

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 014 586**

51 Int. Cl.:

H04N 19/117 (2014.01)

H04N 19/124 (2014.01)

H04N 19/132 (2014.01)

H04N 19/186 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **27.05.2021** **PCT/US2021/034559**

87 Fecha y número de publicación internacional: **07.04.2022** **WO22072002**

96 Fecha de presentación y número de la solicitud europea: **27.05.2021** **E 21876146 (8)**

97 Fecha y número de publicación de la concesión europea: **12.02.2025** **EP 4046380**

54 Título: **Método y aparato para desfase en el filtrado de vídeos**

30 Prioridad:

01.10.2020 US 202063086548 P

03.12.2020 US 202063121081 P

26.04.2021 US 202117240812

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
23.04.2025

73 Titular/es:

TENCENT AMERICA LLC (100.00%)

2747 Park Boulevard

Palo Alto, California 94306, US

72 Inventor/es:

DU, YIXIN;

ZHAO, XIN y

LIU, SHAN

74 Agente/Representante:

DEL VALLE VALIENTE, Sonia

ES 3 014 586 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

Método y aparato para desfase en el filtrado de vídeos

La presente solicitud reivindica el beneficio de prioridad para la solicitud de patente US-17/240.812, "METHOD AND APPARATUS FOR OFFSET IN VIDEO FILTERING" presentada el 26 de abril de 2021, que reivindica el beneficio de prioridad para la solicitud provisional US-63/086.548, "CROSS-COMPONENT SAMPLE OFFSET" presentada el 1 de octubre de 2020, y la solicitud provisional US-63/121.081, "LOCAL SAMPLE OFFSET" presentada el 3 de diciembre de 2020.

Campo técnico

La presente descripción describe realizaciones relacionadas generalmente con la codificación de vídeos.

Antecedentes

La descripción de antecedentes proporcionada en la presente descripción tiene el propósito de presentar generalmente el contexto de la descripción. El trabajo de los inventores actualmente mencionados, en la medida en que se describe el trabajo en esta sección de antecedentes, así como también aspectos de la descripción que de otro modo no califican como técnica anterior en el momento de la presentación, no se admiten expresa ni implícitamente como técnica anterior frente a la presente descripción.

La codificación y decodificación de vídeo se puede realizar usando predicción interimagen con compensación de movimiento. El vídeo digital no comprimido puede incluir una serie de imágenes, teniendo cada imagen una dimensión espacial de, por ejemplo, 1920 x 1080 muestras de luminancia y muestras de crominancia asociadas. La serie de imágenes puede tener una velocidad de imagen fija o variable (también conocida como velocidad de fotogramas), de, por ejemplo, 60 imágenes por segundo o 60 Hz. El vídeo no comprimido tiene requisitos de velocidad de bits específicos. Por ejemplo, el vídeo 1080p60 4:2:0 en 8 bits por muestra (resolución de muestra de luminancia 1920x1080 a una velocidad de fotogramas de 60 Hz) requiere un ancho de banda cercano a 1,5 Gbit/s. Una hora de tal vídeo requiere más de 600 GByte de espacio de almacenamiento.

Un propósito de la codificación y decodificación de vídeo puede ser la reducción de redundancia en la señal de vídeo de entrada, mediante compresión. La compresión puede ayudar a reducir los requisitos de ancho de banda y/o espacio de almacenamiento mencionados anteriormente, en algunos casos en dos órdenes de magnitud o más. Se pueden emplear tanto la compresión sin pérdidas como la compresión con pérdidas, así como una combinación de las mismas. La compresión sin pérdidas se refiere a técnicas donde se puede reconstruir una copia exacta de la señal original a partir de la señal original comprimida. Cuando se utiliza compresión con pérdidas, la señal reconstruida puede no ser idéntica a la señal original, pero la distorsión entre las señales original y reconstruida es lo suficientemente pequeña como para hacer la señal reconstruida útil para la aplicación prevista. En el caso de vídeos, la compresión con pérdidas se emplea ampliamente. La cantidad de distorsión tolerada depende de la aplicación; por ejemplo, los usuarios de ciertas aplicaciones de transmisión para consumidor pueden tolerar una mayor distorsión que los usuarios de aplicaciones para distribución de televisión. La relación de compresión alcanzable puede reflejar que: una distorsión permisible/tolerable más alta puede producir relaciones de compresión más altas.

Un codificador y decodificador de vídeo puede utilizar técnicas de varias categorías amplias, que incluyen, por ejemplo, compensación de movimiento, transformación, cuantificación y codificación por entropía.

Las tecnologías de códec de vídeo pueden incluir técnicas conocidas como intracodificación. En la intracodificación, los valores de muestra se representan sin referencia a muestras u otros datos de imágenes de referencia previamente reconstruidas. En algunos códecs de vídeo, la imagen se subdivide espacialmente en bloques de muestras. Cuando todos los bloques de muestras se codifican en intramodo, esa imagen puede ser una intrainimagen. Pueden utilizarse intrainágenes y sus derivaciones tales como imágenes de actualización de decodificadores independientes, para restablecer el estado del decodificador y, por lo tanto, pueden utilizarse como la primera imagen en un flujo de bits de vídeo codificado y una sesión de vídeo, o como una imagen fija. Las muestras de un intrabloque pueden exponerse a una transformación, y los coeficientes de transformación pueden cuantificarse antes de la codificación por entropía. La intrapredicción puede ser una técnica que minimiza los valores de muestra en el dominio de pre-transformación. En algunos casos, cuanto menor es el valor de CC después de una transformación, y cuanto más pequeños son los coeficientes de CA, menos bits se requieren en un tamaño del paso de cuantificación dada para representar el bloque después de la codificación por entropía.

La intracodificación tradicional tal como se conoce a partir de, por ejemplo, tecnologías de codificación de generación de MPEG-2 no utiliza intrapredicción. Sin embargo, algunas tecnologías de compresión de vídeo más recientes incluyen técnicas que intentan, por ejemplo, datos de muestra circundantes y/o metadatos obtenidos durante la codificación/decodificación de bloques de datos espacialmente vecinos y anteriores en el orden de decodificación. Dichas técnicas se denominan técnicas de "intrapredicción". Obsérvese que, en al menos algunos casos, la

intrapredicción solo utiliza datos de referencia de la imagen actual bajo reconstrucción y no a partir de imágenes de referencia.

Puede haber muchas formas diferentes de intrapredicción. Cuando se puede utilizar más de una de tales técnicas en una tecnología de codificación de vídeo dada, la técnica en uso se puede codificar en un modo de intrapredicción. En ciertos casos, los modos pueden tener submodos y/o parámetros, y estos pueden codificarse individualmente o incluirse en la palabra de código del modo. Qué palabra de código utilizar para una determinada combinación de modo/submodo/parámetro puede tener un impacto en la ganancia de eficiencia de codificación a través de la intrapredicción, y también puede hacerlo la tecnología de codificación de entropía utilizada para traducir las palabras de código en un flujo de bits.

Se introdujo un cierto modo de intrapredicción con H.264, se refinó en H.265, y se refinó adicionalmente en tecnologías de codificación más recientes tales como modelo de exploración conjunta (JEM, por sus siglas en inglés), codificación de vídeo versátil (VVC, por sus siglas en inglés), y conjunto de referencia (BMS, por sus siglas en inglés). Se puede formar un bloque predictor utilizando valores de muestras vecinas que pertenecen a muestras ya disponibles. Los valores de muestra de las muestras vecinas se copian en el bloque predictor según una dirección. Una referencia a la dirección en uso puede codificarse en el flujo de bits o puede predecirse por sí misma.

Con referencia a la Figura 1A, representada en la derecha inferior es un subconjunto de nueve direcciones de predictor conocidas de las 33 direcciones de predictor posibles de H.265 (correspondientes a los 33 modos angulares de los 35 intramodos). El punto donde convergen las flechas (101) representa la muestra que está siendo predicha. Las flechas representan la dirección a partir de la cual se prevé la muestra. Por ejemplo, la flecha (102) indica que la muestra (101) se predice desde una muestra o muestras hasta la derecha superior, en un ángulo de 45 grados desde la horizontal. De manera similar, la flecha (103) indica que la muestra (101) se predice desde una muestra o muestras a la izquierda inferior de la muestra (101), en un ángulo de 22,5 grados desde la horizontal.

Todavía con referencia a la Figura 1A, en la parte superior izquierda se representa un bloque cuadrado (104) de 4 x 4 muestras (indicado por una línea discontinua, en negrita). El bloque cuadrado (104) incluye 16 muestras, cada una marcada con un "S", su posición en la dimensión Y (p. ej., índice de fila) y su posición en dimensión X (p. ej., índice de columna). Por ejemplo, la muestra S21 es la segunda muestra en la dimensión Y (desde la parte superior) y la primera (de la izquierda) muestra en dimensión X. Similarmente, la muestra S44 es la cuarta muestra en el bloque (104) en las dimensiones Y y X. Como el bloque es de 4 x 4 muestras en tamaño, S44 está en la parte inferior derecha. Se muestran además muestras de referencia que siguen un esquema de numeración similar. Una muestra de referencia está marcada con una R, su posición Y (por ejemplo, índice de fila) y posición X (índice de columna) en relación con el bloque (104). Tanto en H.264 como en H.265, las muestras de predicción son vecinas al bloque bajo reconstrucción, por lo tanto, no es necesario utilizar valores negativos.

La intrapredicción de imagen puede funcionar copiando los valores de muestra de referencia de las muestras vecinas según lo apropiado por la dirección de predicción señalizada. Por ejemplo, suponer que el flujo de bits de vídeo codificado incluye señalización que, para este bloque, indica una dirección de predicción consistente con la flecha (102), es decir, las muestras se predicen a partir de una muestra o muestras de predicción a la derecha superior, en un ángulo de 45 grados desde la horizontal. En ese caso, las muestras S41, S32, S23 y S14 se predicen a partir de la misma muestra de referencia R05. Después se predice la muestra S44 a partir de la muestra de referencia R08.

En ciertos casos, los valores de múltiples muestras de referencia pueden combinarse, por ejemplo, a través de interpolación, para calcular una muestra de referencia; especialmente cuando las direcciones no son uniformemente divisibles por 45 grados.

El número de direcciones posibles ha aumentado a medida que se ha desarrollado la tecnología de codificación de vídeo. En H.264 (año 2003), podrían representarse nueve direcciones diferentes. Esto aumentó a 33 en H.265 (año 2013), y JEM/VVC/BMS, en el momento de la descripción, puede soportar hasta 65 direcciones. Se han realizado experimentos para identificar las direcciones más probables, y ciertas técnicas en la codificación por entropía se usan para representar esas direcciones probables en un pequeño número de bits, aceptando una cierta penalización para direcciones menos probables. Además, las propias direcciones a veces se pueden predecir a partir de direcciones vecinas utilizadas en bloques vecinos, ya decodificados.

La Figura 1B muestra un esquema (180) que representa 65 direcciones de intrapredicción según JEM para ilustrar el número creciente de direcciones de predicción a lo largo del tiempo.

El mapeo de bits de direcciones de intrapredicción en el flujo de bits de vídeo codificado que representa la dirección puede ser distinto de una tecnología de codificación de vídeo a otra tecnología de codificación de vídeo; y puede variar, por ejemplo, a partir de mapeos directos simples de dirección de predicción al modo de intrapredicción a palabras de código, a esquemas adaptativos complejos que implican modos más probables y técnicas similares. Sin embargo, en todos los casos puede haber ciertas direcciones que son estadísticamente menos propensas a ocurrir en contenido de vídeo que ciertas otras direcciones. Como el objetivo de la compresión de vídeo es la reducción de

redundancia, esas direcciones menos probables, en una tecnología de codificación de vídeo que trabaje bien, se representan por un mayor número de bits que direcciones más probables.

La compensación de movimiento puede ser una técnica de compresión con pérdidas y puede referirse a técnicas donde un bloque de datos de muestra a partir de una imagen previamente reconstruida o parte de la misma (imagen de referencia), después de haber sido desplazado espacialmente en una dirección indicada por un vector de movimiento (MV en adelante), se utiliza para la predicción de una imagen o parte de imagen recién reconstruida. En algunos casos, la imagen de referencia puede ser la misma que la imagen actualmente bajo reconstrucción. Los MV pueden tener dos dimensiones X e Y, o tres dimensiones, siendo la tercera una indicación de la imagen de referencia en uso (esta última, indirectamente, puede ser una dimensión temporal).

En algunas técnicas de compresión de vídeo, un MV aplicable a una cierta área de datos de muestra puede predecirse a partir de otros MV, por ejemplo, a partir de los relacionados con otra área de datos de muestra espacialmente adyacente al área bajo reconstrucción, y que precede a ese MV en orden de decodificación. Hacerlo puede reducir sustancialmente la cantidad de datos requerida para codificar el MV, eliminando de este modo la redundancia y aumentando la compresión. La predicción de MV puede funcionar eficazmente, por ejemplo, porque al codificar una señal de vídeo de entrada derivada de una cámara (conocida como vídeo natural) existe una probabilidad estadística de que áreas más grandes que el área a la que se aplica un único MV se muevan en una dirección similar y, por lo tanto, en algunos casos se puedan predecir utilizando un vector de movimiento similar derivado de los MV del área vecina. Esto da como resultado que el MV encontrado para un área dada sea similar o igual al MV previsto a partir de los MV circundantes, y eso, a su vez, puede representarse, después de la codificación por entropía, en un número de bits menor que el que se utilizaría si se codificara el MV directamente. En algunos casos, la predicción de MV puede ser un ejemplo de compresión sin pérdidas de una señal (especialmente: los MV) derivada de la señal original (especialmente: el flujo de muestras). En otros casos, la predicción de MV en sí misma puede tener pérdidas, por ejemplo, debido a errores de redondeo al calcular un predictor a partir de varios MV circundantes.

Se describen varios mecanismos de predicción de MV en H.265/HEVC (ITU-T Rec. H.265, “High Efficiency Video Coding”, diciembre de 2016). De los muchos mecanismos de predicción de MV que ofrece H.265, en la presente memoria se describe una técnica denominada en adelante “fusión espacial”.

Haciendo referencia a la Figura 2, un bloque actual (201) comprende muestras que el codificador ha descubierto durante el proceso de búsqueda de movimiento que son predecibles a partir de un bloque anterior del mismo tamaño que se ha desplazado espacialmente. En lugar de codificar ese MV directamente, el MV puede derivarse de los metadatos asociados a una o más imágenes de referencia, por ejemplo, de la imagen de referencia más reciente (en orden de decodificación), utilizando el MV asociado a una de las cinco muestras circundantes, denominadas A0, A1 y B0, B1, B2 (202 a 206, respectivamente). En H.265, la predicción de MV puede utilizar predictores de la misma imagen de referencia que está utilizando el bloque vecino.

Se conoce un método para filtrar en la decodificación de vídeo a partir de la patente US-2016/127747 A1. Un método similar es descrito en MISRA (SHARPLABS) K Y COL. en: “Cross-Component Adaptive Loop Filter for chroma”, 27. MPEG MEETING; 20190708 - 20190712; GOTHENBURG; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), n.º m48779 26 de junio de 2019 (2019-06-26), XP030222219.

La evaluación del estado de componentes cruzados para las decisiones de filtrado se conoce a partir del proceso de filtrado de desbloqueo H.264/AVC, resumido brevemente, p. ej., en las páginas 5-6 de la patente WO 2005/122588 A1 en relación con la decodificación de imágenes de vídeo codificadas por bloques.

Una función de filtro de componentes cruzados no lineal se menciona como una posibilidad en el párrafo [0077] de la descripción disponible en la patente US-2018/063527 A1 relacionada con un método de codificación de datos de vídeo realizado por un dispositivo de codificación de vídeo que genera los componentes primero y segundo de una imagen actual.

Resumen

La invención protegida se define por las características especificadas, respectivamente, en las reivindicaciones independientes adjuntas. Las reivindicaciones dependientes definen realizaciones preferidas. A continuación se facilita la descripción de la invención protegida con la descripción de las realizaciones ilustradas en la Figura 29, mientras que otros aspectos, ejemplos y realizaciones tienen fines ilustrativos y no definen el alcance de la protección.

Los aspectos de la descripción proporcionan métodos y aparatos para la codificación/decodificación de vídeo. En algunos ejemplos, un aparato para la decodificación de vídeo incluye un sistema de circuitos de procesamiento. Por ejemplo, el sistema de circuitos de procesamiento determina una primera combinación a partir de muestras reconstruidas de un primer componente de color dentro de una región de soporte de filtro. A continuación, el sistema de circuitos de procesamiento determina, basándose en un mapeo que asocia los valores de desfase con combinaciones de posibles valores de muestra reconstruida, un primer valor de desfase que está asociado con la

primera combinación, y aplica el primer valor de desfase a una muestra que se filtrará de un segundo componente de color en la región de soporte de filtro para determinar una muestra filtrada del segundo componente de color.

5 En algunas realizaciones, el mapeo es un mapeo no lineal definido en forma de tabla de búsqueda. El segundo componente de color es uno del primer componente de color o un componente de color diferente del primer componente de color.

10 En algunas realizaciones, el sistema de circuitos de procesamiento utiliza al menos una operación lineal y una operación de cuantificación para determinar la primera combinación a partir de las muestras reconstruidas del primer componente de color dentro de la región de soporte de filtro. En algunos ejemplos, el sistema de circuitos de procesamiento calcula las diferencias de las muestras reconstruidas del primer componente de color, y cuantifica las diferencias para determinar la primera combinación. Según la invención, el sistema de circuitos de procesamiento calcula las diferencias entre las muestras reconstruidas circundantes a una muestra colocada dentro de las muestras reconstruidas del primer componente de color. La muestra colocada está ubicada en la misma posición que la muestra que se filtrará del segundo componente de color. A continuación, el sistema de circuitos de procesamiento cuantifica las diferencias basándose en un tamaño del paso de cuantificación para determinar la primera combinación.

20 En una realización, el mapeo asocia los valores de desfase con las combinaciones de un dominio completo para las muestras reconstruidas del primer componente de color dentro de la región de soporte de filtro. En una realización, el mapeo asocia los valores de desfase con las combinaciones de un dominio de subconjunto para las muestras reconstruidas del primer componente de color dentro de la región de soporte de filtro.

25 En algunas realizaciones, el mapeo se realiza en al menos uno de entre un conjunto de parámetros de vídeo (VPS, en inglés), un conjunto de parámetros de secuencia (SPS, en inglés), un conjunto de parámetros de imagen (PPS, en inglés), un conjunto de parámetros de adaptación (APS, en inglés), una cabecera de segmento, y una cabecera de mosaico de un flujo de bits de vídeo codificado.

30 En una realización, el sistema de circuitos de procesamiento deriva los valores de desfase del mapeo basándose en las medidas de coste, e incluyen el mapeo en al menos uno de entre un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de adaptación (APS), una cabecera de segmento, y una cabecera de mosaico de un flujo de bits de vídeo codificado.

35 En algunas realizaciones, el sistema de circuitos de procesamiento suma el primer valor de desfase con la muestra que se filtrará del segundo componente de color para determinar una muestra filtrada intermedia, y recorta la muestra filtrada intermedia basándose en un valor máximo y un valor mínimo para determinar la muestra filtrada del segundo componente de color.

40 Los aspectos de la descripción también proporcionan un medio legible por ordenador no transitorio que almacena instrucciones que, cuando son ejecutadas por un ordenador para la decodificación de vídeo, hacen que el ordenador lleve a cabo cualquiera de los métodos de decodificación de vídeo.

Breve descripción de los dibujos

45 Características adicionales, la naturaleza y diversas ventajas del objeto descrito serán más evidentes a partir de la siguiente descripción detallada y los dibujos adjuntos en los que:

La Figura 1A es una ilustración esquemática de un subconjunto ilustrativo de modos de intrapredicción.

50 La Figura 1B es una ilustración de direcciones de intrapredicción ilustrativas.

La Figura 2 es una ilustración esquemática de un bloque actual y sus candidatos de fusión espacial circundante en un ejemplo.

55 La Figura 3 es una ilustración esquemática de un diagrama de bloques simplificado de un sistema (300) de comunicación según una realización.

La Figura 4 es una ilustración esquemática de un diagrama de bloques simplificado de un sistema (400) de comunicación según una realización.

60 La Figura 5 es una ilustración esquemática de un diagrama de bloques simplificado de un decodificador según una realización.

65 La Figura 6 es una ilustración esquemática de un diagrama de bloques simplificado de un codificador según una realización.

La Figura 7 muestra un diagrama de bloques de un codificador según otra realización.

La Figura 8 muestra un diagrama de bloques de un decodificador según otra realización.

La Figura 9 muestra ejemplos de formas de filtro según realizaciones de la descripción.

Las Figuras 10A-10D muestran ejemplos de posiciones submuestreadas utilizadas para calcular gradientes según realizaciones de la descripción.

Las Figuras 11A-11B muestran ejemplos de un proceso de filtrado de límites virtuales según realizaciones de la descripción.

Las Figuras 12A-12F muestran ejemplos de operaciones de espacios adicionales simétricos en límites virtuales según realizaciones de la descripción.

La Figura 13 muestra un ejemplo de partición de una imagen según algunas realizaciones de la descripción.

La Figura 14 muestra un patrón de separación de árbol cuaternario para una imagen en algunos ejemplos.

La Figura 15 muestra filtros de componentes cruzados según una realización de la descripción.

La Figura 16 muestra un ejemplo de una forma de filtro según una realización de la descripción.

La Figura 17 muestra un ejemplo de sintaxis para un filtro de componentes cruzados según algunas realizaciones de la descripción.

Las Figuras 18A-18B muestran ubicaciones ilustrativas de muestras de croma con respecto a muestras de luma según realizaciones de la descripción.

La Figura 19 muestra un ejemplo de búsqueda de dirección según una realización de la descripción.

La Figura 20 muestra un ejemplo que ilustra la proyección subespacial en algunos ejemplos.

La Figura 21 muestra una tabla de una pluralidad de tipos de desfase adaptativo de muestras (SAO, por sus siglas en inglés) según una realización de la descripción.

La Figura 22 muestra ejemplos de patrones para la clasificación de píxeles en el desfase de bordes en algunos ejemplos.

La Figura 23 muestra una tabla para la regla de clasificación de píxeles para el desfase de bordes en algunos ejemplos.

La Figura 24 muestra un ejemplo de sintaxis que puede señalizarse.

La Figura 25 muestra un ejemplo de un área de soporte de filtro según algunas realizaciones de la descripción.

La Figura 26 muestra un ejemplo de otra área de soporte de filtro según algunas realizaciones de la descripción.

Las Figuras 27A-27C muestran una tabla que tiene 81 combinaciones según una realización de la descripción.

La Figura 28 ilustra una tabla de búsqueda según una realización de la descripción.

La Figura 29 muestra un diagrama de flujo que describe un proceso según una realización de la descripción.

La Figura 30 es una ilustración esquemática de un sistema informático según una realización.

Descripción detallada de las realizaciones

La Figura 3 es un diagrama de bloques simplificado de un sistema de comunicación (300) según una realización de la presente descripción. El sistema de comunicación (300) incluye una pluralidad de dispositivos terminales que pueden comunicarse entre sí, a través de, por ejemplo, una red (350). Por ejemplo, el sistema de comunicación (300) incluye un primer par de dispositivos terminales (310) y (320) interconectados a través de la red (350). En el ejemplo de la Figura 3, el primer par de dispositivos terminales (310) y (320) realiza una transmisión unidireccional de datos. Por ejemplo, el dispositivo terminal (310) puede codificar datos de vídeo (por ejemplo, un flujo de imágenes de vídeo que se capturan por el dispositivo terminal (310) para transmisión al otro dispositivo terminal (320) a través de la red (350). Los datos de vídeo codificados pueden transmitirse en forma de uno o más flujos de bits de vídeo codificado. El dispositivo terminal (320) puede recibir los datos de vídeo codificados de la red (350), decodificar los datos de vídeo

codificados para recuperar las imágenes de vídeo y mostrar imágenes de vídeo según los datos de vídeo recuperados. La transmisión de datos unidireccional puede ser común en aplicaciones de servicio de medios y similares.

En otro ejemplo, el sistema de comunicación (300) incluye un segundo par de dispositivos terminales (330) y (340) que realiza la transmisión bidireccional de datos de vídeo codificados que pueden producirse, por ejemplo, durante videoconferencias. Para la transmisión bidireccional de datos, en un ejemplo, cada dispositivo terminal de los dispositivos terminales (330) y (340) puede codificar datos de vídeo (p. ej., un flujo de imágenes de vídeo capturadas por el dispositivo terminal) para su transmisión al otro dispositivo terminal de los dispositivos terminales (330) y (340) a través de la red (350). Cada dispositivo terminal de los dispositivos terminales (330) y (340) también puede recibir los datos de vídeo codificados transmitidos por el otro dispositivo terminal de los dispositivos terminales (330) y (340), y puede decodificar los datos de vídeo codificados para recuperar las imágenes de vídeo y puede mostrar imágenes de vídeo en un dispositivo de visualización accesible según los datos de vídeo recuperados.

En el ejemplo de la Figura 3, los dispositivos terminales (310), (320), (330) y (340) pueden ilustrarse como servidores, ordenadores personales y teléfonos inteligentes, pero los principios de la presente descripción pueden no estar tan limitados. Las realizaciones de la presente descripción encuentran aplicación en computadoras portátiles, tabletas, reproductores multimedia y/o equipos de videoconferencia dedicados. La red (350) representa cualquier número de redes que transmiten datos de vídeo codificados entre los dispositivos terminales (310), (320), (330) y (340), incluidas, por ejemplo, redes de comunicación alámbricas (cableadas) y/o inalámbricas. La red (350) de comunicación puede intercambiar datos en canales conmutados por circuitos y/o conmutados por paquetes. Las redes representativas incluyen redes de telecomunicaciones, redes de área local, redes de área amplia y/o Internet. Para los fines de la presente discusión, la arquitectura y topología de la red (350) pueden ser irrelevantes para el funcionamiento de la presente descripción a menos que se explique en la presente memoria a continuación.

