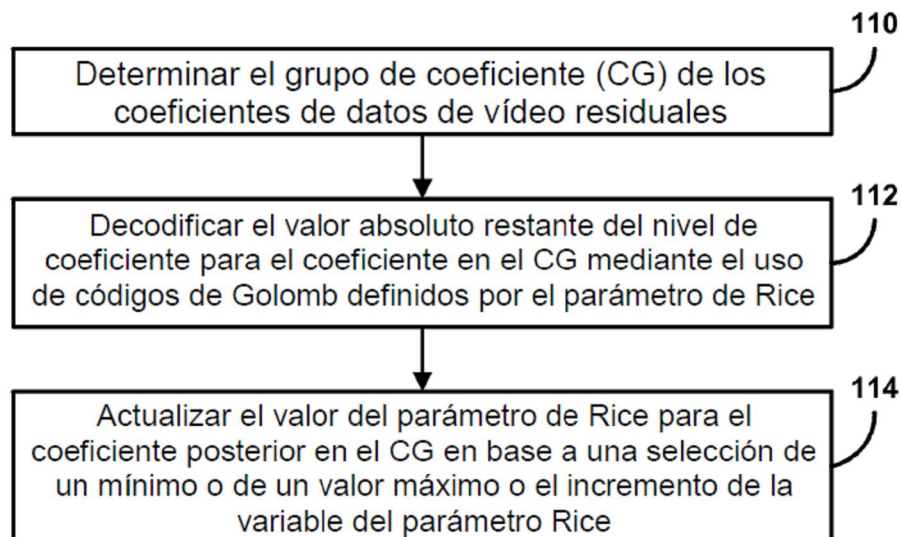


Figura 8

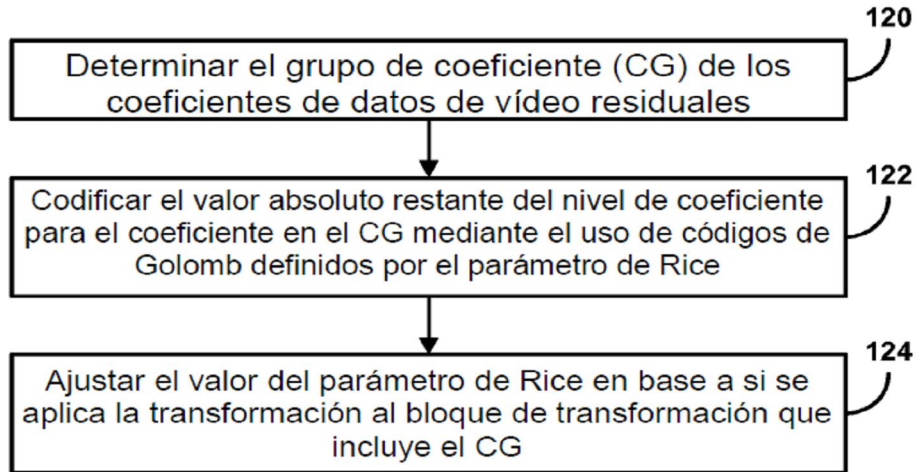


Figura 9

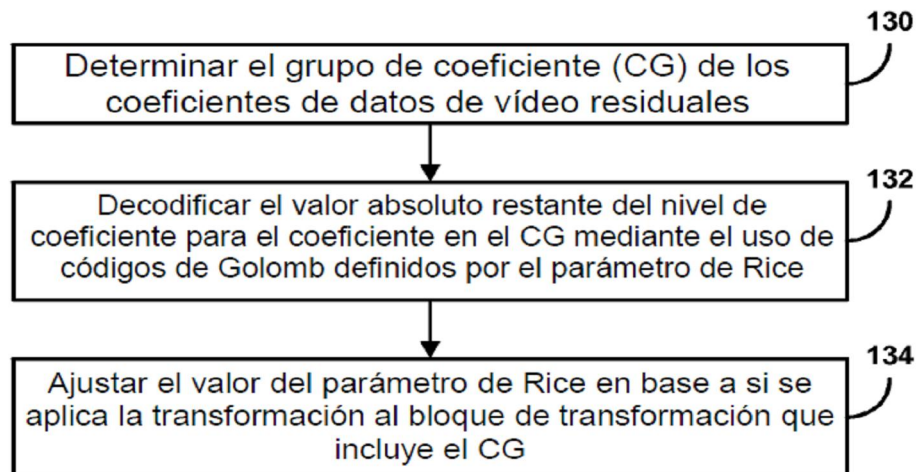


Figura 10