La Figura 4 ilustra, como ejemplo para una aplicación para la materia objeto divulgada, la colocación de un codificador de vídeo y un decodificador de vídeo en un entorno de transmisión. La materia objeto descrita puede ser igualmente aplicable a otras aplicaciones habilitadas para vídeo, incluyendo, por ejemplo, videoconferencias, TV digital, almacenamiento de vídeo comprimido en medios digitales que incluyen CD, DVD, tarjeta de memoria y similares, etc.

Un sistema de transmisión puede incluir un subsistema de captura (413), que puede incluir una fuente de vídeo (401), por ejemplo una cámara digital, creando, por ejemplo, una secuencia de imágenes de vídeo (402) que no están comprimidas. En un ejemplo, el flujo de imágenes de vídeo (402) incluye muestras que son tomadas por la cámara digital. El flujo de imágenes de vídeo (402), representado como una línea en negrita para enfatizar un volumen de datos alto en comparación con los datos de vídeo codificados (404) (o flujos de bits de vídeo codificados), puede ser procesado por un dispositivo electrónico (420) que incluye un codificador de vídeo (403) acoplado a la fuente de vídeo (401). El codificador de vídeo (403) puede incluir hardware, software o una combinación de los mismos para habilitar o implementar aspectos del objeto descrito como se describe con más detalle a continuación. Los datos de vídeo codificados (404) (o flujo de bits de vídeo codificado (404)), representados como una línea delgada para enfatizar el menor volumen de datos en comparación con el flujo de imágenes de vídeo (402), se pueden almacenar en un servidor de transmisión (405), para uso futuro. Uno o más subsistemas de cliente de transmisión por secuencias, tales como los subsistemas de cliente (406) y (408) en la Figura 4 puede acceder al servidor de transmisión por secuencias (405) para recuperar copias (407) y (409) de los datos de vídeo codificados (404). Un subsistema de cliente (406) puede incluir un decodificador de vídeo (410), por ejemplo, en un dispositivo electrónico (430). El decodificador de vídeo (410) decodifica la copia entrante (407) de los datos de vídeo codificados y crea una corriente saliente de imágenes de vídeo (411) que puede representarse en una pantalla (412) (por ejemplo, pantalla de visualización) u otro dispositivo de representación (no representado). En algunos sistemas de transmisión por secuencias, los datos de vídeo codificados (404), (407) y (409) (por ejemplo, flujos de bits de vídeo) se pueden codificar según ciertos estándares de codificación/compresión de vídeo. Ejemplos de esas normas incluyen la Recomendación ITU-T H.265. En un ejemplo, una norma de codificación de vídeo en desarrollo es informalmente conocida como codificación de vídeo versátil (VVC, por sus siglas en inglés). El objeto descrito puede usarse en el contexto de VVC.

Se observa que los dispositivos electrónicos (420) y (430) pueden incluir otros componentes (no mostrados). Por ejemplo, el dispositivo electrónico (420) puede incluir un decodificador de vídeo (no mostrado) y el dispositivo electrónico (430) puede incluir también un codificador de vídeo (no mostrado).

La Figura 5 muestra un diagrama de bloques de un decodificador de vídeo (510) según una realización de la presente descripción. El decodificador de vídeo (510) puede incluirse en un dispositivo electrónico (530). El dispositivo electrónico (530) puede incluir un receptor (531) (por ejemplo, circuitos de recepción). El decodificador de vídeo (510) puede usarse en el lugar del decodificador de vídeo (410) en el ejemplo de la Figura 4.

Un receptor (531) puede recibir una o más secuencias de vídeo de códec que van a decodificarse por el decodificador de vídeo (510); en la misma o una realización, una secuencia de vídeo codificada en un momento, donde la decodificación de cada secuencia de vídeo codificada es independiente de otras secuencias de vídeo codificadas. La secuencia de vídeo codificada puede recibirse desde un canal (501), que puede ser un enlace de hardware/software a un dispositivo de almacenamiento que almacena los datos de vídeo codificados. El receptor (531) puede recibir los datos de vídeo codificados con otros datos, por ejemplo, datos de audio codificados y/o flujos de datos auxiliares, que

pueden enviarse a sus respectivas entidades de uso (no representadas). El receptor (531) puede separar la secuencia de vídeo codificada de los otros datos. Para combatir la fluctuación de fase (*jitter*) de la red, una memoria intermedia (*buffer*) (515) se puede acoplar entre el receptor (531) y el decodificador/analizador por entropía (520) ("analizador" en adelante). En ciertas aplicaciones, la memoria intermedia (515) es parte del decodificador de vídeo (510). En otras, puede estar fuera del decodificador de vídeo (510) (no representado). En otras palabras, puede haber una memoria intermedia (no representada) fuera del decodificador de vídeo (510), por ejemplo para combatir la fluctuación de fase (*jitter*) de la red y, además, otra memoria intermedia (515) dentro del decodificador de vídeo (510), por ejemplo, para manejar la temporización de la reproducción. Cuando el receptor (531) recibe datos desde un dispositivo de almacenamiento/reenvío con suficiente ancho de banda y controlabilidad, o desde una red isosincrónica, la memoria intermedia (515) puede no ser necesaria o puede ser pequeña. Para su uso en las redes de paquetes de mejor esfuerzo, como Internet, puede requerirse la memoria intermedia (515), puede ser comparativamente grande y puede ser ventajosamente de tamaño adaptativo, y puede implementarse al menos parcialmente en un sistema operativo o elementos similares (no representados) fuera del decodificador de vídeo (510).

El decodificador de vídeo (510) puede incluir un analizador (*parser*) (520) para reconstruir símbolos (521) a partir de la secuencia de vídeo codificada. Las categorías de esos símbolos incluyen información utilizada para gestionar el funcionamiento del decodificador (510) de vídeo, y potencialmente información para controlar un dispositivo de representación tal como un dispositivo (512) de representación (p. ej., una pantalla de visualización) que no es una parte integral del dispositivo electrónico (530) pero se puede acoplar al dispositivo electrónico (530), como se muestra en la Figura 5. La información de control para el (los) dispositivo(s) de representación puede estar en forma de los fragmentos de conjunto de parámetros de información de mejora suplementaria (mensajes SEI) o de información de usabilidad de vídeo (VUI) (no representados). El analizador (520) puede analizar/decodificar por entropía la secuencia de vídeo codificada que es recibida. La codificación de la secuencia de vídeo codificada puede realizarse según una tecnología o estándar de codificación de vídeo, y puede seguir varios principios, incluyendo codificación de longitud variable, codificación Huffman, codificación aritmética con o sin sensibilidad al contexto, etc. El analizador (520) puede extraer de la secuencia de vídeo codificada, un conjunto de parámetros de subgrupo para al menos uno de los subgrupos de píxeles en el decodificador de vídeo, basándose en al menos un parámetro correspondiente al grupo. Los subgrupos pueden incluir grupos de imágenes (GOP), imágenes, mosaicos, cortes, macrobloques, unidades de codificación (CU), bloques, unidades de transformada (TU), unidades de predicción (PU) y así sucesivamente. El analizador (520) también puede extraer de la secuencia de vídeo codificada información tal como coeficientes de transformación, valores de parámetros del cuantificador, vectores de movimiento, etc.

El analizador (520) puede realizar una operación de decodificación por entropía/análisis en la secuencia de vídeo recibida desde la memoria intermedia (515), para crear símbolos (521).

La reconstrucción de los símbolos (521) puede implicar múltiples unidades diferentes dependiendo del tipo de imagen de vídeo codificada o partes de la misma (tales como: inter e intraimagen, inter e intrabloque) y otros factores. Qué unidades están involucradas, y cómo, puede controlarse por la información de control de subgrupo que se analizó a partir de la secuencia de vídeo codificada por el analizador (520). El flujo de tal información de control de subgrupo entre el analizador (520) y las múltiples unidades a continuación no se representa para mayor claridad.

Más allá de los bloques funcionales ya mencionados, el decodificador de vídeo (510) puede subdividirse conceptualmente en varias unidades funcionales como se describe a continuación. En una implementación práctica que funciona bajo restricciones comerciales, muchas de estas unidades interactúan estrechamente entre sí y pueden, al menos parcialmente, integrarse entre sí. Sin embargo, con el fin de describir el objeto descrito, la subdivisión conceptual en las unidades funcionales a continuación es apropiada.

Una primera unidad es la unidad (551) de escalador/transformada inversa. La unidad de escalador/transformación inversa (551) recibe un coeficiente de transformación cuantificado, así como información de control, que incluye qué transformación utilizar, el tamaño del bloque, el factor de cuantificación, las matrices de escalado de cuantificación, etc. como símbolos (521) del analizador (520). La unidad (551) de escalador/transformada inversa puede emitir bloques que comprenden valores de muestra, que pueden introducirse en el agregador (555).

En algunos casos, las muestras de salida del escalador/transformada inversa (551) pueden pertenecer a un bloque intracodificado; esto es: un bloque que no usa información predictiva a partir de imágenes previamente reconstruidas, pero puede usar información predictiva a partir de partes previamente reconstruidas de la imagen actual. Tal información predictiva puede ser proporcionada por una unidad (552) de predicción intraimagen. En algunos casos, la unidad de predicción intraimagen (552) genera un bloque del mismo tamaño y forma que el bloque bajo reconstrucción, utilizando información circundante ya reconstruida extraída de la memoria intermedia (558) de imagen actual. La memoria intermedia (558) de imagen actual almacena, por ejemplo, imagen actual reconstruida parcialmente y/o imagen actual completamente reconstruida. El agregador (555), en algunos casos, añade, por muestra, la información de predicción, la unidad (552) de intrapredicción se ha generado a la información de la muestra de salida proporcionada por la unidad (551) de escalador/transformada inversa.

En otros casos, las muestras de salida de la unidad (551) de escalador/transformada inversa pueden pertenecer a un bloque intercodificado y potencialmente compensado en movimiento. En tal caso, una unidad de predicción de

compensación de movimiento (553) puede acceder a la memoria (557) de imágenes de referencia para extraer muestras usadas para la predicción. Después de compensar el movimiento de las muestras obtenidas según los símbolos (521) pertenecientes al bloque, estas muestras pueden añadirse por el agregador (555) a la salida de la unidad de escalador / transformada inversa (551) (en este caso llamadas muestras residuales o señal residual) para generar información de muestra de salida. Las direcciones dentro de la memoria (557) de imágenes de referencia desde donde la unidad (553) de predicción de compensación de movimiento extrae las muestras de predicción pueden controlarse mediante vectores de movimiento, disponibles para la unidad (553) de predicción de compensación de movimiento en forma de símbolos (521) que pueden tener, por ejemplo, X, Y, y componentes de imagen de referencia. La compensación de movimiento también puede incluir la interpolación de valores de muestra extraídos de la memoria (557) de imágenes de referencia cuando se utilizan vectores de movimiento exactos de submuestra, mecanismos de predicción de vectores de movimiento, etc.

Las muestras de salida del agregador (555) pueden someterse a diversas técnicas de filtrado en bucle en la unidad (556) de filtro en bucle. Las tecnologías de compresión de vídeo pueden incluir tecnologías de filtro en bucle que están controladas por parámetros incluidos en la secuencia de vídeo codificado (también denominada flujo de bits de vídeo codificado) y puestos a disposición de la unidad de filtro de bucle (556) como símbolos (521) del analizador (520), pero también puede responder a la metainformación obtenida durante la decodificación de partes anteriores (en orden de decodificación) de la imagen codificada o secuencia de vídeo codificada, así como responder a valores de muestra previamente reconstruidos y filtrados en bucle.

La salida de la unidad (556) de filtro en bucle puede ser un flujo de muestra que se puede emitir al dispositivo (512) de representación así como almacenarse en la memoria (557) de imágenes de referencia para su uso en futuras predicciones entre imágenes.

Ciertas imágenes codificadas, una vez completamente reconstruidas, pueden usarse como imágenes de referencia para la predicción futura. Por ejemplo, una vez que una imagen codificada correspondiente a una imagen actual se reconstruye completamente y la imagen codificada ha sido identificada como una imagen de referencia (mediante, por ejemplo, el analizador (520)), la memoria intermedia (558) de imagen actual puede convertirse en parte de la memoria (557) de imágenes de referencia, y se puede reasignar una memoria intermedia de imágenes actual nueva antes de comenzar la reconstrucción de la siguiente imagen codificada.

El decodificador de vídeo (510) puede realizar operaciones de decodificación según una tecnología de compresión de vídeo predeterminada en un estándar, tal como ITU-T Rec. H.265. La secuencia de vídeo codificada puede ajustarse a una sintaxis especificada por la tecnología de compresión de vídeo o estándar que se utiliza, en el sentido de que la secuencia de vídeo codificada se adhiere tanto a la sintaxis de la tecnología o estándar de compresión de vídeo como a los perfiles según lo documentado en la tecnología o estándar de compresión de vídeo. Específicamente, un perfil puede seleccionar ciertas herramientas como las únicas herramientas disponibles para utilizarse bajo ese perfil de todas las herramientas disponibles en la tecnología o estándar de compresión de vídeo. También es necesario para el cumplimiento que la complejidad de la secuencia de vídeo codificada esté dentro de límites como se define por el nivel de la tecnología o estándar de compresión de vídeo. En algunos casos, los niveles restringen el tamaño máximo de la imagen, la tasa máxima de tramas, la tasa máxima de la muestra de reconstrucción (medida, por ejemplo, en megamuestras por segundo), el tamaño máximo de la imagen de referencia, y así sucesivamente. Los límites establecidos por los niveles pueden, en algunos casos, restringirse adicionalmente a través de especificaciones de Decodificador de Referencia Hipotética (HRD) y metadatos para la gestión de memoria intermedia HRD señaladas en la secuencia de vídeo codificada.

En una realización, el receptor (531) puede recibir datos adicionales (redundantes) con el vídeo codificado. Los datos adicionales pueden incluirse como parte de la(s) secuencia(s) de vídeo codificada(s). El decodificador (510) de vídeo puede utilizar los datos adicionales para decodificar correctamente los datos y/o para reconstruir con mayor precisión los datos de vídeo originales. Los datos adicionales pueden estar en forma de, por ejemplo, capas de mejora temporal, espacial o relación señal a ruido (SNR), segmentos redundantes, imágenes redundantes, códigos de corrección de errores directos, etc.

La Figura 6 muestra un diagrama de bloques de un codificador de vídeo (603) según una realización de la presente descripción. El codificador de vídeo (603) se incluye en un dispositivo electrónico (620). El dispositivo electrónico (620) incluye un transmisor (640) (por ejemplo, circuitos de transmisión). El codificador de vídeo (603) puede usarse en el lugar del codificador de vídeo (403) en el ejemplo de la Figura 4.

El codificador (603) de vídeo puede recibir muestras de vídeo de una fuente (601) de vídeo (que no es parte del dispositivo electrónico [620] en el ejemplo de la Figura 6) que puede capturar imágenes de vídeo para ser codificadas por el codificador (603) de vídeo. En otro ejemplo, la fuente de vídeo (601) es una parte del dispositivo electrónico (620).

La fuente de vídeo (601) puede proporcionar la secuencia de vídeo de origen a codificar por el codificador (603) en forma de una corriente de muestra de vídeo digital que puede ser de cualquier profundidad de bits adecuada (por ejemplo: 8 bit, 10 bit, 12 bit, ...), cualquier espacio de color (por ejemplo, BT.601 Y CrCb, RGB, ...), y cualquier

estructura de muestreo adecuada (por ejemplo, Y CrCb 4:2:0, Y CrCb 4:4:4). En un sistema de servicio de medios, la fuente (601) de vídeo puede ser un dispositivo de almacenamiento que almacena un vídeo previamente preparado. En un sistema de videoconferencias, la fuente (601) de vídeo puede ser una cámara que captura información de imagen local como una secuencia de vídeo. Los datos de vídeo pueden proporcionarse como una pluralidad de imágenes individuales que imparten movimiento cuando se ven en secuencia. Las propias imágenes pueden organizarse como una matriz espacial de píxeles, en donde cada píxel puede comprender una o más muestras dependiendo de la estructura de muestreo, espacio de color, etc., en uso. Un experto en la técnica puede comprender fácilmente la relación entre píxeles y muestras. La siguiente descripción se centra en las muestras.

Según una realización, el codificador de vídeo (603) puede codificar y comprimir las imágenes de la secuencia de vídeo fuente en una secuencia de vídeo codificada (643) en tiempo real o en cualquier otra limitación de tiempo según lo requiera la aplicación. Hacer cumplir la velocidad de codificación adecuada es una función de un controlador (650). En algunas realizaciones, el controlador (650) controla otras unidades funcionales como se describe a continuación y se acopla funcionalmente a las otras unidades funcionales. El acoplamiento no se representa para mayor claridad. Los parámetros establecidos por el controlador (650) pueden incluir parámetros relacionados con el control de velocidad (salto de imagen, cuantificador, valor lambda de técnicas de optimización de distorsión de velocidad,...), tamaño de imagen, diseño de grupo de imágenes (GOP), intervalo máximo de búsqueda de vector de movimiento, etc. El controlador (650) puede configurarse para tener otras funciones adecuadas que pertenecen al codificador de vídeo (603) optimizado para un cierto diseño del sistema.

En algunas realizaciones, el codificador de vídeo (603) está configurado para operar en un bucle de codificación. Como una descripción más simplificada, en un ejemplo, el bucle de codificación puede incluir un codificador fuente (630) (por ejemplo, responsable de crear símbolos, tal como un flujo de símbolos, basándose en una imagen de entrada a codificar, y una(s) imagen(es) de referencia) y un decodificador (633) incorporado en el codificador (603) de vídeo. El decodificador (633) reconstruye los símbolos para crear los datos de muestra de una manera similar a la de un decodificador (remoto) también crearía (ya que cualquier compresión entre símbolos y flujo de bits de vídeo codificado es sin pérdidas en las tecnologías de compresión de vídeo consideradas en la materia objeto divulgada). El flujo de muestra reconstruido (datos de muestra) se introduce en la memoria (634) de imágenes de referencia. Como la decodificación de un flujo de símbolos conduce a resultados exactos en bits independientemente de la ubicación del decodificador (local o remoto), el contenido en la memoria (634) de imágenes de referencia también es exacto en bits entre el codificador local y el codificador remoto. En otras palabras, la parte de predicción de un codificador “ve” como muestras de imagen de referencia exactamente los mismos valores de muestra que un decodificador “ve” cuando se usa predicción durante la decodificación. Este principio fundamental de sincronización de la imagen de referencia (y la deriva resultante, si no se puede mantener la sincronización, por ejemplo debido a errores de canal) también se utiliza en algunas técnicas relacionadas.

El funcionamiento del decodificador “local” (633) puede ser el mismo que el de un decodificador “remoto”, tal como el decodificador de vídeo (510), que ya se ha descrito en detalle anteriormente junto con la Figura 5. Haciendo referencia brevemente también a la Figura 5, sin embargo, como los símbolos están disponibles y la codificación/decodificación de símbolos en una secuencia de vídeo codificada mediante un codificador por entropía (645) y el analizador (520) puede realizarse sin pérdidas, las partes de decodificación de entropía del decodificador de vídeo (510), incluyendo la memoria intermedia (515) y el analizador (520) pueden no estar completamente implementados en el decodificador local (633).

Una observación que puede realizarse en este punto es que cualquier tecnología de decodificador excepto la decodificación de análisis/entropía que está presente en un decodificador también necesariamente necesita estar presente, en forma funcional sustancialmente idéntica, en un codificador correspondiente. Por esta razón, la materia objeto divulgada se enfoca en la operación del decodificador. La descripción de las tecnologías de codificador puede abreviarse como la inversa de las tecnologías de decodificador descritas ampliamente. Solo en ciertas áreas se requiere una descripción más detallada y se proporciona a continuación.

Durante la operación, en algunos ejemplos, el codificador fuente (630) puede realizar codificación predictiva compensada por movimiento, que codifica una imagen de entrada de manera predictiva con referencia a una o más imágenes previamente codificadas de la secuencia de vídeo que fueron designadas como “imágenes de referencia”. De esta manera, el motor de codificación (632) codifica diferencias entre bloques de píxeles de una imagen de entrada y bloques de píxeles de imágenes de referencia que pueden seleccionarse como referencias de predicción para la imagen de entrada.

El decodificador de vídeo local (633) puede decodificar datos de vídeo codificados de imágenes que pueden designarse como imágenes de referencia, basándose en símbolos creados por el codificador fuente (630). Las operaciones del motor (632) de codificación pueden ser ventajosamente procesos con pérdidas. Cuando los datos de vídeo codificados pueden decodificarse en un decodificador de vídeo (no mostrado en la Figura 6), la secuencia de vídeo reconstruida típicamente puede ser una réplica de la secuencia de vídeo de origen con algunos errores. El decodificador de vídeo local (633) replica procesos de decodificación que puede realizar el decodificador de vídeo en imágenes de referencia y puede hacer que las imágenes de referencia reconstruidas se almacenen en la memoria caché de imágenes de referencia (634). De esta manera, el codificador de vídeo (603) puede almacenar copias de

imágenes de referencia reconstruidas localmente que tienen contenido común como las imágenes de referencia reconstruidas que serán obtenidas por un decodificador de vídeo del extremo lejano (sin errores de transmisión).

El predictor (635) puede realizar búsquedas de predicción para el motor (632) de codificación. Es decir, para codificar una nueva imagen, el predictor (635) puede buscar en la memoria (634) de imágenes de referencia datos de muestra (como bloques de píxeles de referencia candidatos) o ciertos metadatos tales como vectores de movimiento de imágenes de referencia, formas de bloques, etc. adelante, que puede servir como referencia de predicción adecuada para las nuevas imágenes. El predictor (635) puede operar sobre una base bloque de muestra por píxel para encontrar referencias de predicción apropiadas. En algunos casos, según lo determinado por los resultados de búsqueda obtenidos por el predictor (635), una imagen de entrada puede tener referencias de predicción extraídas de múltiples imágenes de referencia almacenadas en la memoria (634) de imágenes de referencia.

El controlador (650) puede gestionar operaciones de codificación del codificador fuente (630), incluyendo, por ejemplo, la configuración de parámetros y parámetros de subgrupo usados para codificar los datos de vídeo.

La salida de todas las unidades funcionales mencionadas anteriormente puede someterse a codificación por entropía en el codificador (645) por entropía. El codificador (645) por entropía traduce los símbolos generados por las diversas unidades funcionales en una secuencia de vídeo codificada, comprimiendo sin pérdidas los símbolos según tecnologías conocidas tales como codificación de Huffman, codificación de longitud variable, codificación aritmética, y etc.

El transmisor (640) puede almacenar en memoria intermedia las secuencias de vídeo codificadas creadas por el codificador por entropía (645) para preparar la transmisión a través de un canal de comunicación (660), que puede ser un enlace de hardware/software a un dispositivo de almacenamiento que almacenaría los datos de vídeo codificados. El transmisor (640) puede fusionar datos de vídeo codificados del codificador (603) de vídeo con otros datos que se transmitirán, por ejemplo, datos de audio codificados y/o flujos de datos auxiliares (fuentes no mostradas).

El controlador (650) puede gestionar el funcionamiento del codificador de vídeo (603). Durante la codificación, el controlador (650) puede asignar a cada imagen codificada un cierto tipo de imagen codificada, lo que puede afectar a las técnicas de codificación que pueden aplicarse a la imagen respectiva. Por ejemplo, las imágenes a menudo pueden asignarse como uno de los siguientes tipos de imagen:

Una intraimagen (imagen I) puede ser una que pueda codificarse y decodificarse sin usar ninguna otra imagen en la secuencia como una fuente de predicción. Algunos códecs de vídeo permiten diferentes tipos de imágenes intra, incluidas, por ejemplo, imágenes de actualización de decodificador independiente ("IDR", por sus siglas en inglés). Un experto en la técnica conoce esas variantes de imágenes I y sus respectivas aplicaciones y características.

Una imagen predictiva (imagen P) puede ser una que puede codificarse y decodificarse utilizando intrapredicción o interpredicción utilizando como máximo un vector de movimiento e índice de referencia para predecir los valores de muestra de cada bloque.

Una imagen predictiva bidireccional (imagen B) puede ser una que puede codificarse y decodificarse utilizando intrapredicción o interpredicción utilizando como máximo dos vectores de movimiento e índices de referencia para predecir los valores de muestra de cada bloque. De manera similar, las imágenes multipredictivas pueden usar más de dos imágenes de referencia y metadatos asociados para la reconstrucción de un solo bloque.

Las imágenes de origen comúnmente pueden subdividirse espacialmente en una pluralidad de bloques de muestra (por ejemplo, bloques de 4x4, 8x8, 4x8, o 16x16 muestras cada una) y codificarse en una base bloque por bloque. Los bloques pueden codificarse de manera predictiva con referencia a otros bloques (ya codificados) según lo determinado por la asignación de codificación aplicada a las imágenes respectivas de los bloques. Por ejemplo, los bloques de imágenes I pueden codificarse de forma no predictiva o pueden codificarse de manera predictiva con referencia a bloques ya codificados de la misma imagen (predicción espacial o intrapredicción). Los bloques de píxeles de imágenes P pueden codificarse de forma predictiva, mediante predicción espacial o predicción temporal con respecto a una imagen de referencia previamente codificada. Los bloques de imágenes B pueden codificarse de forma predictiva, mediante predicción espacial o mediante predicción temporal con referencia a una o dos imágenes de referencia previamente codificadas.

El codificador de vídeo (603) puede realizar operaciones de codificación según una tecnología o estándar de codificación de vídeo predeterminado, tal como ITU-T Rec. H.265. En su funcionamiento, el codificador de vídeo (603) puede realizar varias operaciones de compresión, incluidas operaciones de codificación predictiva que explotan redundancias temporales y espaciales en la secuencia de vídeo de entrada. Los datos de vídeo codificados, por lo tanto, pueden ajustarse a una sintaxis especificada por la tecnología de codificación de vídeo o estándar que se utiliza.

En una realización, el transmisor (640) puede transmitir datos adicionales con el vídeo codificado. El codificador fuente (630) puede incluir dichos datos como parte de la secuencia de vídeo codificada. Los datos adicionales pueden

comprender capas de mejora temporal/espaciales/SNR, otras formas de datos redundantes tales como imágenes redundantes y segmentos, SEI, fragmentos de conjunto de parámetros de VUI, etc.

Un vídeo puede capturarse como una pluralidad de imágenes fuente (imágenes de vídeo) en una secuencia temporal. La predicción intraimagen (frecuentemente abreviada como intrapredicción) utiliza correlación espacial en una imagen dada, y la predicción entre imágenes utiliza correlación (temporal u otra) entre las imágenes. En un ejemplo, una imagen específica bajo codificación/decodificación, que se denomina imagen actual, se divide en bloques. Cuando un bloque en la imagen actual es similar a un bloque de referencia en una imagen de referencia previamente codificada y aún almacenada en el vídeo, el bloque en la imagen actual puede codificarse mediante un vector al que se hace referencia como vector de movimiento. El vector de movimiento apunta al bloque de referencia en la imagen de referencia, y puede tener una tercera dimensión que identifica la imagen de referencia, en caso de que se usen múltiples imágenes de referencia.

En algunas realizaciones, se puede usar una técnica de bipredicción en la predicción entre imágenes. Según la técnica de bipredicción, se utilizan dos imágenes de referencia, tales como una primera imagen de referencia y una segunda imagen de referencia que son ambas antes en orden de decodificación a la imagen actual en el vídeo (pero pueden estar en el pasado y en el futuro, respectivamente, en orden de visualización). Un bloque en la imagen actual puede codificarse mediante un primer vector de movimiento que apunta a un primer bloque de referencia en la primera imagen de referencia, y un segundo vector de movimiento que apunta a un segundo bloque de referencia en la segunda imagen de referencia. El bloque puede predecirse mediante una combinación del primer bloque de referencia y el segundo bloque de referencia.

Además, se puede usar una técnica de modo de fusión en la predicción entre imágenes para mejorar la eficiencia de codificación.

Según algunas realizaciones de la descripción, las predicciones, tales como predicciones interimagen e predicciones intraimagen se realizan en la unidad de bloques. Por ejemplo, según el estándar HEVC, una imagen en una secuencia de imágenes de vídeo se divide en unidades de árbol de codificación (CTU) para compresión, las CTU en una imagen tienen el mismo tamaño, tal como 64 x 64 píxeles, 32 x 32 píxeles, o 16 x 16 píxeles. En general, una CTU incluye tres bloques de árbol de codificación (CTB), que son un CTB de luma y dos CTB de croma. Cada CTU se puede dividir de forma recursiva en cuatro árboles en una o varias unidades de codificación (CU). Por ejemplo, una CTU de 64x64 píxeles se puede separar en una CU de 64x64 píxeles, o 4 CU de 32x32 píxeles, o 16 CU de 16x16 píxeles. En un ejemplo, cada CU se analiza para determinar un tipo de predicción para la CU, tal como un tipo de interpredicción o un tipo de intrapredicción. La CU se divide en una o más unidades de predicción (PU) dependiendo de la previsibilidad temporal y/o espacial. En general, cada PU incluye un bloque de predicción de luma (PB), y dos PBs de croma. En una realización, una operación de predicción en codificación (codificación/decodificación) se realiza en la unidad de un bloque de predicción. Utilizando un bloque de predicción de luma como un ejemplo de un bloque de predicción, el bloque de predicción incluye una matriz de valores (p. ej., valores de luma) para píxeles, tales como 8x8 píxeles, 16x16 píxeles, 8x16 píxeles, 16x8 píxeles, y similares.

La Figura 7 muestra un diagrama de un codificador de vídeo (703) según otra realización de la descripción. El codificador de vídeo (703) está configurado para recibir un bloque de procesamiento (por ejemplo, un bloque de predicción) de valores de muestra dentro de una imagen de vídeo actual en una secuencia de imágenes de vídeo, y codificar el bloque de procesamiento en una imagen codificada que es parte de una secuencia de vídeo codificada. En un ejemplo, el codificador de vídeo (703) se usa en el lugar del codificador de vídeo (403) en el ejemplo de la Figura 4.

En un ejemplo de HEVC, el codificador de vídeo (703) recibe una matriz de valores de muestra para un bloque de procesamiento, tal como un bloque de predicción de 8 x 8 muestras, y similares. El codificador de vídeo (703) determina si el bloque de procesamiento se codifica mejor usando modo de intramodo, inter modo o de bipredicción usando, por ejemplo, optimización de tasa-distorsión. Cuando el bloque de procesamiento debe codificarse en modo intra, el codificador de vídeo (703) puede usar una técnica de intrapredicción para codificar el bloque de procesamiento en la imagen codificada; y cuando el bloque de procesamiento se va a codificar en modo inter o de predicción bidireccional, el codificador de vídeo (703) puede usar una técnica de interpredicción o de predicción bidireccional, respectivamente, para codificar el bloque de procesamiento en la imagen codificada. En ciertas tecnologías de codificación de vídeo, el modo de fusión puede ser un submodo de predicción entre imágenes donde el vector de movimiento se deriva de uno o más predictores del vector de movimiento sin el beneficio de un componente de vector de movimiento codificado fuera de los predictores. En ciertas otras tecnologías de codificación de vídeo, puede estar presente un componente de vector de movimiento aplicable al bloque de objeto. En un ejemplo, el codificador de vídeo (703) incluye otros componentes, tales como un módulo de decisión de modo (no mostrado) para determinar el modo de los bloques de procesamiento.

En el ejemplo de la Figura 7, el codificador (703) de vídeo incluye el intercodificador (730), un intracodificador (722), un calculador (723) de residuos, un conmutador (726), un codificador (724) de residuos, un controlador general (721), y un codificador (725) por entropía acoplados juntos como se muestra en la Figura 7.

El intercodificador (730) está configurado para recibir las muestras del bloque actual (por ejemplo, un bloque de procesamiento), comparar el bloque con uno o más bloques de referencia en imágenes de referencia (por ejemplo, bloques en imágenes anteriores e imágenes posteriores), generar información de interpredicción (por ejemplo, descripción de información redundante según la técnica de intercodificación, vectores de movimiento, información de modo de fusión) y calcular resultados de interpredicción (por ejemplo, bloque predicho) basándose en la información de interpredicción usando cualquier técnica adecuada. En algunos ejemplos, las imágenes de referencia son imágenes de referencia decodificadas que se descodifican basándose en la información de vídeo codificada.

El intracodificador (722) está configurado para recibir las muestras del bloque actual (p. ej., un bloque de procesamiento), en algunos casos comparar el bloque con bloques ya codificados en la misma imagen, generar coeficientes cuantificados después de la transformación y, en algunos casos, también información de intrapredicción (p. ej., una información de dirección de intrapredicción según una o más técnicas de codificación intra). En un ejemplo, el intracodificador (722) también calcula resultados de intrapredicción (por ejemplo, bloque predicho) basándose en la información de intrapredicción y bloques de referencia en la misma imagen.

El controlador general (721) está configurado para determinar datos de control generales y controlar otros componentes del codificador de vídeo (703) en base a los datos de control generales. En un ejemplo, el controlador general (721) determina el modo del bloque y proporciona una señal de control al conmutador (726) basándose en el modo. Por ejemplo, cuando el modo es el modo intra, el controlador general (721) controla el conmutador (726) para seleccionar el resultado de modo intra para ser utilizado por la calculadora (723) de residuo, y controla el codificador (725) por entropía para seleccionar la información de intrapredicción e incluir la información de intrapredicción en el flujo de bits; y cuando el modo es el modo inter, el controlador general (721) controla el interruptor (726) para seleccionar el resultado de interpredicción para su uso por el calculador de residuos (723), y controla el codificador por entropía (725) para seleccionar la información de interpredicción e incluir la información de interpredicción en el flujo de bits.

El calculador de residuos (723) está configurado para calcular una diferencia (datos de residuos) entre el bloque recibido y los resultados de predicción seleccionados del intracodificador (722) o del intercodificador (730). El codificador de residuos (724) está configurado para operar basándose en los datos de residuo para codificar los datos de residuo para generar los coeficientes de transformada. En un ejemplo, el codificador (724) de residuo está configurado para convertir los datos de residuo desde un dominio espacial a un dominio de frecuencia, y generar los coeficientes de transformada. Los coeficientes de transformada se someten entonces a un procesamiento de cuantificación para obtener coeficientes de transformada cuantificados. En diversas realizaciones, el codificador de vídeo (703) también incluye un decodificador de residuos (728). El decodificador de residuo (728) está configurado para realizar la transformada inversa, y generar los datos de residuos decodificados. Los datos de residuos decodificados pueden usarse adecuadamente por el intracodificador (722) y el intercodificador (730). Por ejemplo, el intercodificador (730) puede generar bloques decodificados basándose en los datos de residuos decodificados e información de interpredicción, y el intracodificador (722) puede generar bloques decodificados basándose en los datos de residuos decodificados y la información de intrapredicción. Los bloques decodificados se procesan adecuadamente para generar imágenes decodificadas y las imágenes decodificadas pueden almacenarse temporalmente en un circuito de memoria (no mostrado) y usarse como imágenes de referencia en algunos ejemplos.

El codificador por entropía (725) está configurado para formatear el flujo de bits para incluir el bloque codificado. El codificador (725) por entropía está configurado para incluir diversa información según un estándar adecuado, tal como el estándar HEVC. En un ejemplo, el codificador por entropía (725) está configurado para incluir los datos de control generales, la información de predicción seleccionada (por ejemplo, información de intrapredicción o información de interpredicción), la información de residuo y otra información adecuada en el flujo de bits. Obsérvese que, según el objeto descrito, cuando se codifica un bloque en el submodo de fusión de modo inter o de predicción bidireccional, no hay información de residuo.

La Figura 8 muestra un diagrama de un decodificador de vídeo (810) según otra realización de la descripción. El decodificador (810) de vídeo está configurado para recibir imágenes codificadas que forman parte de una secuencia de vídeo codificada, y decodificar las imágenes codificadas para generar imágenes reconstruidas. En un ejemplo, el decodificador de vídeo (810) se usa en el lugar del decodificador de vídeo (410) en el ejemplo de la Figura 4.

En el ejemplo de la Figura 8, el decodificador de vídeo (810) incluye un decodificador por entropía (871), un interdecodificador (880), un decodificador de residuo (873), un módulo de reconstrucción (874) y un intradecodificador (872) acoplado como se muestra en la Figura 8.

El decodificador por entropía (871) puede configurarse para reconstruir, a partir de la imagen codificada, ciertos símbolos que representan los elementos de sintaxis de los cuales la imagen codificada está formada. Tales símbolos pueden incluir, por ejemplo, el modo en el que se codifica un bloque (tal como, por ejemplo, modo intra, modo inter, modo bipredicho, estos dos últimos en el submodo de fusión u otro submodo), información de predicción (tal como, por ejemplo, información de intrapredicción o información de interpredicción) que puede identificar ciertas muestras o metadatos que se utilizan para la predicción por el intradecodificador (872) o el interdecodificador (880), respectivamente, información residual en forma de, por ejemplo, coeficientes de transformada cuantificados, y

similares. En un ejemplo, cuando el modo de predicción es inter o modo de bipredicción, la información de interpredicción se proporciona al interdecodificador (880); y cuando el tipo de predicción es el tipo de intrapredicción, la información de intrapredicción se proporciona al intradecodificador (872). La información residual puede someterse a cuantificación inversa y se proporciona al decodificador de residuo (873).

El interdecodificador (880) está configurado para recibir la información de interpredicción, y generar resultados de interpredicción basándose en la información de interpredicción.

El intradecodificador (872) está configurado para recibir la información de intrapredicción, y generar resultados de predicción basándose en la información de intrapredicción.

El decodificador de residuos (873) está configurado para realizar una cuantificación inversa para extraer coeficientes de transformada descuantificados, y procesar los coeficientes de transformada descuantificados para convertir el residuo del dominio de frecuencia al dominio espacial. El decodificador (873) de residuo también puede requerir cierta información de control (para incluir el parámetro de cuantificación [QP, inglés]), y esa información puede ser proporcionada por el decodificador (871) por entropía (la trayectoria de datos no representada como puede ser únicamente información de control de volumen bajo).

El módulo de reconstrucción (874) está configurado para combinar, en el dominio espacial, el residuo como salida por el decodificador de residuo (873) y los resultados de predicción (como salida por los módulos inter o intrapredicción como el caso puede ser) para formar un bloque reconstruido, que puede ser parte de la imagen reconstruida, que a su vez puede ser parte del vídeo reconstruido. Se observa que otras operaciones adecuadas, tales como una operación de desbloqueo y similares, pueden realizarse para mejorar la calidad visual.

Se observa que los codificadores (403), (603) y (703) de vídeo, y los decodificadores (410), (510) y (810) de vídeo pueden implementarse utilizando cualquier técnica adecuada. En una realización, los codificadores (403), (603) y (703) de vídeo, y los decodificadores (410), (510) y (810) de vídeo pueden implementarse utilizando uno o más circuitos integrados. En otra realización, los codificadores (403), (603) y (603) de vídeo, y los decodificadores (410), (510) y (810) de vídeo pueden implementarse utilizando uno o más procesadores que ejecutan instrucciones de software.

Los aspectos de la descripción proporcionan técnicas de filtrado para la codificación/decodificación de vídeo.

Los codificadores/decodificadores pueden aplicar un filtro en bucle adaptativo (ALF, por sus siglas en inglés) con una adaptación de filtro basada en bloques para reducir los artefactos. Para un componente de luma, se puede seleccionar uno de una pluralidad de filtros (p. ej., 25 filtros) para un bloque de luma de 4x4, por ejemplo, basándose en la dirección y la actividad de los gradientes locales.

Un ALF puede tener cualquier forma y tamaño adecuados. Haciendo referencia a la Figura 9, los ALF (910)-(911) tienen una forma de diamante, tal como una forma de diamante de 5x5 para el ALF (910) y una forma de diamante de 7x7 para el ALF (911). En el ALF (910), los elementos (920)-(932) tienen forma de diamante y se pueden utilizar en el proceso de filtrado. Se pueden utilizar siete valores (p. ej., C0-C6) para los elementos (920)-(932). En el ALF (911), los elementos (940)-(964) tienen forma de diamante y pueden utilizarse en el proceso de filtrado. Se pueden utilizar trece valores (p. ej., C0-C12) para los elementos (940)-(964).

Haciendo referencia a la Figura 9, en algunos ejemplos, se utilizan los dos ALF (910)-(911) con forma de filtro de diamante. El filtro (910) con forma de diamante de 5x5 se puede aplicar para componentes de croma (p. ej., bloques de croma, CB de croma), y el filtro (911) con forma de diamante de 7x7 se puede aplicar para un componente de luma (p. ej., un bloque de luma, un CB de luma). Se pueden utilizar otras formas y tamaños adecuados en el ALF. Por ejemplo, se puede utilizar un filtro con forma de diamante de 9x9.

Los coeficientes de filtro en las ubicaciones indicadas por los valores (p. ej., C0-C6 en (910) o C0-C12 en (920)) pueden ser distintos de cero. Además, cuando el ALF incluye una función de recorte, los valores de recorte en las ubicaciones pueden ser distintos de cero.

Para la clasificación por bloques de un componente de luma, un bloque de 4x4 (o bloque de luma, CB de luma) puede categorizarse o clasificarse como una de las múltiples clases (p. ej., 25). Se puede obtener un índice de clasificación C basándose en un parámetro de direccionalidad D y un valor cuantificado $\hat{A}$ de un valor de actividad A utilizando la Ec. (1).

$$C = 5D + \hat{A} \quad \text{Ec. (1)}$$

Para calcular el parámetro de direccionalidad D y el valor cuantificado $\hat{A}$, los gradientes g_v , g_h , g_{d1} , y g_{d2} de una dirección vertical, una horizontal, y dos diagonales (p. ej., d1 y d2), respectivamente, se pueden calcular utilizando el laplaciano 1-D de la siguiente manera.

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)| \quad \text{Ec. (2)}$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)| \quad \text{Ec. (3)}$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad \text{Ec. (4)}$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad \text{Ec. (5)}$$

donde los índices i y j se refieren a las coordenadas de una muestra superior izquierda dentro del bloque 4×4 y $R(k, l)$ indica una muestra reconstruida en una coordenada (k, l) . Las direcciones (p. ej., $d1$ y $d2$) pueden referirse a 2 direcciones diagonales.

Para reducir la complejidad de la clasificación de bloques descrita anteriormente, se puede aplicar un cálculo laplaciano 1-D submuestreado. Las Figuras 10A-10D muestran ejemplos de posiciones submuestreadas utilizadas para calcular los gradientes g_v , g_h , g_{d1} , y g_{d2} de la dirección vertical (Figura 10A), la horizontal (Figura 10B) y las dos direcciones diagonales $d1$ (Figura 10C) y $d2$ (Figura 10D), respectivamente. Se pueden utilizar las mismas posiciones submuestreadas para el cálculo del gradiente de las diferentes direcciones. En la Figura 10A, las etiquetas 'V' muestran las posiciones submuestreadas para calcular el gradiente vertical g_v . En la Figura 10B, las etiquetas 'H' muestran las posiciones submuestreadas para calcular el gradiente horizontal g_h . En la Figura 10C, las etiquetas 'D1' muestran las posiciones submuestreadas para calcular el gradiente diagonal $d1$ g_{d1} . En la Figura 10D, las etiquetas 'D2' muestran las posiciones submuestreadas para calcular el gradiente diagonal $d2$ g_{d2} .

Un valor máximo $g_{h,v}^{\max}$ y un valor mínimo $g_{h,v}^{\min}$ de los gradientes de las direcciones horizontal y vertical g_v y g_h se pueden establecer de la siguiente manera:

$$g_{h,v}^{\max} = \max(g_h, g_v), \quad g_{h,v}^{\min} = \min(g_h, g_v) \quad \text{Ec. (6)}$$

Un valor máximo $g_{d1,d2}^{\max}$ y un valor mínimo $g_{d1,d2}^{\min}$ de los gradientes de dos direcciones diagonales g_{d1} y g_{d2} se pueden establecer como:

$$g_{d1,d2}^{\max} = \max(g_{d1}, g_{d2}), \quad g_{d1,d2}^{\min} = \min(g_{d1}, g_{d2}) \quad \text{Ec. (7)}$$

El parámetro de direccionalidad D se puede derivar basándose en los valores anteriores y de los dos umbrales t_1 y t_2 , como se indica a continuación.

Paso 1. Si (1) $g_{h,v}^{\max} \leq t_1 \cdot g_{h,v}^{\min}$ y (2) $g_{d1,d2}^{\max} \leq t_1 \cdot g_{d1,d2}^{\min}$ son verdaderos, D se establece en 0.

Paso 2. Si $g_{h,v}^{\max} / g_{h,v}^{\min} > g_{d1,d2}^{\max} / g_{d1,d2}^{\min}$, continúe con el paso 3; de lo contrario, continúe con el Paso 4.

Paso 3. Si $g_{h,v}^{\max} > t_2 \cdot g_{h,v}^{\min}$, D se establece en 2; de lo contrario, D se establece en 1.

Paso 4. Si $g_{d1,d2}^{\max} > t_2 \cdot g_{d1,d2}^{\min}$, D se establece en 4; de otro modo, D se establece a 3.

El valor de actividad A se puede calcular como:

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}) = g_v + g_h \quad \text{Ec. (8)}$$

A se puede cuantificar adicionalmente en un rango de 0 a 4, inclusive, y el valor cuantificado se indica como $\hat{A}$.

Para los componentes de croma en una imagen, no se aplica ninguna clasificación de bloques y, por lo tanto, se puede aplicar un único conjunto de coeficientes de ALF para cada componente de croma.

Las transformaciones geométricas se pueden aplicar a los coeficientes de filtro y a los valores de recorte de filtro correspondientes (también denominados valores de recorte). Antes de filtrar un bloque (p. ej., un bloque de luma 4x4), se pueden aplicar transformaciones geométricas tales como la rotación o el intercambio diagonal y vertical a los coeficientes de filtro $f(k, l)$ y a los valores de recorte de filtro correspondientes $c(k, l)$, por ejemplo, dependiendo de los valores de gradiente (p. ej., g_v, g_h, g_{d1} , y/o g_{d2}) calculados para el bloque. Las transformaciones geométricas aplicadas a los coeficientes de filtro $f(k, l)$ y los valores de recorte de filtro correspondientes $c(k, l)$ pueden ser equivalentes a aplicar las transformaciones geométricas a las muestras en una región soportada por el filtro. Las transformaciones geométricas pueden hacer que los diferentes bloques a los que se aplica un ALF sean más similares al alinear la direccionalidad respectiva.

Se pueden realizar tres transformaciones geométricas, que incluyen un intercambio diagonal, un intercambio vertical y una rotación, tal como se describe en las ecuaciones (9)-(11), respectivamente.

$$f_D(k, l) = f(l, k), c_D(k, l) = c(l, k), \quad \text{Ec. (9)}$$

$$f_V(k, l) = f(k, K - l - 1), c_V(k, l) = c(k, K - l - 1) \quad \text{Ec. (10)}$$

$$f_R(k, l) = f(K - l - 1, k), c_R(k, l) = c(K - l - 1, k) \quad \text{Ec. (11)}$$

donde K es un tamaño del ALF o del filtro, y $0 \leq k, l \leq K - 1$ son coordenadas de coeficientes. Por ejemplo, una ubicación (0,0) está en una esquina superior izquierda y una ubicación $(K - 1, K - 1)$ está en una esquina inferior derecha del filtro f o una matriz de valores de recorte (o matriz de recorte) c . Las transformaciones se pueden aplicar a los coeficientes de filtro $f(k, l)$ y los valores de recorte $c(k, l)$ dependiendo de los valores de gradiente calculados para el bloque. En la Tabla 1 se resume un ejemplo de una relación entre la transformación y los cuatro gradientes.

Tabla 1: Mapeo del gradiente calculado para un bloque y la transformación

Valores de gradiente	Transformación
$g_{d2} < g_{d1}$ y $g_h < g_v$	Sin transformación
$g_{d2} < g_{d1}$ y $g_v < g_h$	Intercambio diagonal
$g_{d1} < g_{d2}$ y $g_h < g_v$	Intercambio vertical
$g_{d1} < g_{d2}$ y $g_v < g_h$	Rotación

En algunas realizaciones, los parámetros del filtro de ALF se señalizan en un conjunto de parámetros de adaptación (APS) para una imagen. En el APS, se pueden señalar uno o más conjuntos (p. ej., hasta 25 conjuntos) de coeficientes de filtro de luma e índices de valores de recorte. En un ejemplo, un conjunto de uno o más conjuntos puede incluir coeficientes de filtro de luma y uno o más índices de valores de recorte. Se pueden señalar uno o más conjuntos (p. ej., hasta 8 conjuntos) de coeficientes de filtro de croma e índices de valores de recorte. Para reducir la sobrecarga de señalización, se pueden fusionar los coeficientes de filtro de diferentes clasificaciones (p. ej., que tienen diferentes índices de clasificación) para los componentes de luma. En una cabecera de segmento, se pueden señalar los índices de los APS utilizados para un segmento actual.

En una realización, un índice de valor de recorte (también denominado índice de recorte) puede decodificarse desde el APS. El índice de valor de recorte se puede utilizar para determinar un valor de recorte correspondiente, por ejemplo, basándose en una relación entre el índice de valor de recorte y el valor de recorte correspondiente. La relación puede predefinirse y almacenarse en un decodificador. En un ejemplo, la relación se describe mediante una tabla, tal como una tabla de luma (p. ej., utilizada para una luma de CB) del índice de valores de recorte y el valor de recorte correspondiente, una tabla de croma (p. ej., utilizada para una croma de CB) del índice de valores de recorte y el valor de recorte correspondiente. El valor de recorte puede depender de una profundidad de bits B . La profundidad de bits B puede referirse a una profundidad de bits interna, a una profundidad de bits de las muestras reconstruidas en un CB que se filtrarán, o similares. En algunos ejemplos, se obtiene una tabla (p. ej., una tabla de luma, una tabla de croma) utilizando la ecuación. (12).

$$\text{AlfClip} = \left\{ \text{redondear} \left(2^{\frac{B-n+1}{N}} \right) \text{ para } n \in [1..N] \right\}, \quad \text{Ec. (12)}$$

donde AlfClip es el valor de recorte, B es la profundidad de bits (p. ej., bitDepth), N (p. ej., $N = 4$) es un número de valores de recorte permitidos y $(n-1)$ es el índice de valores de recorte (también denominado índice de recorte o clipIdx). La Tabla 2 muestra un ejemplo de una tabla obtenida utilizando la ecuación (12) con $N = 4$. El índice de recorte $(n-1)$ puede ser 0, 1, 2 y 3 en la Tabla 2, y n puede ser 1, 2, 3 y 4, respectivamente. La Tabla 2 se puede utilizar para bloques de luma o bloques de croma.

Tabla 2: AlfClip puede depender de la profundidad de bits B y clipIdx

bitDepth	clipIdx			
	0	1	2	3
8	255	64	16	4
9	511	108	23	5
10	1023	181	32	6
11	2047	304	45	7
12	4095	512	64	8
13	8191	861	91	10
14	16383	1448	128	11
15	32767	2435	181	13
16	65535	4096	256	16

En una cabecera de segmento para un segmento actual, se pueden señalar uno o más índices APS (p. ej., hasta 7 índices APS) para especificar conjuntos de filtros de luma que se pueden utilizar para el segmento actual. El proceso de filtrado se puede controlar en uno o más niveles adecuados, tales como un nivel de imagen, un nivel de segmento, un nivel de CTB, y/o similares. En una realización, el proceso de filtrado puede controlarse adicionalmente a nivel de CTB. Se puede señalar un indicador para indicar si el ALF se aplica a un CTB de luma. El CTB de luma puede elegir un conjunto de filtros entre una pluralidad de conjuntos de filtros fijos (p. ej., 16 conjuntos de filtros fijos) y el (los) conjunto(s) de filtros (también denominados conjuntos de filtros señalizados) que se señalizan en los APS. Se puede señalar un índice de conjunto de filtros para que el CTB de luma indique el conjunto de filtros (p. ej., el conjunto de filtros entre la pluralidad de conjuntos de filtros fijos y el (los) conjunto(s) de filtros señalizados) que se aplicará. La pluralidad de conjuntos de filtros fijos puede predefinirse y codificarse de forma rígida en un codificador y un decodificador, y pueden denominarse conjuntos de filtros predefinidos.

Para un componente de croma, se puede señalar un índice de APS en la cabecera de segmento para indicar los conjuntos de filtros de croma que se utilizarán para el segmento actual. A nivel de CTB, se puede señalar un índice de conjunto de filtros para cada CTB de croma si hay más de un conjunto de filtros de croma en el APS.

Los coeficientes de filtro se pueden cuantificar con una norma igual a 128. Para disminuir la complejidad de la multiplicación, se puede aplicar una conformidad de flujo de bits de modo que el valor del coeficiente de la posición no central pueda estar en un rango de -27 a 27 -1, ambos inclusive. En un ejemplo, el coeficiente de posición central no se señala en el flujo de bits y puede considerarse igual a 128.

En algunas realizaciones, la sintaxis y la semántica del índice de recorte y los valores de recorte se definen de la siguiente manera:

$\text{alf_luma_clip_idx}[\text{sIdx}][j]$ se puede utilizar para especificar el índice de recorte del valor de recorte que se utilizará antes de multiplicar por el coeficiente j -ésimo del filtro de luma señalado indicado por sIdx . Un requisito de conformidad del flujo de bits puede incluir que los valores de

$\text{alf_luma_clip_idx}[\text{sIdx}][j]$ con $\text{sIdx} = 0$ para $\text{alf_luma_num_filters_signalled_minus1}$ y $j = 0$ a 11 deberá estar en el rango de 0 a 3, inclusive.

Los valores de recorte del filtro de luma $\text{AlfClipL}[\text{adaptation_parameter_set_id}]$ con los elementos $\text{AlfClipL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j]$, con $\text{filtIdx} = 0$ para $\text{NumAltFilters} - 1$ y $j = 0$ a 11 se pueden derivar como se especifica en la Tabla 2, dependiendo de bitDepth establecido igual para BitDepthY y clipIdx establecido igual para $\text{alf_luma_clip_idx}[\text{alf_luma_coeff_delta_idx}[\text{filtIdx}][j]]$.

$\text{alf_chroma_clip_idx}[\text{altIdx}][j]$ se puede utilizar para especificar el índice de recorte del valor de recorte que se utilizará antes de multiplicar por el coeficiente j -ésimo del filtro de croma alternativo con índice altIdx . Un requisito de conformidad del flujo de bits puede incluir que los valores de

alf_chroma_clip_idx[altIdx][j] con altIdx = 0 para alf_chroma_num_alt_filters_minus1, j = 0 a 5 deberá estar en el rango de 0 a 3, inclusive.

Los valores de recorte de filtro de croma AlfClipC[adaptation_parameter_set_id][altIdx] con elementos AlfClipC[adaptation_parameter_set_id][altIdx][j], con altIdx = 0 para alf_chroma_num_alt_filters_minus1, j = 0 a 5 pueden derivarse como se especifica en la tabla 2 dependiendo de bitDepth establecido igual a BitDepthC y clipIdx establecido igual para alf_chroma_clip_idx[altIdx][j].

En una realización, el proceso de filtrado se puede describir a continuación. En el lado del decodificador, cuando el ALF está habilitado para un CTB, se puede filtrar una muestra R(i,j) dentro de una CU (o CB), lo que da como resultado un valor de muestra filtrado R'(i,j) como se muestra a continuación utilizando la ecuación. (13). En un ejemplo, cada muestra en la CU se filtra.

$$R'(i,j) = R(i,j) + \left(\left(\sum_{k \neq 0} \sum_{l \neq 0} f(k,l) \times K(R(i+k,j+l) - R(i,j), c(k,l)) + 64 \right) \gg 7 \right)$$

Ec. (13)

donde f(k,l) indica los coeficientes de filtro decodificados, K(x,y) es una función de recorte y c(k,l) indica los parámetros de recorte decodificados (o valores de recorte). Las variables k y l pueden variar entre -L/2 y L/2, donde L indica una longitud de filtro. La función de recorte K(x, y) = min (y, max(-y, x)) corresponde a una función de recorte Clip3 (-y, y, x). Al incorporar la función de recorte K(x, y), el método de filtrado en bucle (p. ej., ALF) se convierte en un proceso no lineal y puede denominarse ALF no lineal.

En el ALF no lineal, se pueden proporcionar múltiples conjuntos de valores de recorte en la Tabla 3. En un ejemplo, un conjunto de luma incluye cuatro valores de recorte {1024, 181, 32, 6}, y un conjunto de croma incluye 4 valores de recorte {1024, 161, 25, 4}. Los cuatro valores de recorte en el conjunto de luma se pueden seleccionar separando aproximadamente por igual, en un dominio logarítmico, un rango completo (p. ej., 1024) de los valores de muestra (codificados en 10 bits) para un bloque de luma. El rango puede ser de 4 a 1024 para el conjunto de croma.

Tabla 3: Ejemplos de valores de recorte

	Grupo de mosaicos INTRA/INTER
LUMA	{ 1024, 181, 32, 6 }
CHROMA	{ 1024, 161, 25, 4 }

Los valores de recorte seleccionados se pueden codificar en un elemento sintáctico “alf_data” de la siguiente manera: se puede utilizar un esquema de codificación adecuado (p. ej., un esquema de codificación Golomb) para codificar un índice de recorte correspondiente al valor de recorte seleccionado, tal como se muestra en la tabla 3. El esquema de codificación puede ser el mismo esquema de codificación utilizado para codificar el índice del conjunto de filtros.

En un ejemplo, se puede utilizar un proceso de filtrado de límites virtuales para reducir el requisito de memoria intermedia de línea del ALF. En consecuencia, la clasificación y el filtrado de bloques modificados se pueden emplear para muestras cercanas a los límites de la CTU (p. ej., un límite de CTU horizontal). Un límite virtual (1130) se puede definir como una línea desplazando un límite (1120) de CTU horizontal por muestras de “N_{muestras}”, como se muestra en la Figura 11A, donde N_{muestras} pueden ser un número entero positivo. En un ejemplo, N_{muestras} son iguales a 4 para un componente de luma, y N_{muestras} son iguales a 2 para un componente de croma.

Haciendo referencia a la Figura 11A, se puede aplicar una clasificación de bloques modificada para un componente de luma. En un ejemplo, para el cálculo del gradiente laplaciano 1D de un bloque de 4x4 (1110) por encima del límite virtual (1130), solo se utilizan muestras por encima del límite virtual (1130). Similarmente, haciendo referencia a la Figura 11B, para un cálculo del gradiente laplaciano 1D de un bloque (1111) de 4x4 por debajo de un límite virtual (1131) que se desplaza desde un límite (1121) de CTU, solo se utilizan muestras por debajo del límite virtual (1131). La cuantificación de un valor de actividad A se puede escalar en consecuencia teniendo en cuenta un número reducido de muestras utilizadas en el cálculo del gradiente laplaciano 1D.

Para un procesamiento de filtrado, se puede utilizar una operación de espacios adicionales simétricos en los límites virtuales tanto para un componente de luma como para un componente de croma. Las Figuras 12A-12F ilustran ejemplos de tal filtrado de ALF modificado para un componente de luma en límites virtuales. Cuando una muestra que se está filtrando se encuentra por debajo de un límite virtual, las muestras vecinas que se encuentran por encima del límite virtual se pueden rellenar. Cuando una muestra que se está filtrando se encuentra por encima de un límite virtual, las muestras vecinas que se encuentran por debajo del límite virtual se pueden rellenar. Haciendo referencia a la Figura 12A, una muestra vecina C0 puede incluir espacios adicionales con una muestra C2 que se encuentra por

debajo de un límite virtual (1210). Haciendo referencia a la Figura 12B, una muestra vecina C0 puede incluir espacios adicionales con una muestra C2 que se encuentra por encima de un límite virtual (1220). Haciendo referencia a la Figura 12C, las muestras vecinas C1-C3 pueden incluir espacios adicionales con las muestras C5-C7, respectivamente, que se encuentran por debajo de un límite virtual (1230). Haciendo referencia a la Figura 12D, las muestras vecinas C1-C3 pueden incluir espacios adicionales con las muestras C5-C7, respectivamente, que se encuentran por encima de un límite virtual (1240). Haciendo referencia a la Figura 12E, las muestras vecinas C4-C8 se pueden incluir espacios adicionales con las muestras C10, C11, C12, C11 y C10, respectivamente, que se encuentran por debajo de un límite virtual (1250). Haciendo referencia a la Figura 12F, las muestras vecinas C4-C8 pueden incluir espacios adicionales con las muestras C10, C11, C12, C11 y C10, respectivamente, que se encuentran por encima de un límite virtual (1260).

En algunos ejemplos, la descripción anterior se puede adaptar adecuadamente cuando la(s) muestra(s) y la(s) muestra(s) vecinas está(n) ubicada(s) a la izquierda (o a la derecha) y a la derecha (o a la izquierda) de un límite virtual.

Según un aspecto de la descripción, para mejorar la eficiencia de la codificación, las imágenes se pueden particionar basándose en el proceso de filtrado. En algunos ejemplos, una CTU también se denomina unidad de codificación más grande (LCU, en inglés). En un ejemplo, la CTU o LCU pueden tener un tamaño de 64x64 píxeles. En algunas realizaciones, la separación de árbol cuaternario de imágenes alineada con la LCU se puede utilizar para la partición basada en el filtrado. En algunos ejemplos, se puede utilizar el filtro en bucle adaptativo basado en árboles cuádruples de imágenes sincrónicas de la unidad de codificación. Por ejemplo, la imagen de luma se puede separar en varias particiones de árbol cuaternario de varios niveles, y cada límite de partición se alinea con los límites de las LCU. Cada partición tiene su propio proceso de filtrado y, por lo tanto, se denomina unidad de filtro (FU, por sus siglas en inglés).

En algunos ejemplos, se puede utilizar un flujo de codificación de 2 pasos. En un primer paso del flujo de codificación de 2 pasos, se puede determinar el patrón de separación en árbol cuaternario de la imagen y el mejor filtro de cada FU. En algunas realizaciones, la determinación del patrón de separación en árbol cuaternario de la imagen y la determinación de los mejores filtros para las FU se basan en las distorsiones de filtrado. Las distorsiones de filtrado se pueden estimar mediante la técnica de estimación rápida de distorsión de filtrado (FFDE, por sus siglas en inglés) durante el proceso de determinación. La imagen se particiona utilizando una división de árbol cuádruple. Según el patrón de separación de árbol cuaternario determinado y los filtros seleccionados de todas las FU, la imagen reconstruida se puede filtrar.

En un segundo paso del flujo de codificación de 2 pasos, se realiza el control de encendido/apagado del ALF síncrono de la CU. Según los resultados de encendido/apagado de ALF, la imagen reconstruida recupera parcialmente la primera imagen filtrada.

Específicamente, en algunos ejemplos, se adopta una estrategia de separación de arriba hacia abajo para dividir una imagen en particiones de árbol cuaternario de varios niveles mediante el uso de un criterio de distorsión de velocidad. Cada partición se denomina unidad de filtro (FU). El proceso de separación alinea las particiones de árbol cuaternario con los límites de la LCU. El orden de codificación de las FU sigue el orden de escaneo z.

La Figura 13 muestra un ejemplo de partición según algunas realizaciones de la descripción. En el ejemplo de la Figura 13, una imagen (1300) se separa en 10 FU, y el orden de codificación es FU0, FU1, FU2, FU3, FU4, FU5, FU6, FU7, FU8 y FU9.

La Figura 14 muestra un patrón (1400) de división de árbol cuaternario para la imagen (1300). En el ejemplo de la Figura 14, los indicadores de separación se utilizan para indicar el patrón de partición de la imagen. Por ejemplo, "1" indica que se ha realizado una partición de árbol cuaternario en el bloque; y "0" indica que el bloque no se particiona adicionalmente. En algunos ejemplos, un tamaño mínimo de FU tiene el tamaño de LCU, y no se necesita ningún indicador de separación para el tamaño mínimo de FU. Los indicadores separados se codifican y transmiten en orden z, como se muestra en la Figura 14.

En algunos ejemplos, el filtro de cada FU se selecciona entre dos conjuntos de filtros basándose en el criterio de distorsión de velocidad. El primer conjunto tiene filtros con forma cuadrada y rómbica simétricos 1/2 derivados de la FU actual. El segundo conjunto proviene de memorias intermedias de filtro con retardo temporal; las memorias intermedias de filtro con retardo temporal almacenan los filtros previamente derivados para las FU de imágenes anteriores. El filtro con el coste mínimo de distorsión de velocidad de estos dos conjuntos se puede elegir para la FU actual. Similarmente, si la FU actual no es la FU más pequeña y se puede separar adicionalmente en 4 FU hijas, se calculan los costes de distorsión de velocidad de las 4 FU hijas. Al comparar el coste de distorsión de la velocidad de los casos separados y no separados de forma recursiva, se puede decidir el patrón de separación en árbol cuaternario de imagen.

En algunos ejemplos, se puede utilizar un nivel máximo de separación de árbol cuaternario para limitar el número máximo de FU. En un ejemplo, cuando el nivel máximo de separación del árbol cuádruple es 2, el número máximo de FU es 16. Además, durante la determinación de la separación en árbol cuaternario, se pueden reutilizar los valores de

correlación para derivar los coeficientes de Wiener de las 16 FU en el nivel inferior del árbol cuaternario (FU más pequeñas). El resto de las FU pueden derivar sus filtros Wiener a partir de las correlaciones de las 16 FU en el nivel inferior del árbol cuaternario. Por lo tanto, en el ejemplo, solo se realiza un acceso a la memoria intermedia de fotografías para derivar los coeficientes de filtro de todas las FU.

Una vez decidido el patrón de separación en árbol cuaternario, para reducir adicionalmente la distorsión del filtrado, se puede realizar el control de encendido/apagado del ALF síncrono de la CU. Al comparar la distorsión de filtrado y la distorsión de no filtrado en cada CU de hoja, la CU de hoja puede activar/desactivar explícitamente el ALF en su región local. En algunos ejemplos, la eficiencia de codificación puede mejorarse adicionalmente rediseñando los coeficientes de filtro según los resultados de encendido/apagado del ALF.

Un proceso de filtrado de componentes cruzados puede aplicar filtros de componentes cruzados, tales como los filtros en bucle adaptativo de componentes cruzados (CC-ALF). El filtro de componentes cruzados puede utilizar valores de muestra de luma de un componente de luma (p. ej., una luma de CB) para refinar un componente de croma (p. ej., una croma de CB correspondiente a la luma de CB). En un ejemplo, el luma de CB y el croma de CB se incluyen en una CU.

La Figura 15 muestra filtros de componentes cruzados (p. ej., CC-ALF) utilizados para generar componentes de croma según una realización de la descripción. En algunos ejemplos, la Figura 15 muestra procesos de filtrado para un primer componente de croma (p. ej., un primer croma de CB), un segundo componente de croma (p. ej., un segundo croma de CB), y un componente de luma (p. ej., un luma d CB). El componente de luma puede filtrarse mediante un filtro (1510) de desfase adaptativo de muestras (SAO) para generar un componente (1541) de luma filtrado por SAO. El componente (1541) de luma filtrado por SAO puede filtrarse adicionalmente mediante un filtro (1516) de luma de ALF para convertirse en una luma (1561) de CB filtrada (p. ej., "Y").

El primer componente de croma se puede filtrar mediante un filtro (1512) de SAO y un filtro (1518) de croma ALF para generar un primer componente intermedio (1552). Además, el componente (1541) de luma filtrado por SAO puede filtrarse mediante un filtro (1521) de componentes cruzados (p. ej., CC-ALF) para que el primer componente de croma genere un segundo componente intermedio (1542). Posteriormente, se puede generar un primer componente (1562) de croma filtrado (p. ej., 'Cr') basándose en al menos uno del segundo componente intermedio (1542) y el primer componente intermedio (1552). En un ejemplo, el primer componente (1562) de croma filtrado (p. ej., 'Cb') se puede generar combinando el segundo componente intermedio (1542) y el primer componente intermedio (1552) con un sumador (1522). El proceso de filtrado en bucle adaptativo entre componentes para el primer componente de croma puede incluir un paso realizado por el CC-ALF (1521) y un paso realizado, por ejemplo, por el sumador (1522).

La descripción anterior se puede adaptar al segundo componente de croma. El segundo componente de croma se puede filtrar mediante un filtro (1514) de SAO y el filtro (1518) de croma ALF para generar un tercer componente intermedio (1553). Además, el componente (1541) de luma filtrado por SAO puede filtrarse mediante un filtro (1531) de componentes cruzados (p. ej., CC-ALF) para que el segundo componente de croma genere un cuarto componente intermedio (1543). Posteriormente, se puede generar un segundo componente (1563) de croma filtrado (p. ej., 'Cr') basándose en al menos uno del cuarto componente intermedio (1543) y el tercer componente intermedio (1553). En un ejemplo, el segundo componente (1563) de croma filtrado (p. ej., 'Cr') se puede generar combinando el cuarto componente intermedio (1543) y el tercer componente intermedio (1553) con un sumador (1532). En un ejemplo, el proceso de filtrado en bucle adaptativo de componentes cruzados para el segundo componente de croma puede incluir un paso realizado por el CC-ALF (1531) y un paso realizado, por ejemplo, por el sumador (1532).

Un filtro de componentes cruzados (p. ej., el CC-ALF (1521), el CC-ALF (1531)) puede operar aplicando un filtro lineal que tenga cualquier forma de filtro adecuada al componente de luma (o un canal de luma) para refinar cada componente de croma (p. ej., el primer componente de croma, el segundo componente de croma).

La Figura 16 muestra un ejemplo de un filtro (1600) según una realización de la descripción. El filtro (1600) puede incluir coeficientes de filtro distintos de cero y coeficientes de filtro cero. El filtro (1600) tiene una forma (1620) de diamante formada por coeficientes (1610) de filtro (indicados por círculos con relleno negro). En un ejemplo, los coeficientes de filtro distintos de cero en el filtro (1600) se incluyen en los coeficientes (1610) de filtro, y los coeficientes de filtro no incluidos en los coeficientes (1610) de filtro son cero. Por lo tanto, los coeficientes de filtro distintos de cero en el filtro (1600) se incluyen en la forma (1620) de diamante, y los coeficientes de filtro no incluidos en la forma (1620) de diamante son cero. En un ejemplo, un número de los coeficientes de filtro del filtro (1600) es igual a un número de los coeficientes (1610) de filtro, que es 18 en el ejemplo mostrado en la Figura 16.

El CC-ALF puede incluir cualquier coeficiente de filtro adecuado (también denominado coeficientes de filtro CC-ALF). Haciendo referencia de nuevo a la Figura 15, el CC-ALF (1521) y el CC-ALF (1531) pueden tener la misma forma de filtro, tal como la forma (1620) de diamante mostrada en la Figura 16, y un mismo número de coeficientes de filtro. En un ejemplo, los valores de los coeficientes de filtro en el CC-ALF (1521) son diferentes de los valores de los coeficientes de filtro en el CC-ALF (1531).

En general, los coeficientes de filtro (p. ej., coeficientes de filtro distintos de cero) en un CC-ALF pueden transmitirse, por ejemplo, en el APS. En un ejemplo, los coeficientes de filtro se pueden escalar por un factor (p. ej., 2^{10}) y se pueden redondear para una representación de punto fijo. La aplicación de un CC-ALF puede controlarse en un tamaño de bloque variable y señalizarse mediante un indicador codificado por contexto (p. ej., un indicador de habilitación de CC-ALF) recibido para cada bloque de muestras. El indicador codificado por contexto, tal como el indicador de habilitación CC-ALF, se puede señalar en cualquier nivel adecuado, tal como un nivel de bloque. El tamaño del bloque junto con el indicador de habilitación de CC-ALF se pueden recibir a nivel de segmento para cada componente de croma. En algunos ejemplos, se pueden admitir tamaños de bloque (en muestras de croma) de 16×16 , 32×32 , y 64×64 .

La Figura 17 muestra un ejemplo de sintaxis para CC-ALF según algunas realizaciones de la descripción. En el ejemplo de la Figura 17,

`alf_ctb_cross_component_cb_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]` es un índice para indicar si se utiliza un filtro Cb de componentes cruzados y un índice del filtro Cb de componentes cruzados, si se utiliza. Por ejemplo, cuando

`alf_ctb_cross_component_cb_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]` es igual a 0, el filtro Cb de componentes cruzados no se aplica al bloque de muestras de componentes de color Cb en la ubicación de luma (`xCtb`, `yCtb`); cuando

`alf_ctb_cross_component_cb_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]` no es igual a 0,

`alf_ctb_cross_component_cb_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]` es un índice para aplicar un filtro. Por ejemplo,

`alf_ctb_cross_component_cb_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]`-ésimo filtro Cb de componentes cruzados se aplica al bloque de muestras de componentes de color Cb en la ubicación de luma (`xCtb`, `yCtb`)

Además, en el ejemplo de la Figura 17,

`alf_ctb_cross_component_cr_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]` se utiliza para indicar si se utiliza un filtro de Cr de componentes cruzados y se utiliza el índice del filtro Cr de componentes cruzados. Por ejemplo, cuando

`alf_ctb_cross_component_cr_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]` es igual a 0, el filtro Cr de componentes cruzados no se aplica al bloque de muestras de componentes de color Cr en la ubicación de luma (`xCtb`, `yCtb`); cuando

`alf_ctb_cross_component_cr_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]` no es igual a 0,

`alf_ctb_cross_component_cr_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]` es el índice del filtro Cr de componentes cruzados. Por ejemplo,

`alf_cross_component_cr_idc[ xCtb >> CtbLog2SizeY ][ yCtb >> CtbLog2SizeY ]`-ésimo filtro Cr de componentes cruzados se puede aplicar al bloque de muestras del componente de color Cr en la ubicación de luma (`xCtb`, `yCtb`)

En algunos ejemplos, se utilizan técnicas de submuestreo de croma, por lo tanto, un número de muestras en cada uno de los bloques de croma puede ser menor que un número de muestras en el bloque de luma. Un formato de submuestreo de croma (también denominado formato de submuestreo de croma, p. ej., especificado por `chroma_format_idc`) puede indicar un factor de submuestreo horizontal de croma (p. ej., `SubWidthC`) y un factor de submuestreo vertical de croma (p. ej., `SubHeightC`) entre cada uno de los bloques de croma y el bloque de luma correspondiente. En un ejemplo, el formato de submuestreo de croma es $4:2:0$ y, por lo tanto, el factor de submuestreo horizontal de croma (p. ej., `SubWidthC`) y el factor de submuestreo vertical de croma (p. ej., `SubHeightC`) es 2, como se muestra en las Figuras 18A-18B. En un ejemplo, el formato de submuestreo de croma es $4:2:2$, y, por lo tanto, el factor de submuestreo horizontal de croma (p. ej., `SubWidthC`) es 2, y el factor de submuestreo vertical de croma (p. ej., `SubHeightC`) es 1. En un ejemplo, el formato de submuestreo de croma es $4:4:4$, y, por lo tanto, el factor de submuestreo horizontal de croma (p. ej., `SubWidthC`) y el factor de submuestreo vertical de croma (p. ej., `SubHeightC`) son 1. Un tipo de muestra de croma (también denominado posición de muestra de croma) puede indicar una posición relativa de una muestra de croma en el bloque de croma con respecto a al menos una muestra de luma correspondiente en el bloque de luma.

Las Figuras 18A-18B muestran ubicaciones ilustrativas de muestras de croma con respecto a muestras de luma según realizaciones de la descripción. Haciendo referencia a la Figura 18A, las muestras (1801) de luma están ubicadas en las filas (1811)-(1818). Las muestras (1801) de luma mostradas en la Figura 18A pueden representar una parte de una imagen. En un ejemplo, un bloque de luma (p. ej., un luma CB) incluye las muestras (1801) de luma. El bloque de luma puede corresponder a dos bloques de croma que tienen el formato de submuestreo de croma de $4:2:0$. En un ejemplo, cada bloque de croma incluye muestras (1803) de croma. Cada muestra de croma (p. ej., la muestra (1803(1)) de croma corresponde a cuatro muestras de luma (p. ej., las muestras (1801(1))-(1801(4)) de luma). En un ejemplo, las cuatro muestras de luma son la muestra (1801(1)) superior izquierda, la muestra (1801(2)) superior derecha, la muestra (1801(3)) inferior izquierda, y la muestra (1801(4)) inferior derecha. La muestra de croma (p. ej., (1803(1))) está ubicada en una posición central izquierda que está entre la muestra (1801(1)) superior izquierda y la muestra (1801(3)) inferior izquierda, y un tipo de muestra de croma del bloque de croma que tiene las muestras (1803) de croma puede denominarse muestra de croma tipo 0. La muestra de croma tipo 0 indica una posición relativa 0 correspondiente a la posición central izquierda en el centro de la muestra (1801(1)) superior izquierda y la muestra

(1801(3)) inferior izquierda. Las cuatro muestras de luma (p. ej., (1801(1))-(1801(4))) pueden denominarse muestras de luma vecinas de la muestra (1803)(1) de croma.

En un ejemplo, cada bloque de croma incluye muestras (1804) de croma. La descripción anterior con referencia a las muestras (1803) de croma se puede adaptar a las muestras (1804) de croma y, por lo tanto, las descripciones detalladas se pueden omitir por motivos de brevedad. Cada una de las muestras (1804) de croma puede estar ubicada en una posición central de cuatro muestras de luma correspondientes, y un tipo de muestra de croma del bloque de croma que tiene las muestras (1804) de croma puede denominarse muestra de croma tipo 1. La muestra de croma tipo 1 indica una posición relativa 1 que corresponde a la posición central de las cuatro muestras de luma (p. ej., (1801(1))-(1801(4))). Por ejemplo, una de las muestras (1804) de croma puede estar ubicada en una parte central de las muestras (1801(1))-(1801(4)) de luma.

En un ejemplo, cada bloque de croma incluye muestras (1805) de croma. Cada una de las muestras (1805) de croma se puede ubicar en una posición superior izquierda que se ubica junto con la muestra superior izquierda de las cuatro muestras (1801) de luma correspondientes, y un tipo de muestra de croma del bloque de croma que tiene las muestras (1805) de croma se puede denominar muestra de croma tipo 2. En consecuencia, cada una de las muestras (1805) de croma se ubica junto con la muestra superior izquierda de las cuatro muestras (1801) de luma correspondientes a la muestra de croma respectiva. La muestra de croma tipo 2 indica una posición relativa 2 correspondiente a la posición superior izquierda de las cuatro muestras (1801) de luma. Por ejemplo, una de las muestras (1805) de croma puede estar ubicada en la posición superior izquierda de las muestras (1801(1))-(1801(4)) de luma.

En un ejemplo, cada bloque de croma incluye muestras (1806) de croma. Cada una de las muestras (1806) de croma puede ubicarse en una posición central superior entre una muestra correspondiente de arriba a la izquierda y una muestra de arriba a la derecha correspondiente, y un tipo de muestra de croma del bloque de croma que tiene las muestras (1806) de croma puede denominarse muestra de croma tipo 3. La muestra de croma tipo 3 indica una posición relativa 3 que corresponde a la posición central superior entre la muestra superior izquierda (y la muestra superior derecha). Por ejemplo, una de las muestras (1806) de croma puede estar ubicada en la posición central superior de las muestras (1801(1))-(1801(4)) de luma.

En un ejemplo, cada bloque de croma incluye muestras (1807) de croma. Cada una de las muestras (1807) de croma se puede ubicar en una posición inferior izquierda que se ubica junto con la muestra inferior izquierda de las cuatro muestras (1801) de luma correspondientes, y un tipo de muestra de croma del bloque de croma que tiene las muestras (1807) de croma se puede denominar muestra de croma tipo 4. En consecuencia, cada una de las muestras (1807) de croma se ubica junto con la muestra inferior izquierda de las cuatro muestras (1801) de luma correspondientes a la muestra de croma respectiva. La muestra de croma tipo 4 indica una posición relativa 4 correspondiente a la posición inferior izquierda de las cuatro muestras (1801) de luma. Por ejemplo, una de las muestras (1807) de croma puede estar ubicada en la posición inferior izquierda de las muestras (1801(1))-(1801(4)) de luma.

En un ejemplo, cada bloque de croma incluye muestras (1808) de croma. Cada una de las muestras (1808) de croma está ubicada en una posición central inferior entre la muestra de la parte inferior izquierda y la muestra de la parte inferior derecha, y un tipo de muestra de croma del bloque de croma que tiene las muestras (1808) de croma puede denominarse muestra de croma tipo 5. La muestra de croma tipo 5 indica una posición relativa 5 que corresponde a la posición central inferior entre la muestra inferior izquierda y la muestra inferior derecha de las cuatro muestras (1801) de luma. Por ejemplo, una de las muestras (1808) de croma puede estar ubicada entre la muestra inferior izquierda y la muestra inferior derecha de las muestras (1801(1))-(1801(4)) de luma.

En general, se puede utilizar cualquier tipo de muestra de croma adecuado para un formato de submuestreo de croma. Los tipos 0-5 de muestras de croma son tipos de muestras de croma ilustrativos descritos con el formato 4:2:0 de submuestreo de croma. Se pueden utilizar tipos de muestras de croma adicionales para el formato 4:2:0 de submuestreo de croma. Además, se pueden utilizar otros tipos de muestras de croma y/o variaciones de los tipos 0-5 de muestras de croma para otros formatos de submuestreo de croma, tales como 4:2:2, 4:4:4 o similares. En un ejemplo, se utiliza un tipo de muestra de croma que combina las muestras (1805) y (1807) de croma para el formato 4:2:2 de submuestreo de croma.

En un ejemplo, se considera que el bloque de luma tiene filas alternas, tales como las filas (1811)-(1812) que incluyen las dos muestras superiores (p. ej., (1801(1))-(1801(2))) de las cuatro muestras de luma (p. ej., (1801(1))-(1801(4))) y las dos muestras inferiores (p. ej., (1801(3))-(1801(4))) de las cuatro muestras de luma (p. ej., (1801(1))-(1801(4))), respectivamente. En consecuencia, las filas (1811), (1813), (1815) y (1817) pueden denominarse filas actuales (también denominadas campo superior), y las filas (1812), (1814), (1816) y (1818) pueden denominarse filas siguientes (también denominadas campo inferior). Las cuatro muestras de luma (p. ej., (1801(1))-(1801(4))) están ubicadas en la fila actual (p. ej., (1811)) y en la siguiente fila (p. ej., (1812)). Las posiciones relativas 2-3 están ubicadas en las filas actuales, las posiciones relativas 0-1 están ubicadas entre cada fila actual y la fila siguiente respectiva, y las posiciones relativas 4-5 están ubicadas en las siguientes filas.

Las muestras (1803), (1804), (1805), (1806), (1807), o (1808) de croma están ubicadas en las filas (1851)-(1854) en cada bloque de croma. Las ubicaciones específicas de las filas (1851)-(1854) pueden depender del tipo de muestra

de croma de las muestras de croma. Por ejemplo, para las muestras (1803)-(1804) de croma que tienen los respectivos tipos 0-1 de muestra de croma, la fila (1851) está ubicada entre las filas (1811)-(1812). Para las muestras (1805)-(1806) de croma que tienen los respectivos tipos 2-3 de muestra de croma, la fila (1851) se ubica junto con la fila actual (1811). Para las muestras (1807)-(1808) de croma que tienen los respectivos tipos 4-5 de muestra de croma, la fila (1851) se ubica junto con la siguiente fila (1812). Las descripciones anteriores se pueden adaptar adecuadamente a las filas (1852)-(1854), y las descripciones detalladas se omiten por motivos de brevedad.

Se puede utilizar cualquier método de escaneo adecuado para mostrar, almacenar, y/o transmitir el bloque de luma y el (los) bloque(s) de croma correspondientes descritos anteriormente en la Figura 18A. En un ejemplo, se utiliza el escaneo progresivo.

Se puede utilizar un escaneo entrelazado, como se muestra en la Figura 18B. Como se ha descrito anteriormente, el formato de submuestreo de croma es 4:2:0 (p. ej., `chroma_format_idc` es igual a 1). En un ejemplo, un tipo de ubicación de croma variable (p. ej., `ChromaLocType`) indica las filas actuales (p. ej., `ChromaLocType` es `chroma_sample_loc_type_top_field`) o las siguientes filas (p. ej., `ChromaLocType` es `chroma_sample_loc_type_bottom_field`). Las filas actuales (1811), (1813), (1815) y (1817) y las siguientes filas (1812), (1814), (1816) y (1818) se pueden escanear por separado, por ejemplo, las filas actuales (1811), (1813), (1815) y (1817) se pueden escanear primero seguidas de las siguientes filas (1812), (1814), (1816) y (1818) siendo escaneados. Las filas actuales pueden incluir las muestras (1801) de luma mientras que las siguientes filas pueden incluir las muestras (1802) de luma.

Similarmente, el bloque de croma correspondiente se puede escanear de forma entrelazada. Las filas (1851) y (1853) que incluyen las muestras (1803), (1804), (1805), (1806), (1807), o (1808) de croma sin relleno pueden denominarse filas actuales (o filas de croma actuales), y las filas (1852) y (1854) incluyen las muestras (1803), (1804), (1805), (1806), (1807) o (1808) de croma con relleno gris pueden denominarse filas siguientes (o filas de croma siguientes). En un ejemplo, durante el escaneo entrelazado, primero se escanean las filas (1851) y (1853) y, a continuación, se escanean las filas (1852) y (1854).

En algunos ejemplos, se pueden utilizar técnicas de filtrado de mejora direccional restringida. El uso de un filtro de mejora direccional restringido (CDEF, por sus siglas en inglés) en bucle puede filtrar artefactos de codificación, mientras conserva los detalles de la imagen. En un ejemplo (p. ej., HEVC), el algoritmo de desfase adaptativo de muestras (SAO) puede lograr un objetivo similar definiendo compensaciones de señal para diferentes clases de píxeles. A diferencia de SAO, el CDEF es un filtro espacial no lineal. En algunos ejemplos, el CDEF puede restringirse para que sea fácilmente vectorizable (es decir, implementable con operaciones de datos múltiples de una única instrucción (SIMD, por sus siglas en inglés)). Se observa que otros filtros no lineales, tales como un filtro de mediana, o un filtro bilateral, no pueden manejarse de la misma manera.

En algunos casos, la cantidad de artefactos de timbre en una imagen codificada tiende a ser aproximadamente proporcional al tamaño del paso de cuantificación. La cantidad de detalle es una propiedad de la imagen de entrada, pero el detalle más pequeño retenido en la imagen cuantificada también tiende a ser proporcional al tamaño del paso de cuantificación. Para un tamaño del paso de cuantificación dado, la amplitud de la oscilación es generalmente menor que la amplitud de los detalles.

El CDEF se puede utilizar para identificar la dirección de cada bloque y después filtrar de manera adaptativa a lo largo de la dirección identificada y filtrar en menor medida a lo largo de las direcciones giradas 45 grados desde la dirección identificada. En algunos ejemplos, un codificador puede buscar las intensidades del filtro y las intensidades del filtro se pueden señalar explícitamente, lo que permite un alto grado de control sobre el desenfoque.

Específicamente, en algunos ejemplos, la búsqueda de dirección se realiza en los píxeles reconstruidos, justo después del filtro de desbloqueo. Dado que esos píxeles están disponibles para el decodificador, el decodificador puede buscar las direcciones y, por lo tanto, las direcciones no requieren señalización en un ejemplo. En algunos ejemplos, la búsqueda de direcciones puede operar en ciertos tamaños de bloque, tales como bloques de 8×8 , que son lo suficientemente pequeños como para manejar adecuadamente los bordes no rectos, a la vez que son lo suficientemente grandes como para estimar de manera confiable las direcciones cuando se aplican a una imagen cuantificada. También, tener una dirección constante en una región de 8×8 facilita la vectorización del filtro. En algunos ejemplos, cada bloque (p. ej., 8×8) se puede comparar con bloques perfectamente direccionales para determinar la diferencia. Un bloque perfectamente direccional es un bloque donde todos los píxeles de una línea en una dirección tienen el mismo valor. En un ejemplo, se puede calcular una medida de diferencia del bloque y cada uno de los bloques perfectamente direccionales, tal como la suma de diferencias cuadráticas (SSD, por sus siglas en inglés), error de valor cuadrático medio (RMS, por sus siglas en inglés). A continuación, se puede determinar un bloque perfectamente direccional con una diferencia mínima (p. ej., SSD mínima, RMS mínima y similares) y la dirección del bloque perfectamente direccional determinado puede ser la dirección que mejor se adapte al patrón del bloque.

La Figura 19 muestra un ejemplo de búsqueda de dirección según una realización de la descripción. En un ejemplo, un bloque (1910) es un bloque de 8×8 que se reconstruye, y se emite desde un filtro de desbloqueo. En el ejemplo de la Figura 19, la búsqueda de dirección puede determinar una dirección a partir de las 8 direcciones mostradas por

(1920) para el bloque (1910). Se forman 8 bloques (1930) perfectamente direccionales que corresponden respectivamente a las 8 direcciones (1920). Un bloque perfectamente direccional correspondiente a una dirección es un bloque donde los píxeles a lo largo de una línea de la dirección tienen el mismo valor. Además, se puede calcular una medida de diferencia, tal como SSD, error de RMS y similares, del bloque (1910) y cada uno de los bloques (1930) perfectamente direccionales. En el ejemplo de la Figura 19, los errores de RMS se muestran con (1940). Como muestra en (1943), el error de RMS del bloque (1910) y del bloque (1933) perfectamente direccional es el más pequeño, por lo tanto, la dirección (1923) es la dirección que mejor coincide con el patrón en el bloque (1910).

Una vez identificada la dirección del bloque, se puede determinar un filtro direccional de paso bajo no lineal. Por ejemplo, las derivaciones de filtro del filtro direccional de paso bajo no lineal se pueden alinear a lo largo de la dirección identificada para reducir la oscilación, mientras conserva los bordes o patrones direccionales. Sin embargo, en algunos ejemplos, el filtrado direccional por sí solo a veces no puede reducir suficientemente el timbre. En un ejemplo, también se utilizan derivaciones de filtro adicionales en píxeles que no se encuentran a lo largo de la dirección identificada. Para reducir el riesgo de borrosidad, las derivaciones de filtro adicionales se tratan de forma más conservadora. Por este motivo, el CDEF incluye derivaciones de filtro primarios y derivaciones de filtro secundarios. En un ejemplo, un filtro de CDEF bidimensional completo se puede expresar como la ecuación (14):

$$y(i,j) = x(i,j) + \text{redondear} \left(\sum_{m,n} w_{d,m,n}^{(p)} f(x(m,n) - x(i,j), S^{(p)}, D) + \sum_{m,n} w_{d,m,n}^{(s)} f(x(m,n) - x(i,j), S^{(s)}, D) \right), \quad \text{Ec. (14)}$$

donde D indica un parámetro de amortiguamiento, $S^{(p)}$ indica la intensidad de las derivaciones del filtro primario, $S^{(s)}$ indica la intensidad de las derivaciones del filtro secundario, $\text{round}(\cdot)$ indica una operación que redondea los lazos alejándolos de cero, w indica los pesos del filtro y $f(d, S, D)$ es una función de restricción que opera en la diferencia entre el píxel filtrado y cada uno de los píxeles vecinos. En un ejemplo, para pequeñas diferencias, la función $f(d, S, D)$ es igual a D , lo que puede hacer que el filtro se comporte como un filtro lineal; cuando la diferencia es grande, la función $f(d, S, D)$ es igual a 0, lo que puede ignorar eficazmente los toques del filtro.

En algunos ejemplos, los esquemas de restauración en bucle se utilizan en la codificación de vídeo después del desbloqueo para, en general, eliminar el ruido y mejorar la calidad de los bordes, más allá de la operación de desbloqueo. En un ejemplo, los esquemas de restauración en bucle se pueden cambiar dentro de un fotograma por cada mosaico del tamaño adecuado. Los esquemas de restauración en bucle se basan en filtros Wiener simétricos separables, filtros autoguiados duales con proyección subespacial, y filtros recursivos de transformación de dominio. Debido a que las estadísticas de contenido pueden variar sustancialmente dentro de un fotograma, los esquemas de restauración en bucle se integran dentro de un marco conmutable donde se pueden activar diferentes esquemas en diferentes regiones del fotograma.

El filtro Wiener simétrico separable puede ser uno de los esquemas de restauración en bucle. En algunos ejemplos, cada píxel de un fotograma degradado se puede reconstruir como una versión filtrada no causal de los píxeles dentro de una ventana $w \times w$ a su alrededor, donde $w = 2r + 1$ es impar para el número entero r . Si las derivaciones del filtro 2D se indican con un vector F de elementos $w^2 \times 1$ en forma vectorizada de columnas, una optimización sencilla de LMMSE hace que los parámetros del filtro se den como $F = H^{-1} M$, donde $H = E[XX^T]$ es la autocovarianza de x , la versión vectorizada por columnas de las muestras w^2 de la ventana $w \times w$ alrededor de un píxel, y $M = E[YX^T]$ es la correlación cruzada de x con la muestra fuente escalar y , que se va a estimar. En un ejemplo, el codificador puede estimar H y M a partir de las realizaciones en los fotogramas desbloqueados y la fuente y puede enviar el filtro F resultante al decodificador. Sin embargo, eso no solo implicaría un coste de velocidad de bits sustancial al transmitir con w^2 derivaciones, sino que el filtrado no separable haría que la decodificación fuera prohibitivamente compleja. En algunas realizaciones, se imponen varias restricciones adicionales a la naturaleza de F . Para la primera restricción, F está restringido para que sea separable, de modo que el filtrado pueda implementarse como convoluciones de derivación de w horizontales y verticales separables. Para la segunda restricción, cada uno de los filtros horizontal y vertical está restringido para que sea simétrico. Para la tercera restricción, se supone que la suma de los coeficientes de filtro horizontal y vertical suma 1.

El filtrado autoguiado dual con proyección subespacial puede ser uno de los esquemas de restauración en bucle. El filtrado guiado es una técnica de filtrado de imágenes donde se muestra un modelo lineal local mediante la ecuación (15):

$$y = Fx + G \quad \text{Ec. (15)}$$

se utiliza para calcular la salida filtrada y a partir de una muestra x sin filtrar, donde F y G se determinan basándose en las estadísticas de la imagen degradada y de una imagen de guía en las proximidades del píxel filtrado. Si la imagen guía es la misma que la imagen degradada, el denominado filtrado autoguiado resultante tiene el efecto de suavizar los bordes. En un ejemplo, se puede utilizar una forma específica de filtrado autoguiado. La forma específica de filtrado autoguiado depende de dos parámetros: un radio r y un parámetro de ruido e , y se enumera de la siguiente manera:

1. Obtener la media μ y la varianza σ^2 de píxeles en una ventana de $(2r + 1) \times (2r + 1)$ alrededor de cada píxel. Este paso se puede implementar de manera eficiente con el filtrado de cajas basado en imágenes integrales.

2. Calcular para cada píxel: $f = \sigma^2 / (\sigma^2 + e)$; $g = (1 - f)\mu$

3. Calcular F y G para cada píxel como promedios de valores de g en una ventana de 3×3 alrededor del píxel que se utilizará.

La forma específica del filtro autoguiado está controlada por r y e , donde un r más alto implica una mayor varianza espacial y un e más alto implica una mayor varianza de rango.

La Figura 20 muestra un ejemplo que ilustra la proyección subespacial en algunos ejemplos. Como se muestra en la Figura 20, aunque ninguna de las restauraciones X_1 , X_2 está cerca de la fuente Y , los multiplicadores apropiados $\{\alpha, \beta\}$ pueden acercarlas mucho más a la fuente Y siempre que se muevan un poco en la dirección correcta.

En algunos ejemplos (p. ej., HEVC), se puede utilizar una técnica de filtrado que se denomina desfase adaptativo de muestras (SAO). En algunos ejemplos, el SAO se aplica a la señal de reconstrucción después de un filtro de desbloqueo. El SAO puede utilizar los valores de desfase dados en la cabecera del segmento. En algunos ejemplos, para las muestras de luma, el codificador puede decidir si aplicar (habilitar) el SAO en un segmento. Cuando el SAO está habilitado, la imagen actual permite la separación recursiva de una unidad de codificación en cuatro subregiones y cada subregión puede seleccionar un tipo de SAO entre múltiples tipos de SAO basándose en las características de la subregión.

La Figura 21 muestra una tabla (2100) de una pluralidad de tipos de SAO según una realización de la descripción. En la tabla (2100), se muestran los tipos 0-6 de SAO. Se observa que el tipo 0 de SAO se utiliza para indicar que no hay ninguna aplicación de SAO. Además, cada tipo de SAO de tipo 1 de SAO al tipo 6 de SAO incluye múltiples categorías. El SAO puede clasificar los píxeles reconstruidos de una subregión en categorías y reducir la distorsión añadiendo un desfase a los píxeles de cada categoría de la subregión. En algunos ejemplos, las propiedades de borde se pueden utilizar para la clasificación de píxeles en los tipos 1-4 de SAO, y la intensidad de los píxeles se puede utilizar para la clasificación de píxeles en los tipos 5-6 de SAO.

Específicamente, en una realización, tal como los tipos 5-6 de SAO, el desfase de banda (BO, por sus siglas en inglés) se puede utilizar para clasificar todos los píxeles de una subregión en múltiples bandas. Cada banda de las múltiples bandas incluye píxeles en el mismo intervalo de intensidad. En algunos ejemplos, el rango de intensidad se divide equitativamente en una pluralidad de intervalos, tales como 32 intervalos desde cero hasta el valor de intensidad máximo (p. ej., 255 para píxeles de 8 bits), y cada intervalo está asociado a un desplazamiento. Además, en un ejemplo, las 32 bandas se dividen en dos grupos, tales como un primer grupo y un segundo grupo. El primer grupo incluye las 16 bandas centrales (p. ej., 16 intervalos que están en el medio del rango de intensidad), mientras que el segundo grupo incluye las 16 bandas restantes (p. ej., 8 intervalos que están en el lado bajo del rango de intensidad y 8 intervalos que están en el lado alto del rango de intensidad). En un ejemplo, solo se transmiten las compensaciones de uno de los dos grupos. En algunas realizaciones, cuando se utiliza la operación de clasificación de píxeles en BO, los cinco bits más significativos de cada píxel se pueden utilizar directamente como índice de banda.

Además, en una realización, tal como los tipos 1-4 de SAO, el desfase de bordes (EO, por sus siglas en inglés) se puede utilizar para la clasificación de píxeles y la determinación de las compensaciones. Por ejemplo, la clasificación de píxeles se puede determinar basándose en patrones unidimensionales de 3 píxeles con la información direccional del borde.

La Figura 22 muestra ejemplos de patrones de 3 píxeles para la clasificación de píxeles en el desfase de bordes en algunos ejemplos. En el ejemplo de la Figura 22, un primer patrón (2210) (como se muestra con 3 píxeles grises) se denomina patrón de 0 grados (la dirección horizontal está asociada con el patrón de 0 grados), un segundo patrón (2220) (como se muestra con 3 píxeles grises) se denomina patrón de 90 grados (la dirección vertical está asociada con el patrón de 90 grados), se hace referencia a un tercer patrón (2230) (como se muestra con 3 píxeles grises) como patrón de 135 grados (la dirección diagonal de 135 grados está asociada con el patrón de 135 grados) y un cuarto patrón (2240) (como se muestra con 3 píxeles grises) se denomina 45 grados patrón (la dirección diagonal de 45 grados está asociada con el patrón de 45 grados). En un ejemplo, uno de los cuatro patrones direccionales mostrados en la Figura 22 puede seleccionarse teniendo en cuenta la información direccional del borde para una subregión. La selección puede enviarse en el flujo de bits de vídeo codificado como información secundaria en un ejemplo. A continuación, los píxeles en la subregión se pueden clasificar en múltiples categorías comparando cada píxel con sus dos píxeles vecinos en la dirección asociada con el patrón direccional.

La Figura 23 muestra una tabla (2300) para la regla de clasificación de píxeles para el desfase de bordes en algunos ejemplos. Específicamente, un píxel c (que también se muestra en cada patrón de la Figura 22) se compara con dos píxeles vecinos (también se muestran en color gris en cada patrón de la Figura 22), y el píxel c se puede clasificar en uno de la categoría 0-4 basándose en la comparación según la regla de clasificación de píxeles mostrada en la Figura 23.

En algunas realizaciones, el SAO en el lado del decodificador se puede operar independientemente de la unidad de codificación más grande (LCU) (ej., CTU), de modo que se puedan guardar las memorias intermedias de línea. En algunos ejemplos, los píxeles de las filas superior e inferior de cada LCU no se procesan en SAO cuando se eligen los patrones de clasificación de 90 grados, 135 grados, y 45 grados; los píxeles de las columnas situadas más a la izquierda y a la derecha en cada LCU no se procesan en SAO cuando se eligen los patrones de 0 grados, 135 grados, y 45 grados.

La Figura 24 muestra un ejemplo (2400) de sintaxis que puede ser necesario señalar para una CTU si los parámetros no se fusionan desde la CTU vecina. Por ejemplo, un elemento sintáctico `sao_type_idx[cldx][rx][ry]` puede señalizarse para indicar el tipo de SAO de una subregión. El tipo de SAO puede ser BO (desfase de banda) o EO (desfase de bordes). Cuando `sao_type_idx[cldx][rx][ry]` tiene un valor de 0, indica que SAO está DESACTIVADO; un valor de uno a cuatro indica que se utiliza una de las cuatro categorías de EO correspondientes a 0°, 90°, 135° y 45°, y un valor de cinco indica que se utiliza BO. En el ejemplo de la Figura 24, cada uno de los tipos BO y EO tiene cuatro valores de desfase de SAO (`sao_offset[cldx][rx][ry][0]` para `sao_offset[cldx][rx][ry][3]`) que se señalizan.

Como se muestra en la Figura 24, el SAO es un proceso de filtrado que requiere sobrecargas de señalización adicionales para indicar toda la información relacionada que se requiere para realizar el filtrado de SAO, como se muestra en la Figura 24, lo que puede limitar el potencial del SAO.

Generalmente, un proceso de filtrado puede utilizar las muestras reconstruidas de un primer componente de color como entrada (p. ej., Y o Cb o Cr, o R o G o B) para generar una salida, y la salida del proceso de filtrado se aplica a un segundo componente de color que puede ser el mismo que el primer componente de color o que puede ser otro componente de color que es diferente del primer componente de color.

En un ejemplo relacionado de filtrado de componentes cruzados (CCF, por sus siglas en inglés), los coeficientes de filtro se derivan basándose en algunas ecuaciones matemáticas. Los coeficientes de filtro derivados se señalizan desde el lado del codificador al lado del decodificador, y los coeficientes de filtro derivados se utilizan para generar desplazamientos utilizando combinaciones lineales. A continuación, las compensaciones generadas se añaden a las muestras reconstruidas como un proceso de filtrado. Por ejemplo, las compensaciones se generan basándose en combinaciones lineales de los coeficientes de filtrado con muestras de luma, y las compensaciones generadas se añaden a las muestras de croma reconstruidas. El ejemplo relacionado de CCF se basa en la suposición de una relación de mapeo lineal entre los valores de la muestra de luma reconstruida y los valores delta entre las muestras de croma originales y reconstruidas. Sin embargo, el mapeo entre los valores de la muestra de luma reconstruida y los valores delta entre las muestras de croma originales y reconstruidas no sigue necesariamente un proceso de mapeo lineal, por lo tanto, el rendimiento de codificación de CCF puede estar limitado bajo el supuesto de una relación de mapeo lineal.

Los aspectos de la presente descripción proporcionan técnicas de mapeo no lineal que pueden utilizarse en el filtrado de componentes cruzados y/o en el filtrado de componentes del mismo color sin sobrecargas de señalización significativas. En un ejemplo, las técnicas de mapeo no lineal se pueden utilizar en el filtrado de componentes cruzados para generar un desplazamiento de muestra de componentes cruzados. En otro ejemplo, las técnicas de mapeo no lineal se pueden utilizar en el mismo filtrado de componentes de color para generar un desplazamiento de muestra local.

Por conveniencia, un proceso de filtrado que utiliza las técnicas de mapeo no lineal puede denominarse desfase de muestra mediante mapeo no lineal (SO-NLM, por sus siglas en inglés). El SO-NLM en el proceso de filtrado de componentes cruzados se puede denominar desfase de muestra de componentes cruzados (CCSO, por sus siglas en inglés). El SO-NLM en el mismo filtrado de componentes de color se puede denominar desfase de muestra local (LSO, por sus siglas en inglés).

En un ejemplo, el CCSO y el LSO se pueden utilizar como filtrado en bucle para reducir la distorsión de las muestras reconstruidas, el CCSO y el LSO no se basan en la suposición de mapeo lineal utilizada en el CCF ilustrativo relacionado. Por ejemplo, el CCSO no se basa en la suposición de una relación de mapeo lineal entre los valores de muestra reconstruida por luma y los valores delta entre las muestras de croma originales y las muestras reconstruidas por croma. Similarmente, el LSO no se basa en la suposición de una relación de mapeo lineal entre los valores de muestra reconstruida de un componente de color y los valores delta entre las muestras originales del componente de color y las muestras reconstruidas del componente de color.

En la siguiente descripción, se describe el proceso de filtrado SO-NLM que utiliza las muestras reconstruidas de un primer componente de color como entrada (p. ej., Y o Cb o Cr, o R o G o B) para generar una salida, y la salida del proceso de filtrado se aplica a un segundo componente de color. Cuando el segundo componente de color es el mismo componente de color que el primer componente de color, la descripción es aplicable para LSO; y cuando el segundo componente de color es diferente del primer componente de color, la descripción es aplicable para CCSO.

En SO-NLM, se obtiene un mapeo no lineal en el lado del codificador. Un mapeo no lineal se produce entre muestras reconstruidas de un primer componente de color en la región de soporte de filtro y las compensaciones que se añadirán a un segundo componente de color en la región de soporte de filtro. Cuando el segundo componente de color es el mismo que el primer componente de color, el mapeo no lineal se utiliza en LSO; cuando el segundo componente de color es diferente del primer componente de color, el mapeo no lineal se utiliza en el CCSO. El dominio del mapeo no lineal está determinado por diferentes combinaciones de muestras reconstruidas de entrada procesadas (también denominadas combinaciones de posibles valores de muestra reconstruida).

Las técnicas de SO-NLM se pueden ilustrar utilizando un ejemplo específico. En el ejemplo específico, se determinan las muestras reconstruidas de un primer componente de color ubicado en un área de soporte de filtro (también denominada "región de soporte de filtro"). El área de soporte de filtro es un área dentro de la cual se puede aplicar el filtro, y el área de soporte de filtro puede tener cualquier forma adecuada.

La Figura 25 muestra un ejemplo de un área (2500) de soporte de filtro según algunas realizaciones de la descripción. El área (2500) de soporte de filtro incluye cuatro muestras reconstruidas: P0, P1, P2 y P3 de un primer componente de color. En el ejemplo de la Figura 25, las cuatro muestras reconstruidas pueden formar una forma de cruz en la dirección vertical y la dirección horizontal, y la ubicación central de la forma de cruz es la ubicación de la muestra a filtrar. Una muestra en la ubicación central y del mismo componente de color que P0-P3 se indica con C. Una muestra en la ubicación central y de un segundo componente de color se indica con F. El segundo componente de color puede ser el mismo que el primer componente de color de P0-P3 o puede ser diferente del primer componente de color de P0-P3.

La Figura 26 muestra un ejemplo de otra área (2600) de soporte de filtro según algunas realizaciones de la descripción. El área (2600) de soporte de filtro incluye cuatro muestras reconstruidas P0, P1, P2 y P3 de un primer componente de color que forman una forma cuadrada. En el ejemplo de la Figura 26, la ubicación central de la forma cuadrada es la ubicación de la muestra que se filtrará. Una muestra en la ubicación central y del mismo componente de color que P0-P3 se indica con C. Una muestra en la ubicación central y de un segundo componente de color se indica con F. El segundo componente de color puede ser el mismo que el primer componente de color de P0-P3 o puede ser diferente del primer componente de color de P0-P3.

A continuación, las muestras reconstruidas se procesan adecuadamente para formar derivaciones de filtro. En el ejemplo específico, las muestras reconstruidas se procesan en los dos pasos siguientes.

En un primer paso, se calculan los valores delta, respectivamente, entre P0-P3 y C. Por ejemplo, m0 indica el valor delta entre P0 y C; m1 indica el valor delta entre P1 y C; m2 indica el valor delta entre P2 y C; m3 indica el valor delta entre P3 y C.

En un segundo paso, los valores delta m0-m3 se cuantifican adicionalmente, los valores cuantificados se indican como d0, d1, d2, d3. En un ejemplo, el valor cuantificado puede ser uno de -1, 0, 1 basándose en un proceso de cuantificación. Por ejemplo, un valor m se puede cuantificar a -1 cuando m es menor que -N (N es un valor positivo y se denomina tamaño de paso de cuantificación); el valor m se puede cuantificar a 0 cuando m está en un rango de [-N, N]; y el valor m se puede cuantificar a 1 cuando m es mayor que N. En algunos ejemplos, el tamaño del paso de cuantificación N puede ser uno de 4, 8, 12, 16 y similares.

En algunas realizaciones, los valores cuantificados d0-d3 son derivaciones de filtro y se pueden utilizar para identificar una combinación en el dominio de filtro. Por ejemplo, las derivaciones d0-d3 de filtro pueden formar una combinación en el dominio del filtro. Cada derivación de filtro puede tener tres valores cuantificados, por lo tanto, cuando se utilizan cuatro tomas de filtro, el dominio de filtro incluye 81 (3 x 3 x 3 x 3) combinaciones.

Las Figuras 27A-27C muestran una tabla (2700) que tiene 81 combinaciones según una realización de la descripción. La tabla (2700) incluye 81 filas que corresponden a 81 combinaciones. En cada fila correspondiente a una combinación, la primera columna incluye un índice de las combinaciones; la segunda columna incluye el valor de la derivación d0 de filtro para la combinación; la tercera columna incluye el valor de la derivación d1 de filtro para la combinación; la cuarta columna incluye el valor de la derivación d2 de filtro para la combinación; la quinta columna incluye el valor de la derivación d3 de filtro para la combinación; la columna seis incluye el valor del desfase asociado a la combinación para el mapeo no lineal. En un ejemplo, cuando se determinan las derivaciones d0-d3 de filtro, el valor de desfase (indicado por s) asociado a la combinación de d0-d3 se puede determinar según la tabla (2700). En un ejemplo, los valores s0-s80 de desfase son números enteros, tales como 0, 1, -1, 3, -3, 5, -5, -7, y similares.

En algunas realizaciones, el proceso de filtrado final de SO-NLM se puede aplicar como se muestra en la ecuación. (16):

$$f' = \text{recortar}(F + s) \quad \text{Ec. (16)}$$

donde F es la muestra reconstruida del segundo componente de color que se filtrará, y s es el valor de desfase determinado según las derivaciones de filtro que procesan los resultados de las muestras reconstruidas del primer componente de color, tal como, utilizando la tabla (2700). La suma de la muestra reconstruida F y el valor de desfase s se recortan adicionalmente en el rango asociado con la profundidad de bits para determinar la muestra filtrada final f del segundo componente de color.

Se observa que, en el caso de LSO, el segundo componente de color de la descripción anterior es el mismo que el primer componente de color y, en el caso de CCSO, el segundo componente de color de la descripción anterior puede ser diferente del primer componente de color.

Se observa que la descripción anterior se puede ajustar para otras realizaciones de la presente descripción.

Según un aspecto de la descripción, en el lado del codificador, el dispositivo de codificación puede derivar un mapeo entre las muestras reconstruidas de un primer componente de color en una región de soporte de filtro y las compensaciones que se añadirán a las muestras reconstruidas de un segundo componente de color. El mapeo puede ser cualquier mapeo lineal o no lineal adecuado. A continuación, el proceso de filtrado se puede aplicar en el lado del codificador y/o en el lado del decodificador basándose en el mapeo. Por ejemplo, el mapeo se informa adecuadamente al decodificador (p. ej., el mapeo se incluye en un flujo de bits de vídeo codificado que se transmite desde el lado del codificador al lado del decodificador) y, a continuación, el decodificador puede realizar el proceso de filtrado basándose en el mapeo.

Según algunos aspectos de la descripción, el mapeo es un mapeo no lineal entre las muestras reconstruidas de un primer componente de color y el desfase que se aplicará a las muestras reconstruidas de un segundo componente de color. En algunos ejemplos, para realizar un proceso de filtrado en una región de soporte de filtro, las muestras reconstruidas del primer componente de color en la región de soporte de filtro pueden introducirse en el proceso de filtrado, y el desfase es la salida del proceso de filtrado. El desfase se puede aplicar a una muestra reconstruida del segundo componente de color en la región de soporte de filtro.

En algunas realizaciones, se utiliza una tabla de búsqueda para describir el mapeo. La tabla de consulta puede asociar la entrada del proceso de filtrado (muestras reconstruidas del primer componente de color) con la salida del proceso de filtrado (compensaciones que se aplicarán a las muestras reconstruidas del segundo componente de color). Por lo tanto, dado un conjunto de valores de muestra reconstruida como entrada, el valor de desfase de salida correspondiente se puede recuperar de la tabla de búsqueda.

En una realización, la entrada son muestras de luma reconstruidas ubicadas en un área (región) de soporte de filtro dada, y la salida se compensa para aplicarse a la muestra reconstruida de croma.

En otra realización, las muestras reconstruidas se cuantifican primero antes de utilizarse como entrada del proceso de filtrado. Por ejemplo, las muestras reconstruidas se pueden cuantificar basándose en los pasos de cuantificación.

En otra realización, las muestras reconstruidas se procesan utilizando operaciones lineales y después se utilizan como entrada del proceso de filtrado. Por ejemplo, se pueden realizar restas de las muestras reconstruidas a un valor (p. ej., una muestra reconstruida central, un promedio de las muestras reconstruidas). Los resultados de las restas se pueden utilizar como entrada del proceso de filtrado.

En otra realización, las muestras reconstruidas se procesan primero utilizan operaciones lineales y la salida de las operaciones lineales se cuantifica adicionalmente y después se utiliza como entrada del proceso de filtrado. En el ejemplo específico descrito anteriormente, el primer paso utiliza las operaciones lineales para calcular los valores delta m0-m3 respectivamente entre P0-P3 y C. A continuación, en el segundo paso, los valores delta m0-m3 se cuantifican adicionalmente a d0-d3.

Según un aspecto de la descripción, el dominio del mapeo se determina mediante diferentes combinaciones de muestras reconstruidas de entrada procesadas (p. ej., procesamiento lineal), tales como combinaciones de P0-P3, combinaciones de m0-m3, combinaciones de d0-d3. En algunos ejemplos, todas las combinaciones posibles de muestras reconstruidas de entrada procesadas se incluyen en el dominio (denominado dominio completo). En un ejemplo, una combinación de muestras reconstruidas procesadas se identifica mediante las diferencias cuantificadas entre las muestras circundantes y la muestra central, tal como una combinación de d0-d3. En otro ejemplo, una combinación de muestras reconstruidas procesadas se identifica mediante las diferencias cuantificadas entre las muestras circundantes. Por ejemplo, n0 es una diferencia entre P0 y P1, n1 es diferente entre P1 y P2, n2 es una

diferencia entre P2 y P3, y n3 es diferente entre P3 y P0. A continuación, n0-n3 se puede cuantificar similarmente a m0-m3 para determinar e0-e3 que se puede utilizar como una combinación de muestras reconstruidas procesadas.

En el ejemplo específico anterior, se utilizan cuatro muestras circundantes (p. ej., mostradas en la Figura 25 o la Figura 26) para identificar cada combinación de muestras reconstruidas procesadas. Por ejemplo, la diferencia m0-m3 de las muestras circundantes P0-P3 con respecto a la muestra central C se puede cuantificar en [-1, 0, 1], basándose en un tamaño de paso de cuantificación N (N es un número entero positivo, tal como 1, 3, 5, 7). Por ejemplo, para cuantificar un valor, cuando el valor es menor que -N, el valor se cuantifica para que sea -1; cuando el valor está en el rango de [-N, N], el valor se cuantifica para que sea 0; y cuando el valor es mayor que N, el valor se cuantifica para que sea 1. Cuando se utilizan cuatro muestras circundantes, el número total de combinaciones es 81 (3⁴), tal como la tabla (2700) mostrada en las Figuras 27A-27C.

Según un aspecto de la descripción, las combinaciones de muestras reconstruidas procesadas dependen de la profundidad de bits interna con la que opera el códec. Por ejemplo, los cálculos de m0-m3 dependen de la profundidad de bits interna, y m0-m3 se cuantifican para determinar d0-d3. Por lo tanto, las combinaciones de d0-d3 pueden depender de la profundidad de bits interna.

Según otro aspecto de la descripción, las combinaciones de muestras reconstruidas procesadas dependen de los parámetros de cuantificación, tales como el tamaño del paso de cuantificación. Por ejemplo, la determinación de d0-d3 en el segundo paso del ejemplo específico se basa en el tamaño del paso de cuantificación N.

Se observa que se puede utilizar cualquier muestra reconstruida adecuada en la región de soporte de filtro para determinar la combinación. En una realización, las muestras reconstruidas ubicadas en el límite de la región de soporte de filtro se denominan muestras circundantes y se pueden utilizar para determinar la combinación.

En otra realización, las muestras reconstruidas en el área de soporte de filtro, que excluye la posición central se denominan muestras circundantes, y se pueden utilizar para determinar la combinación.

En otra realización, las muestras reconstruidas en el área de soporte de filtro, que excluye las muestras cercanas a la posición central, se denominan muestras circundantes y pueden utilizarse para determinar la combinación.

En algunas realizaciones, el dominio del mapeo se determina mediante diferentes combinaciones de muestras reconstruidas procesadas (p. ej., procesamiento lineal) (p. ej., d0-d3). En un ejemplo, un subconjunto predefinido de combinaciones de muestras reconstruidas procesadas (p. ej., d0-d3) se incluye en el dominio (denominado dominio de subconjunto).

La Figura 28 ilustra una tabla (2800) de búsqueda según una realización de la descripción. El dominio del mapeo en la tabla (2800) de búsqueda incluye un subconjunto de todas las combinaciones de d0-d3, tal como 5 combinaciones en la Figura 28.

En algunas realizaciones, las compensaciones generadas a partir del proceso de filtrado, tal como s0-s80 en las Figuras 27A-27C, son números enteros dentro de un rango predefinido [P, Q]. P y Q pueden ser números enteros. Un ejemplo del rango es [-K, K - 1], donde K es un entero positivo. Los valores ilustrativos de K incluyen, aunque no de forma limitativa, 16, 8, 4, 2, 1.

En una realización, los valores de P y Q dependen de la profundidad de bits interna en la que esté operando el códec. En otra realización, los valores de P y Q dependen de los parámetros de cuantificación, tales como el tamaño del paso de cuantificación N. En otra realización, las compensaciones emitidas a partir del proceso de filtrado son un subconjunto predefinido de números enteros. Por ejemplo, el subconjunto predefinido de números enteros incluye números enteros con valores -2^T y/o 2^T (donde T son números enteros), tales como 16, -16, 8, -8, 4, -4, 2, -2, 1, y similares.

Según un aspecto de la descripción, el codificador puede determinar un desfase asociado con una combinación de muestras reconstruidas procesadas en el mapeo basándose en medidas de coste predefinidas. En un ejemplo, para una combinación de muestras reconstruidas procesadas en el primer componente de color, la medida de coste se define como el promedio de la diferencia entre el valor original y los valores de muestra reconstruida de las muestras que se filtrarán en el segundo componente de color.

En los ejemplos de CCSO, las muestras reconstruidas para introducir en el proceso de filtrado provienen de un primer componente de color, y las muestras que se filtrarán provienen de un segundo componente de color que es diferente del primer componente de color. En los ejemplos de LSO, las muestras reconstruidas para introducir en el proceso de filtrado provienen de un primer componente de color, y las muestras que se filtrarán provienen de un segundo componente de color que es el mismo componente de color que el primer componente de color.

En algunas realizaciones, la información de mapeo, tal como 81 desfases respectivamente para 81 combinaciones en las Figuras 27A-27C, se puede señalar desde el lado del codificador al lado del decodificador. En un ejemplo, la

información de mapeo se transporta en el flujo de bits de vídeo codificado, tal como en el conjunto de parámetros de vídeo (VPS), en el conjunto de parámetros de secuencia (SPS), en el conjunto de parámetros de imagen (PPS), en el conjunto de parámetros de adaptación (APS), en la cabecera de segmento, en la cabecera de mosaico y similares.

Según un aspecto de la descripción, el proceso de filtrado determina el valor de desfase basándose en el mapeo y una combinación determinada basándose en muestras reconstruidas del primer componente de color. Por ejemplo, el valor de desfase está asociado a la combinación en el mapeo (p. ej., tabla de búsqueda). A continuación, el valor de desfase se aplica a la muestra que se aplicará en el segundo componente de color. En algunas realizaciones, el recorte se realiza después de añadir el valor de desfase a la muestra que se filtrará para garantizar que el valor de la muestra filtrada no supere los valores máximos o mínimos. En un ejemplo, el recorte se realiza según la ecuación. (17):

$$\text{clip}(x) = \text{recortar}(x) = \max(\min(x, \text{máx_val}), \min_val) \quad \text{Ec. (17)}$$

donde *máx* indica el operador máximo, *mín* indica el operador mínimo, *max_val* indica el valor máximo de la muestra, *min_val* indica los valores mínimos de la muestra, *x* es el valor de la muestra que se recortará.

La Figura 29 muestra un diagrama de flujo que describe un proceso (2900) según una realización de la descripción. El proceso (2900) se puede utilizar para reconstruir un bloque en una imagen de una secuencia de vídeo codificada. El término bloque puede interpretarse como un bloque de predicción, una unidad de codificación, un bloque de luma, un bloque de croma, o similares. En diversas realizaciones, el proceso (2900) se ejecuta mediante un sistema de circuitos de procesamiento, tales como los sistema de circuitos de procesamiento en los dispositivos terminales (310), (320), (330) y (340), los sistemas de circuitos de procesamiento que realizan las funciones del codificador (403) de vídeo, los sistemas de circuitos de procesamiento que realizan las funciones del decodificador (410) de vídeo, los sistemas de circuitos de procesamiento que realizan las funciones del decodificador (510) de vídeo, los sistemas de circuitos de procesamiento que realiza las funciones del codificador (603) de vídeo, y similares. En algunas realizaciones, el proceso (2900) se implementa en instrucciones de software, por lo tanto, cuando el sistema de circuitos de procesamiento ejecuta las instrucciones de software, el sistema de circuitos de procesamiento realiza el proceso (2900). El proceso comienza en (S2901) y procede a (S2910).

En (S2910), se determina una primera combinación a partir de muestras reconstruidas de un primer componente de color dentro de una región de soporte de filtro. En algunas realizaciones, se puede utilizar al menos una operación lineal y una operación de cuantificación para determinar la primera combinación a partir de las muestras reconstruidas del primer componente de color dentro de la región de soporte de filtro.

En una realización, se pueden calcular las diferencias de las muestras reconstruidas del primer componente de color. A continuación, las diferencias se cuantifican para determinar la primera combinación. En algunos ejemplos, que entran dentro del alcance de la invención protegida, se calculan las diferencias entre las muestras reconstruidas circundantes de las muestras reconstruidas a una muestra colocada dentro de las muestras reconstruidas del primer componente de color. La muestra colocada está ubicada en la misma posición que la muestra que se filtrará del segundo componente de color. A continuación, las diferencias se cuantifican basándose en un tamaño de paso de cuantificación para determinar la primera combinación. Se observa que la primera combinación puede depender de la profundidad de bits interna y del parámetro de cuantificación, tal como el tamaño del paso de cuantificación y similares.

Se observa que, en una realización, las muestras reconstruidas circundantes están ubicadas en el límite de la región de soporte de filtro. En otro ejemplo, las muestras reconstruidas circundantes son muestras en la región de soporte de filtro, que excluye la muestra en la posición central de la región de soporte de filtro. En otro ejemplo, las muestras reconstruidas circundantes son muestras en la región de soporte de filtro, que excluye las muestras ubicadas cerca de la posición central de la región de soporte de filtro.

En (S2920), se determina un primer valor de desfase asociado con la primera combinación basándose en un mapeo que asocia los valores de desfase con las combinaciones. En algunas realizaciones, el mapeo es un mapeo no lineal definido en forma de tabla de búsqueda.

En algunas realizaciones, el mapeo asocia los valores de desfase con las combinaciones de un dominio completo para las muestras reconstruidas del primer componente de color dentro de la región de soporte de filtro. El dominio completo del mapeo incluye todas las combinaciones posibles. Por ejemplo, cuando cuatro muestras reconstruidas se procesan y cuantifican cada una a uno de los tres valores, el número de todas las combinaciones posibles es 81 (3⁴). En algunas realizaciones, el mapeo asocia los valores de desfase con las combinaciones de un dominio de subconjunto para las muestras reconstruidas del primer componente de color dentro de la región de soporte de filtro. El dominio del subconjunto incluye un subconjunto de combinaciones del dominio completo.

En algunas realizaciones, el mapeo se realiza en al menos uno de entre un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de adaptación (APS), una cabecera de segmento, y una cabecera de mosaico de un flujo de bits de vídeo codificado.

En algunas realizaciones, los valores de desfase del mapeo se derivan basándose en las medidas de coste. A continuación, el mapeo puede incluirse en al menos uno de un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de adaptación (APS), una cabecera de segmento, y una cabecera de mosaico de un flujo de bits de vídeo codificado.

En (S2930), el primer valor de desfase se aplica a una muestra que se aplicará de un segundo componente de color en la región de soporte de filtro para determinar una muestra filtrada del segundo componente de color. En un ejemplo, el segundo componente de color es el primer componente de color. En otro ejemplo, el segundo componente de color es un componente de color diferente del primer componente de color. Por ejemplo, el primer componente de color es un componente de luma, y el segundo componente de color es un componente de croma.

En algunos ejemplos, el primer valor de desfase se añade a la muestra que se aplicará del segundo componente de color para determinar una muestra filtrada intermedia. A continuación, la muestra filtrada intermedia se recorta basándose en un valor máximo y un valor mínimo para determinar la muestra filtrada del segundo componente de color.

El proceso (2900) pasa a (S2999) y termina.

El proceso (2900) se puede adaptar adecuadamente. Lo(s) paso(s) del proceso (2900) puede(n) modificarse y/u omitirse. Sin embargo, en las realizaciones de la invención protegida, deben realizarse todos los pasos del proceso tal como se define en la reivindicación 1 adjunta.

Se pueden añadir pasos adicionales. Se puede utilizar cualquier orden de implementación adecuado.

Las realizaciones en la descripción pueden utilizarse por separado o combinarse en cualquier orden. Además, cada uno de los métodos (realizaciones), un codificador, y un decodificador puede implementarse mediante sistemas de circuitos de procesamiento (p. ej., uno o más procesadores o uno o más circuitos integrados). En un ejemplo, el uno o más procesadores ejecutan un programa que se almacena en un medio legible por ordenador no transitorio.

Las técnicas descritas anteriormente, pueden implementarse como software de ordenador usando instrucciones legibles por ordenador y almacenarse físicamente en uno o más medios legibles por ordenador. Por ejemplo, la Figura 30 muestra un sistema informático (3000) adecuado para implementar ciertas realizaciones de la materia objeto divulgada.

El software informático puede codificarse usando cualquier código de máquina o lenguaje informático adecuado, que puede estar sujeto a ensamblaje, compilación, vinculación o mecanismos similares para crear código que comprende instrucciones que pueden ejecutarse directamente, o por interpretación, ejecución de microcódigo y similares, por unidades centrales de procesamiento (CPU), unidades de procesamiento gráfico (GPU) y similares.

Las instrucciones pueden ejecutarse en diversos tipos de ordenadores o componentes de los mismos, que incluyen, por ejemplo, ordenadores personales, tabletas, servidores, teléfonos inteligentes, dispositivos de juego, internet de los dispositivos de cosas y similares.

Los componentes mostrados en la Figura 30 para el sistema informático (3000) son de naturaleza ilustrativa y no pretenden sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad de las realizaciones de implementación de software informático. Tampoco debe interpretarse que la configuración de componentes tiene cualquier dependencia o requisito relacionado con uno cualquiera o una combinación de componentes ilustrados en la realización ilustrativa de un sistema informático (3000).

El sistema informático (3000) puede incluir ciertos dispositivos de entrada de interfaz humana. Dicho dispositivo de entrada de la interfaz humana puede ser sensible a la entrada por uno o más usuarios humanos a través de, por ejemplo, entrada táctil (tales como: pulsaciones, deslizamientos, movimientos del guante de datos), entrada de audio (tales como: voz, bloqueo), entrada visual (tales como: gestos), entrada olfativa (no representada). Los dispositivos de interfaz humana también se pueden usar para capturar ciertos medios no necesariamente relacionados directamente con la entrada consciente por un ser humano, tal como audio (tales como: voz, música, sonido ambiental), imágenes (tales como: imágenes escaneadas, imágenes fotográficas obtenidas de una cámara de imagen fija), vídeo (tal como vídeo bidimensional, vídeo tridimensional que incluye vídeo estereoscópico).

Los dispositivos de interfaz humana de entrada pueden incluir uno o más de (solo uno de cada uno de los representados): teclado (3001), ratón (3002), panel táctil (3003), pantalla táctil (3010), guante de datos (no mostrado), palanca de mando o joystick (3005), micrófono (3006), escáner (3007), cámara (3008).

El sistema informático (3000) también puede incluir ciertos dispositivos de salida de interfaz humana. Dichos dispositivos de salida de la interfaz humana pueden estimular los sentidos de uno o más usuarios humanos a través de, por ejemplo, salida táctil, sonido, luz y olor/sabor. Tales dispositivos de salida de interfaz humana pueden incluir

dispositivos de salida táctiles (por ejemplo, retroalimentación táctil mediante la pantalla táctil (3010), guante de datos (no mostrado) o joystick (3005), pero también puede haber dispositivos de retroalimentación táctil que no sirven. como dispositivos de entrada), dispositivos de salida de audio (tales como: altavoces (3009), auriculares (no representados)), dispositivos de salida visual (tales como pantallas (3010) para incluir pantallas CRT, pantallas LCD, pantallas de plasma, pantallas OLED, cada una con o sin capacidad de entrada de pantalla táctil, cada uno con o sin capacidad de retroalimentación táctil, algunos de los cuales pueden ser capaces de generar una salida visual bidimensional o una salida más de tridimensional a través de medios tales como una salida estereográfica; gafas de realidad virtual (no representadas), pantallas holográficas y tanques de humo (no representados), e impresoras (no representadas).

El sistema informático (3000) también puede incluir dispositivos de almacenamiento accesibles para humanos y sus medios asociados tales como medios ópticos que incluyen CD/DVD ROM/RW (3020) con CD/DVD o medios (3021) similares, memoria USB (3022), disco duro o unidad (3023) de estado sólido extraíble, medios magnéticos heredados tales como cintas y disquetes (no representados), dispositivos especializados basados en ROM/ASIC/PLD tales como dongles de seguridad (no representados), y similares.

Los expertos en la técnica también deben comprender que el término “medios legibles por ordenador” como se usa en relación con la materia objeto actualmente descrita no abarca medios de transmisión, ondas portadoras u otras señales transitorias.

El sistema informático (3000) también puede incluir una interfaz (3054) a una o más redes (3055) de comunicación. Las redes pueden ser, por ejemplo, inalámbricas, cableadas, ópticas. Las redes pueden ser, además, locales, de gran área, metropolitana, vehicular e industriales, en tiempo real, tolerante a la demora, etc. Ejemplos de redes incluyen redes de área local como Ethernet, LAN inalámbricas, redes celulares que incluyen GSM, 3G, 4G, 5G, LTE y similares, redes de TV por cable o redes digitales inalámbricas de área amplia que incluyen TV por cable, TV satelital y transmisión terrestre. TV, vehicular e industrial para incluir CANBus, etc. Ciertas redes comúnmente requieren adaptadores de interfaz de red externos que se conectan a ciertos puertos de datos de propósito general o buses periféricos (3049) (tales como, por ejemplo, puertos USB del sistema informático (3000)); otras se integran comúnmente en el núcleo del sistema informático (3000) mediante la unión a un bus del sistema como se describe a continuación (por ejemplo, interfaz Ethernet en un sistema informático PC o interfaz de red celular en un sistema informático inteligente). Utilizando cualquiera de estas redes, el sistema informático (3000) puede comunicarse con otras entidades. Dicha comunicación puede ser unidireccional, recibir solo (por ejemplo, difusión de difusión), solo envío unidireccional (por ejemplo, NANbus a ciertos dispositivos de CANbus), o bidireccional, por ejemplo, a otros sistemas informáticos que usan redes digitales de área local o amplia. Se pueden usar ciertos protocolos y pilas de protocolos en cada una de esas redes e interfaces de red como se describió anteriormente.

Los dispositivos de interfaz humana anteriormente mencionados, dispositivos de almacenamiento accesibles para humanos e interfaces de red pueden unirse a un núcleo (3040) del sistema informático (3000).

El núcleo (3040) puede incluir una o más unidades (3041) centrales de procesamiento (CPU), unidades (3042) de procesamiento de gráficos (GPU), unidades de procesamiento programables especializadas en forma de áreas (3043) de puerta programables en campo (FPGA), hardware aceleradores para determinadas tareas (3044), adaptador (3050) de gráficos, y etc. Estos dispositivos, junto con la memoria (3045) de solo lectura (ROM), la memoria (3046) de acceso aleatorio, el almacenamiento masivo interno tal como discos duros internos no accesibles para el usuario, SSD, y similares (3047), pueden conectarse a través de un bus (3048) de sistema. En algunos sistemas informáticos, el bus (3048) de sistema puede ser accesible en forma de uno o más tapones físicos para habilitar extensiones por CPU adicionales, GPU y similares. Los dispositivos periféricos se pueden unir directamente al bus (3048) de sistema del núcleo, o a través de un bus periférico (3049). En un ejemplo, se puede conectar una pantalla (3010) al adaptador (3050) de gráficos. Las arquitecturas para un bus periférico incluyen PCI, USB y similares.

Las CPU (3041), las GPU (3042), las FPGA (3043) y aceleradores (3044) pueden ejecutar ciertas instrucciones que, en combinación, pueden constituir el código informático mencionado anteriormente. Ese código informático puede almacenarse en la ROM (3045) o RAM (3046). Los datos transitorios también pueden almacenarse en la RAM (3046), mientras que los datos permanentes pueden almacenarse, por ejemplo, en el almacenamiento (3047) masivo interno. El almacenamiento rápido y la recuperación a cualquiera de los dispositivos de memoria pueden habilitarse a través del uso de memoria caché, que puede asociarse estrechamente con una o más CPU (3041), GPU (3042), almacenamiento masivo (3047), ROM (3045), RAM (3046) y similares.

Los medios legibles por ordenador pueden tener código informático en el mismo para realizar diversas operaciones implementadas por ordenador. Los medios y el código informático pueden ser aquellos especialmente diseñados y contruidos para los fines de las realizaciones, o pueden ser del tipo bien conocido y disponible para aquellos que tienen habilidad en las técnicas de software informático.

Como ejemplo y no a modo de limitación, el sistema informático tiene arquitectura (3000) y, específicamente, el núcleo (3040) puede proporcionar funcionalidad como resultado del (de los) procesador(es) (incluidas CPU, GPU, FPGA, aceleradores, y similares) que ejecuta software incorporado en uno o más medios tangibles legibles por ordenador. Tales medios legibles por ordenador pueden ser medios asociados con el almacenamiento masivo accesible por el

usuario como se ha introducido anteriormente, así como un cierto almacenamiento del núcleo (3040) que son de naturaleza no transitoria, tal como el almacenamiento (3047) masivo interno en el núcleo o ROM (3045). El software que implementa diversas realizaciones de la presente descripción puede almacenarse en dichos dispositivos y ejecutarse mediante el núcleo (3040). Un medio legible por ordenador puede incluir uno o más dispositivos o chips de memoria, según las necesidades particulares. El software puede hacer que el núcleo (3040) y específicamente los procesadores en el mismo (incluidas CPU, GPU, FPGA y similares) ejecuten procesos particulares o partes particulares de procesos particulares descritos en la presente memoria, incluyendo definir estructuras de datos almacenadas en RAM (3046) y modificar tales estructuras de datos según los procesos definidos por el software. Además o como alternativa, el sistema informático puede proporcionar funcionalidad como resultado de una lógica cableada o incorporarse de otro modo en un circuito (por ejemplo: acelerador (3044)), que puede operar en lugar de o junto con software para ejecutar procesos particulares o partes particulares de procesos particulares descritos en la presente memoria. La referencia al software puede abarcar lógica y viceversa, cuando sea apropiado. La referencia a un medio legible por ordenador puede abarcar un circuito (tal como un circuito integrado (IC) que almacena software para su ejecución, un circuito que incorpora lógica para su ejecución, o ambos, cuando sea apropiado. La presente descripción abarca cualquier combinación adecuada de hardware y software.

Apéndice A: Acrónimos:

JEM: modelo de exploración conjunta

VVC: codificación de vídeo versátil

BS: conjunto de referencia

MV: vector de movimiento

HEVC: codificación de vídeo de alta eficiencia

MPM: modo más probable

WAIP: Intrapredicción gran angular

SEI: información de mejora suplementaria

VUI: información de usabilidad de vídeo

GOP: grupos de imágenes

TU: Unidades de transformación,

PU: Unidades de predicción

CTU: unidades de árbol de codificación

CTB: bloques de árbol de codificación

PB: bloques de predicción

HRD: decodificador de referencia hipotético

SDR: rango dinámico estándar

SNR: relación ruido señal

CPU: unidades de procesamiento central

GPU: unidades de procesamiento gráfico

CRT: tubo de rayos catódicos

LCD: pantalla de cristal líquido

OLED: diodo emisor de luz orgánico

CD: disco compacto

DVD: disco de vídeo Digital

ROM: memoria de solo lectura

5 RAM: memoria de acceso aleatorio

ASIC: circuito integrado específico de aplicación

PLD: dispositivo lógico programable

10

LAN: Red de área local

GSM: sistema global para comunicaciones móviles

15

LTE: Evolución a Largo Plazo

CANBus: Bus de red de área de controlador

USB: USB: bus de serie universal

20

PCI: interconexión de componentes periféricos

FPGA: áreas de puertas programables en campo

25

SSD: unidad de estado sólido

IC: circuito integrado

CU: unidad de codificación

30

PDPC: Combinación de predicción dependiente de la posición

ISP: Intra subparticiones

35

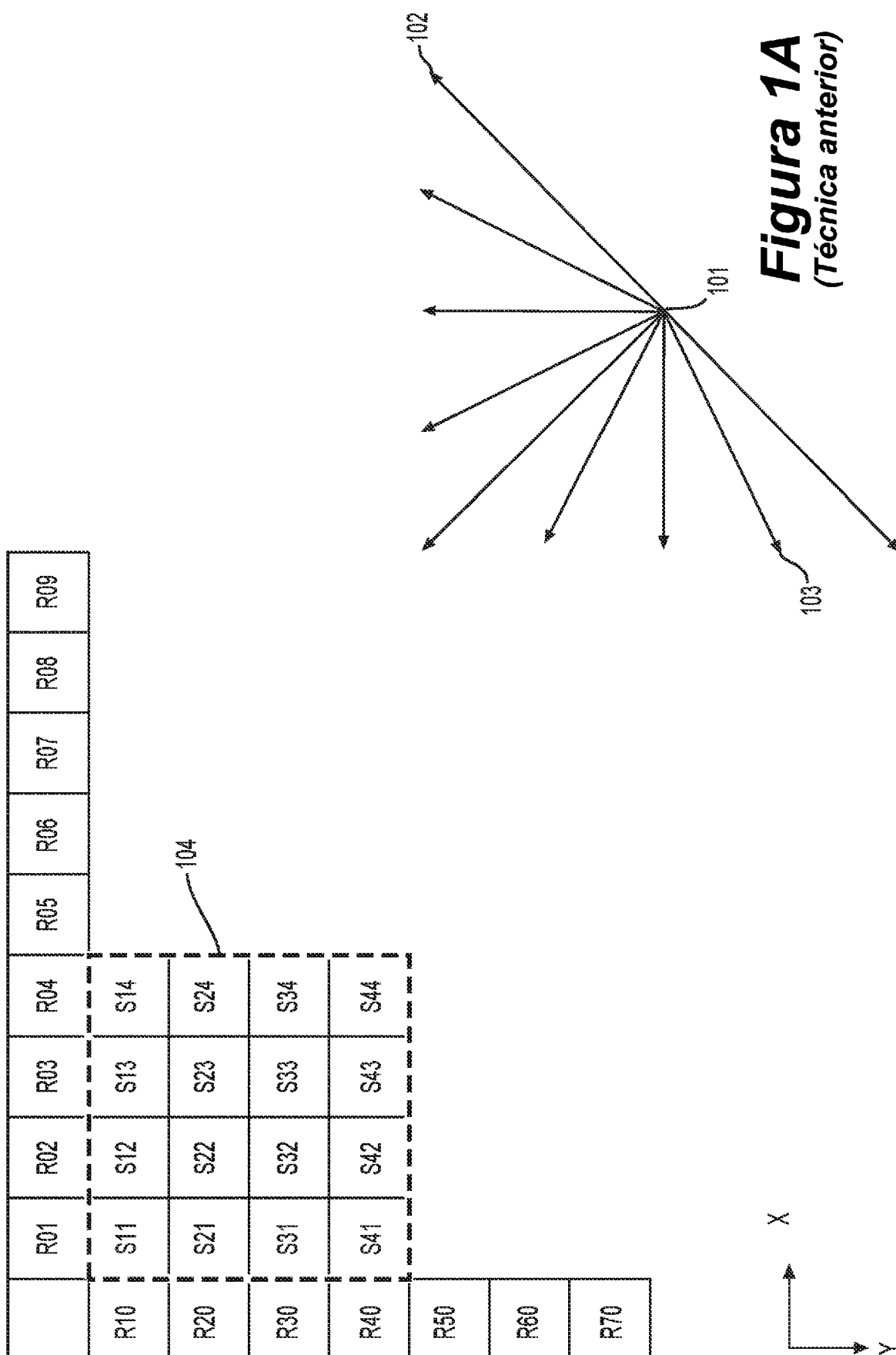
SPS: Configuración de parámetros de secuencia

Si bien esta divulgación ha descrito varias realizaciones ilustrativas, existen alteraciones, permutaciones, y varios equivalentes sustitutivos y, por lo tanto, se apreciará que los expertos en la técnica podrán diseñar numerosos sistemas y métodos basándose en esta descripción.

40

REIVINDICACIONES

1. Un método para filtrar en la decodificación de vídeo, que comprende:
5 determinar, mediante un procesador, una primera combinación a partir de muestras reconstruidas de un primer componente de color dentro de una región de soporte de filtro;
 determinar, mediante el procesador y basándose en un mapeo que asocia valores de desfase con combinaciones de posibles valores de muestras reconstruidas, un primer valor de desfase que está asociado con la primera combinación; y
10 aplicar, mediante el procesador, el primer valor de desfase a una muestra que se filtrará de un segundo componente de color en la región de soporte de filtro para determinar una muestra filtrada del segundo componente de color,
 caracterizado por
15 calcular las diferencias entre las muestras reconstruidas circundantes de las muestras reconstruidas para una muestra colocada dentro de las muestras reconstruidas del primer componente de color, estando la muestra colocada ubicada en la misma posición que la muestra que se filtrará del segundo componente de color; y
 cuantificar las diferencias basándose en un tamaño de paso de cuantificación para determinar la primera combinación.
20
2. El método de la reivindicación 1, en donde el mapeo es un mapeo no lineal definido en forma de una tabla de búsqueda.
3. El método de la reivindicación 1, en donde el mapeo asocia los valores de desfase con las combinaciones de un dominio completo para las muestras reconstruidas del primer componente de color dentro de la región de soporte de filtro, todas las combinaciones posibles de muestras reconstruidas de entrada procesadas se incluyen en el dominio completo.
25
4. El método de la reivindicación 1, en donde el mapeo asocia los valores de desfase con las combinaciones de un dominio de subconjunto para las muestras reconstruidas del primer componente de color dentro de la región de soporte de filtro, un subconjunto predefinido de combinaciones de muestras reconstruidas procesadas se incluye en el dominio.
30
5. El método de la reivindicación 1, que comprende además:
35 recibir el mapeo que se transporta en al menos uno de un conjunto de parámetros de vídeo, VPS, un conjunto de parámetros de secuencia, SPS, un conjunto de parámetros de imagen, PPS, un conjunto de parámetros de adaptación, APS, una cabecera de segmento y una cabecera de mosaico de un flujo de bits de vídeo codificado.
- 40 6. El método de la reivindicación 1, que comprende además:
 derivar, durante una operación de codificación de vídeo, los valores de desfase del mapeo basándose en las medidas de coste; e
 incluir el mapeo en al menos uno de un conjunto de parámetros de vídeo, VPS, un conjunto de parámetros de secuencia, SPS, un conjunto de parámetros de imagen, PPS, un conjunto de parámetros de adaptación, APS, una cabecera de segmento y una cabecera de mosaico de un flujo de bits de vídeo codificado.
45
- 50 7. El método de la reivindicación 1, que comprende además:
 añadir el primer valor de desfase con la muestra que se filtrará del segundo componente de color para determinar una muestra filtrada intermedia; y
 recortar la muestra filtrada intermedia basándose en un valor máximo y un valor mínimo para determinar la muestra filtrada del segundo componente de color.
55
8. Un aparato para filtrar en decodificación de vídeo, que comprende:
 un sistema de circuitos de procesamiento configurado para realizar el método según una cualquiera de las reivindicaciones 1 a 7.
- 60 9. Un medio informático no transitorio que almacena instrucciones, en donde las instrucciones, cuando las ejecuta un procesador, hacen que el procesador realice el método según una cualquiera de las reivindicaciones 1 a 7.



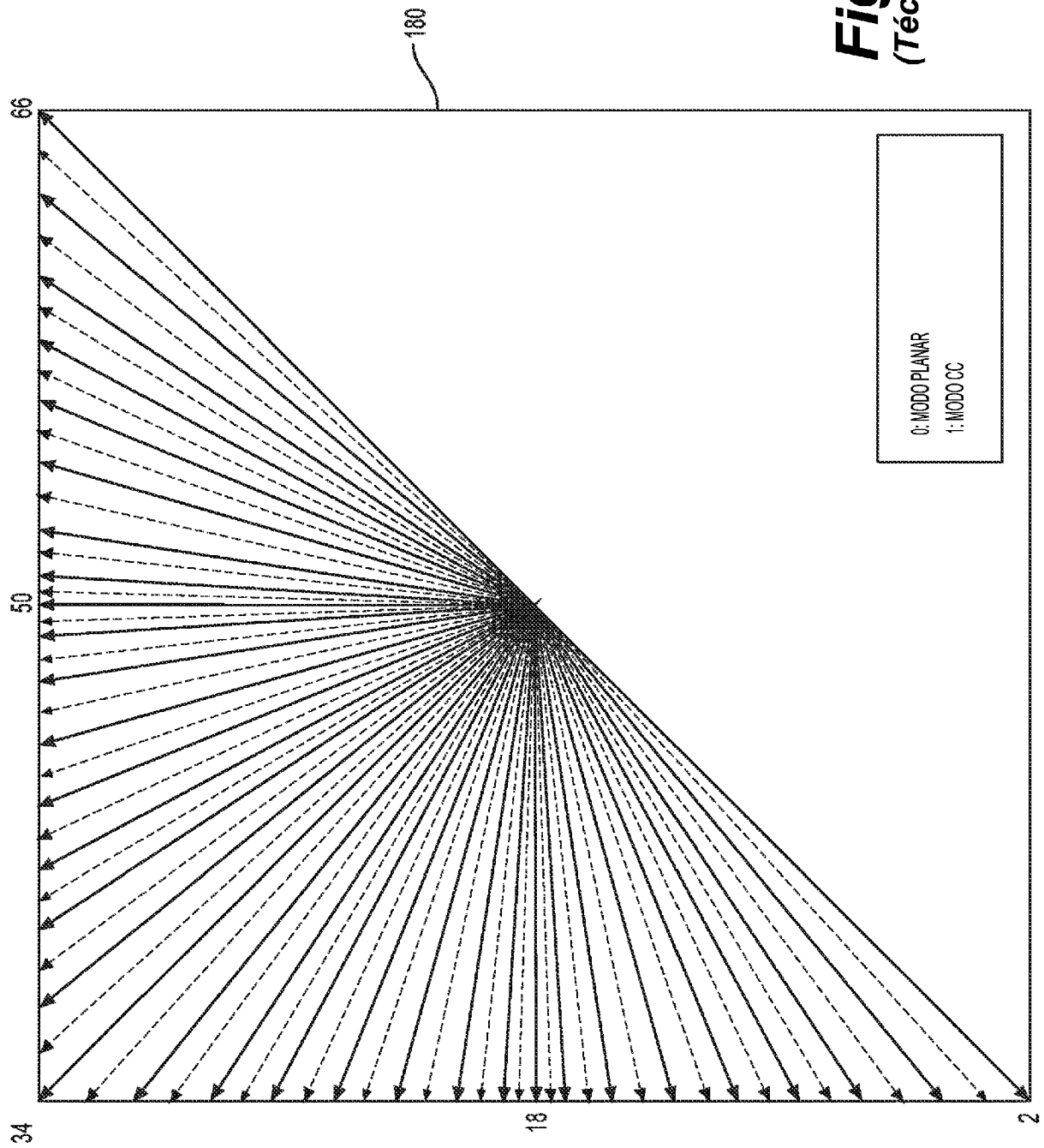


Figura 1B
(Técnica anterior)

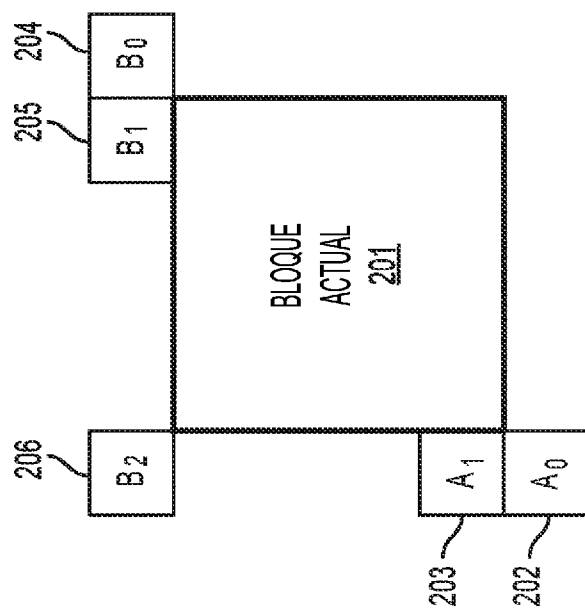


Figura 2
(Técnica anterior)

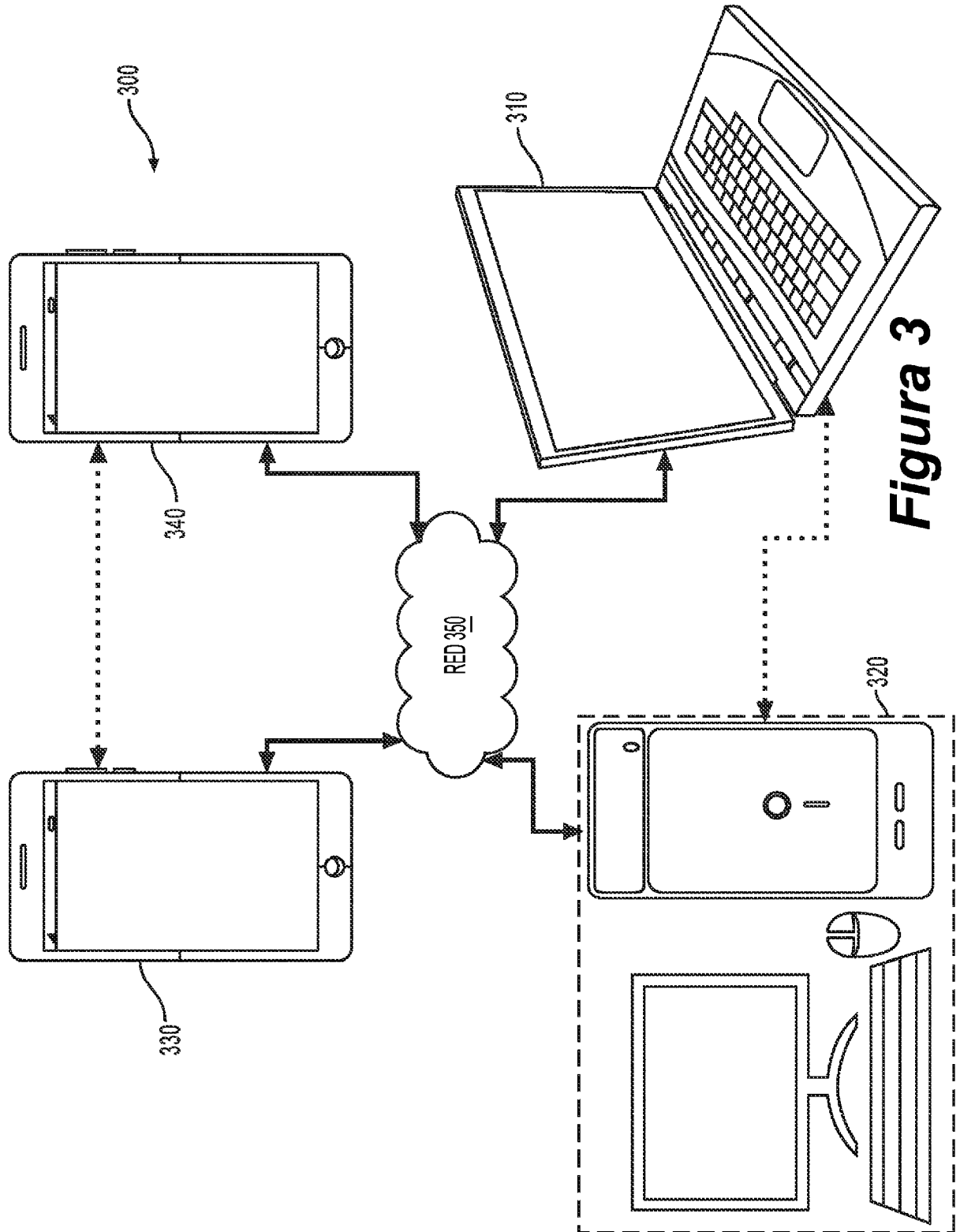


Figura 3

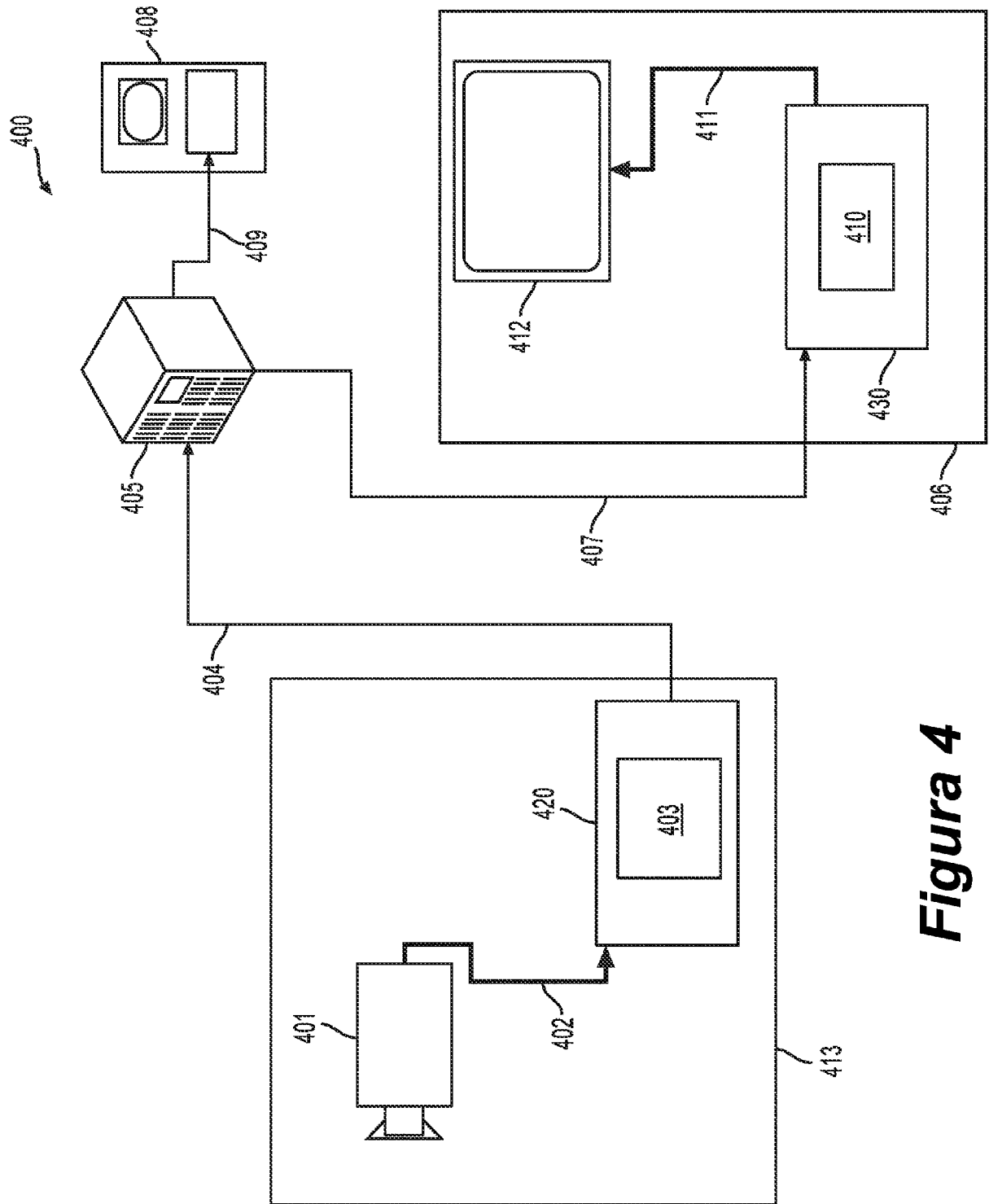


Figure 4

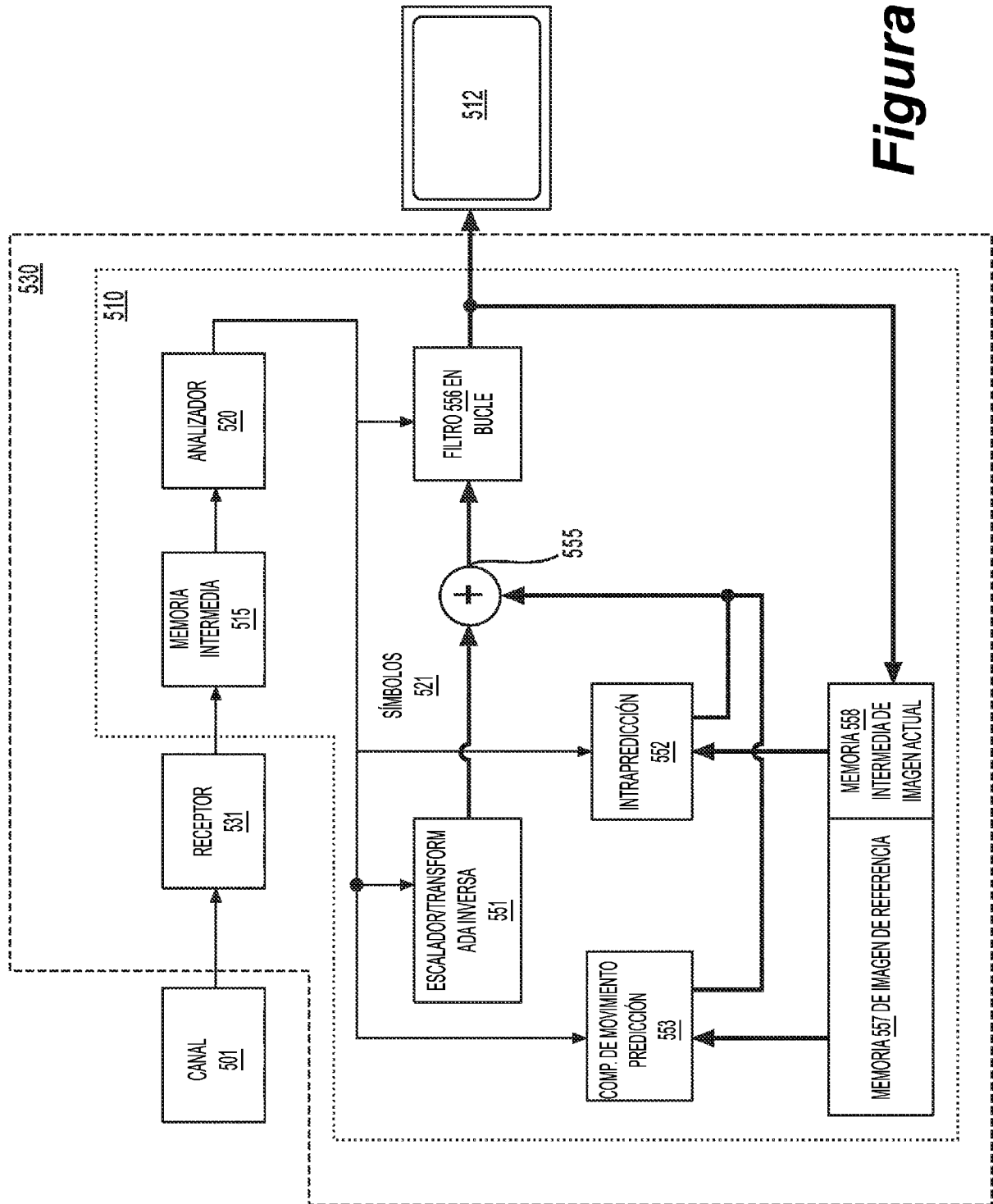


Figura 5

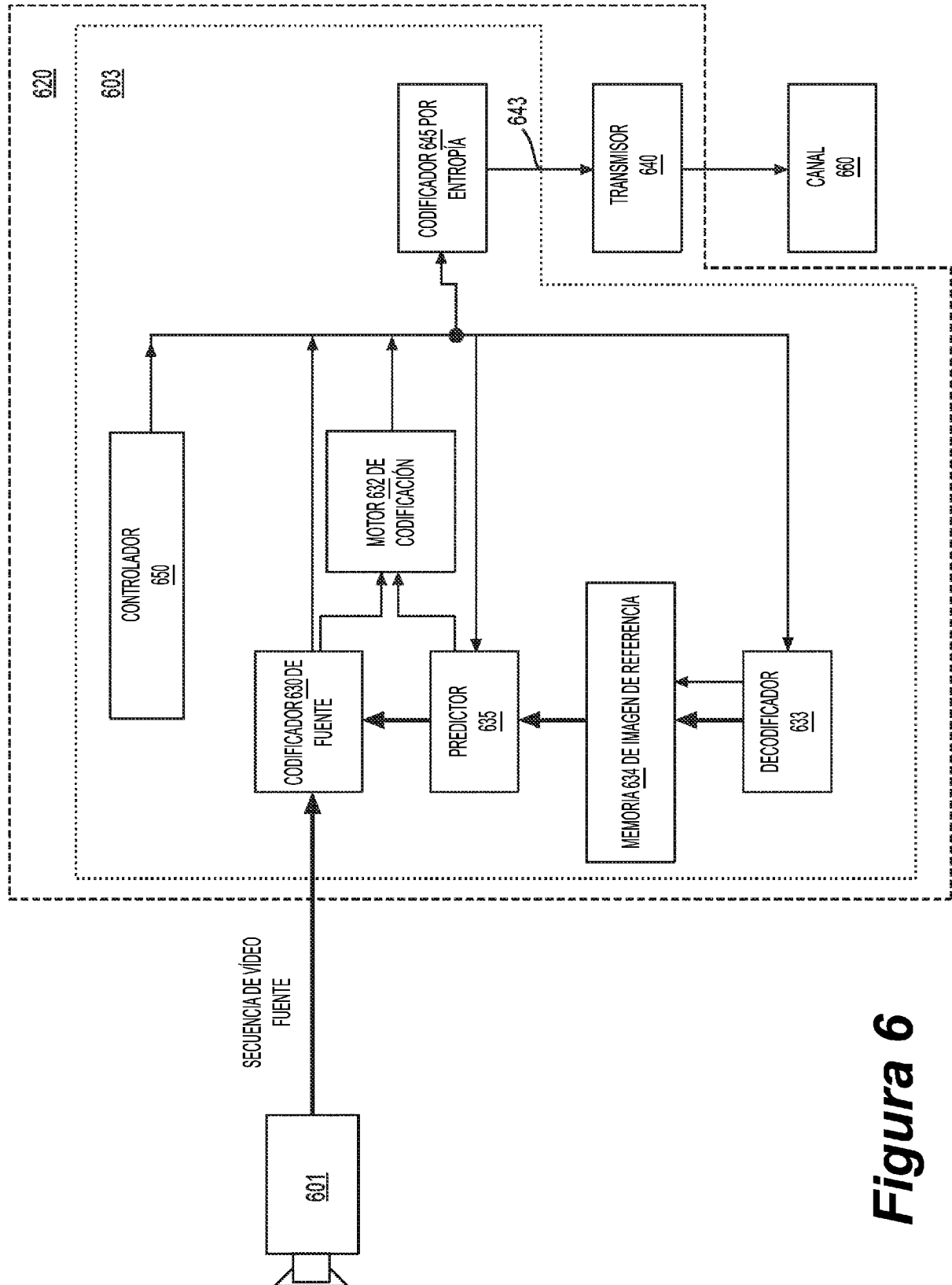


Figura 6

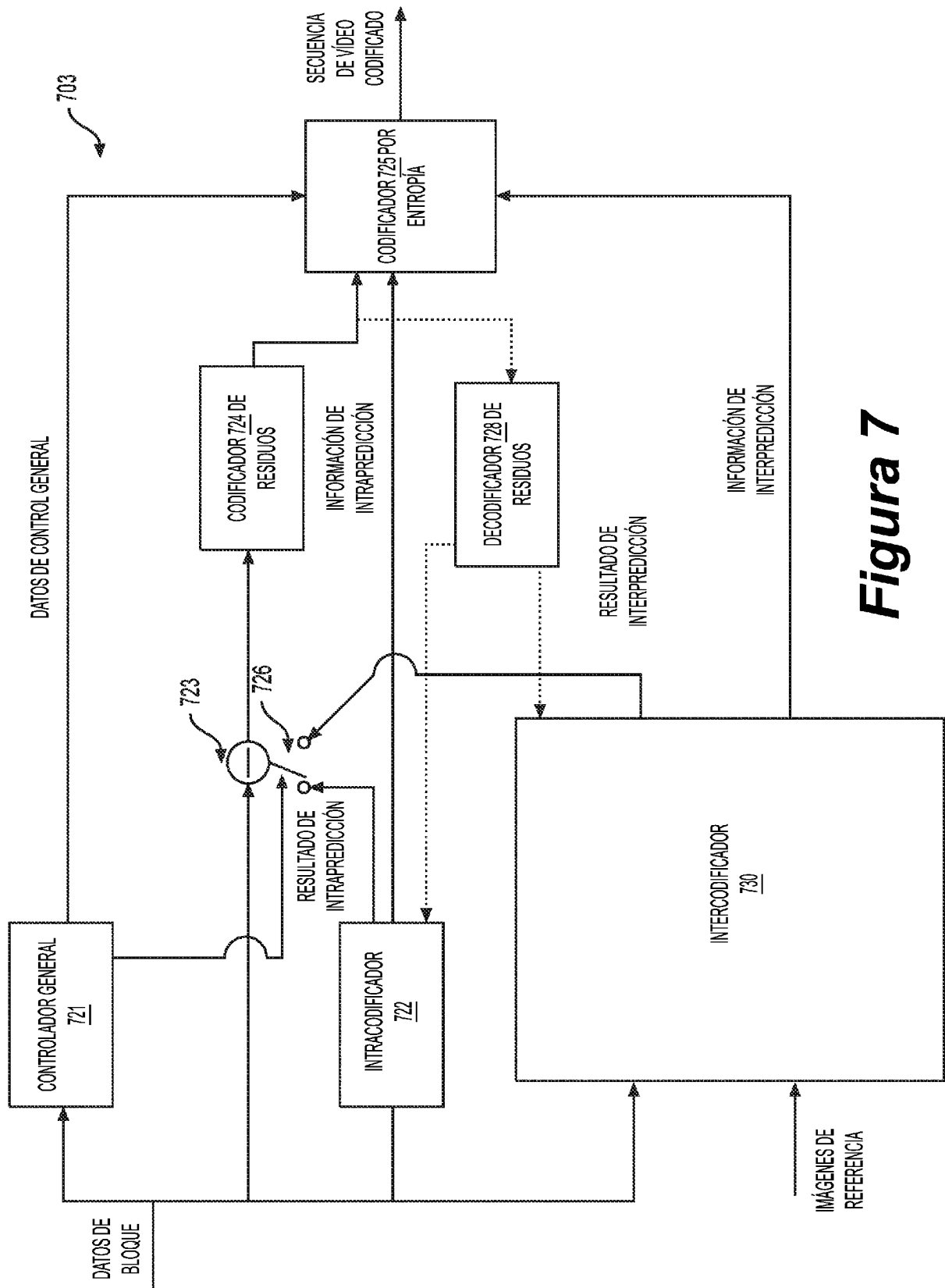


Figura 7

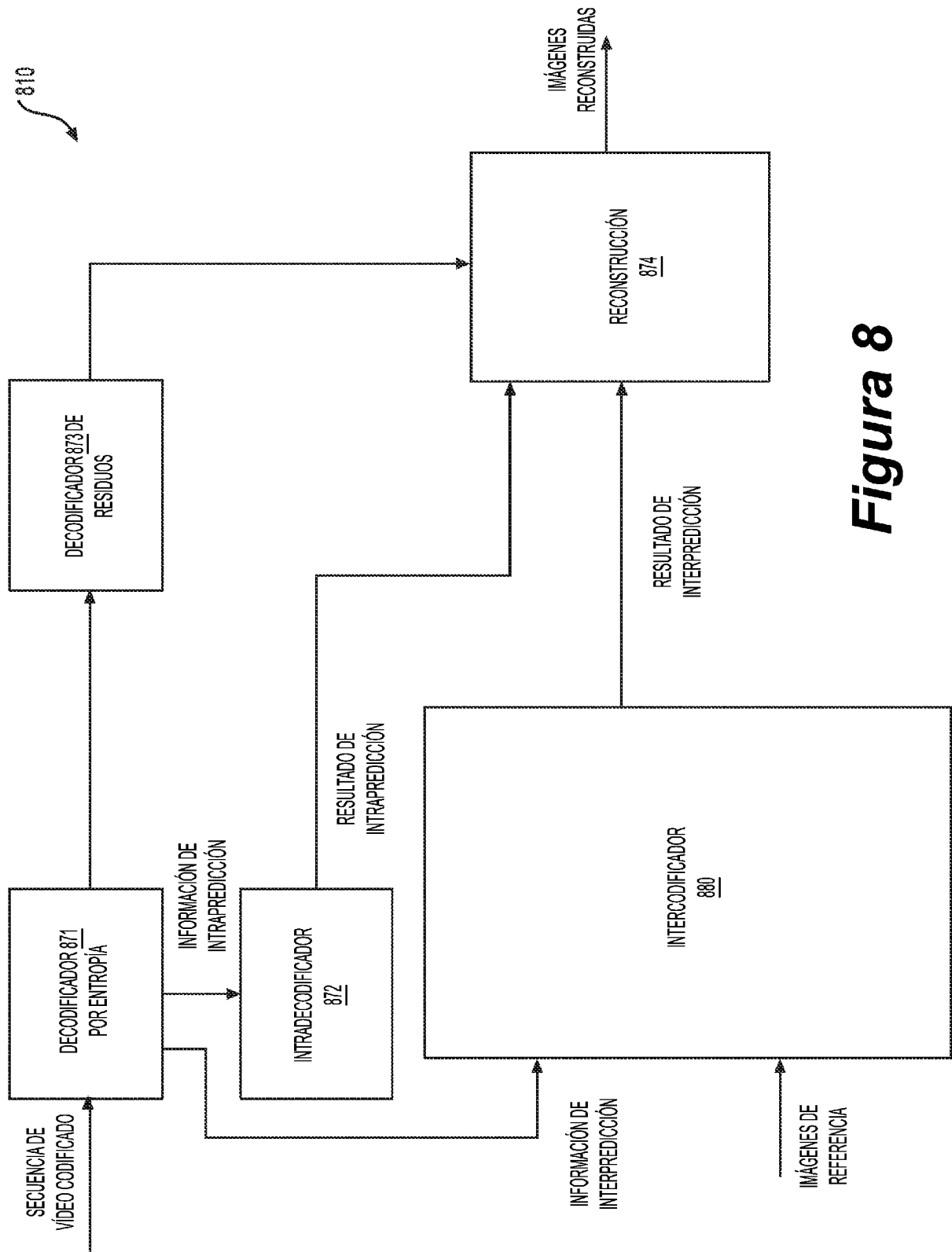


Figura 8

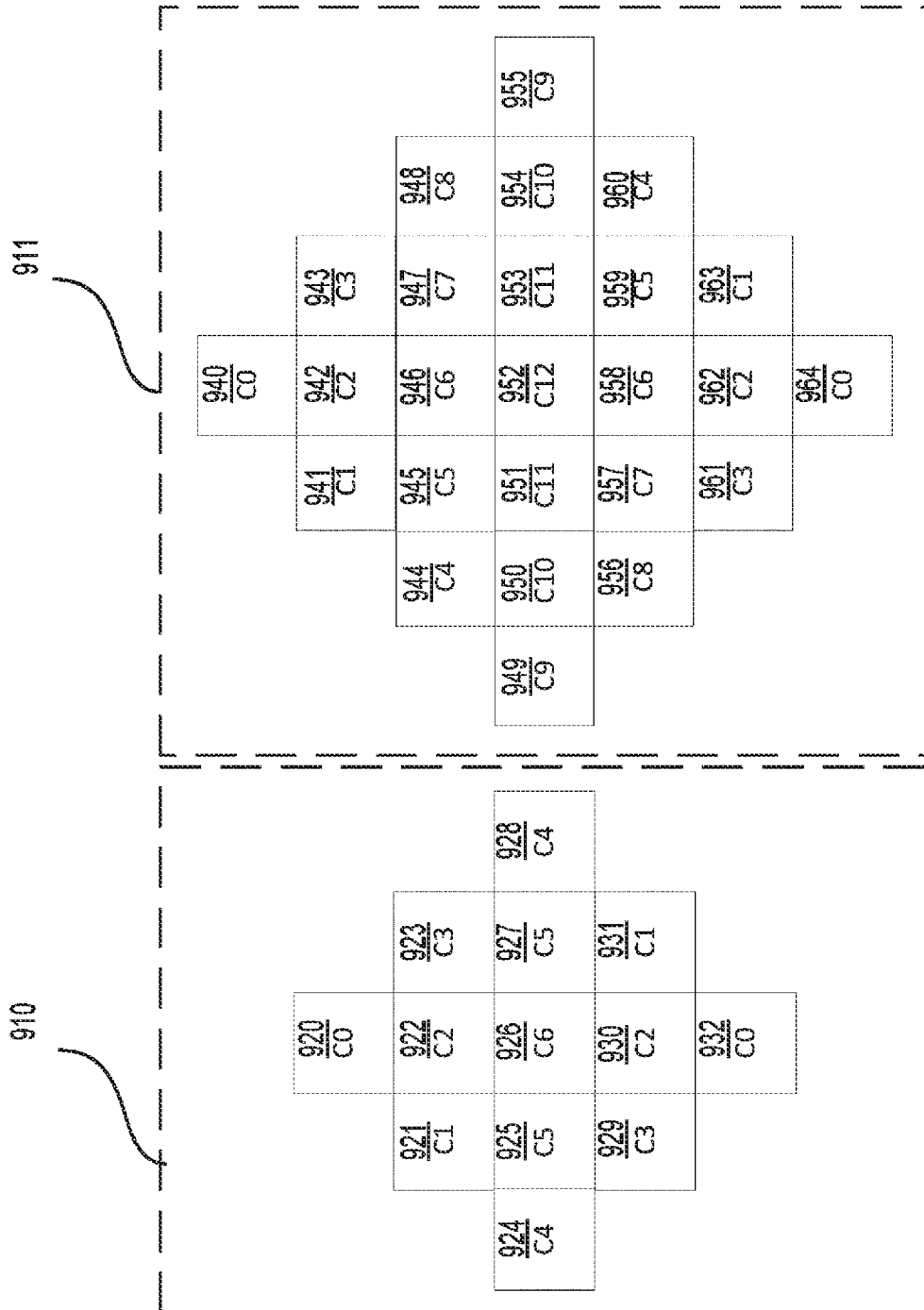


Figura 9

	H		H		H		H
H		H		H		H	
	H		H		H		H
H						H	
	H						H
H						H	
	H		H		H		H
H		H		H		H	

Figura 10B

	✓		✓		✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓
✓						✓	
	✓						✓
✓						✓	
	✓		✓		✓		✓
✓		✓		✓		✓	

Figura 10A

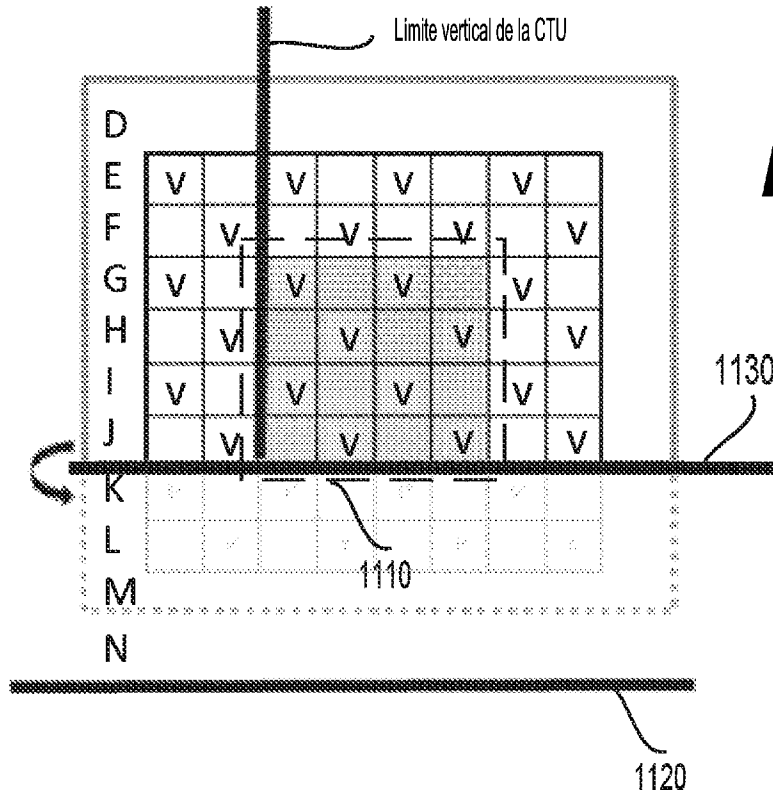


Figura 11A

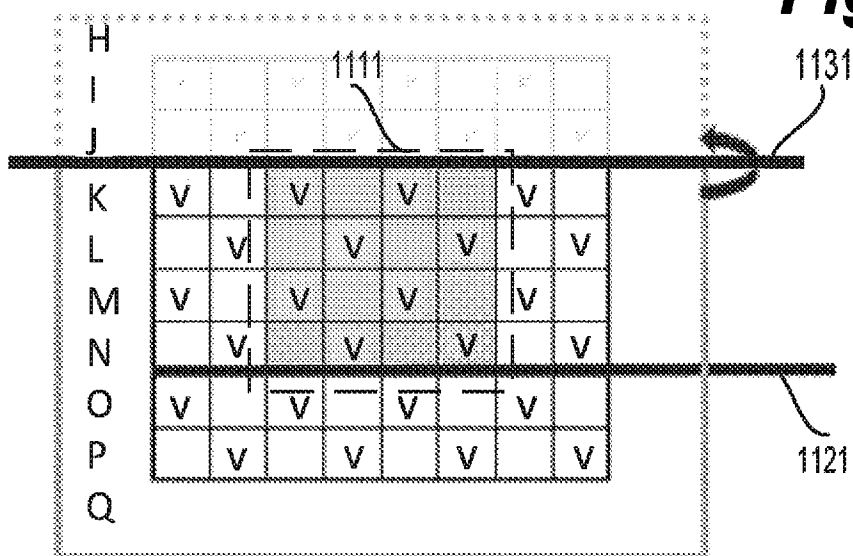


Figura 11B

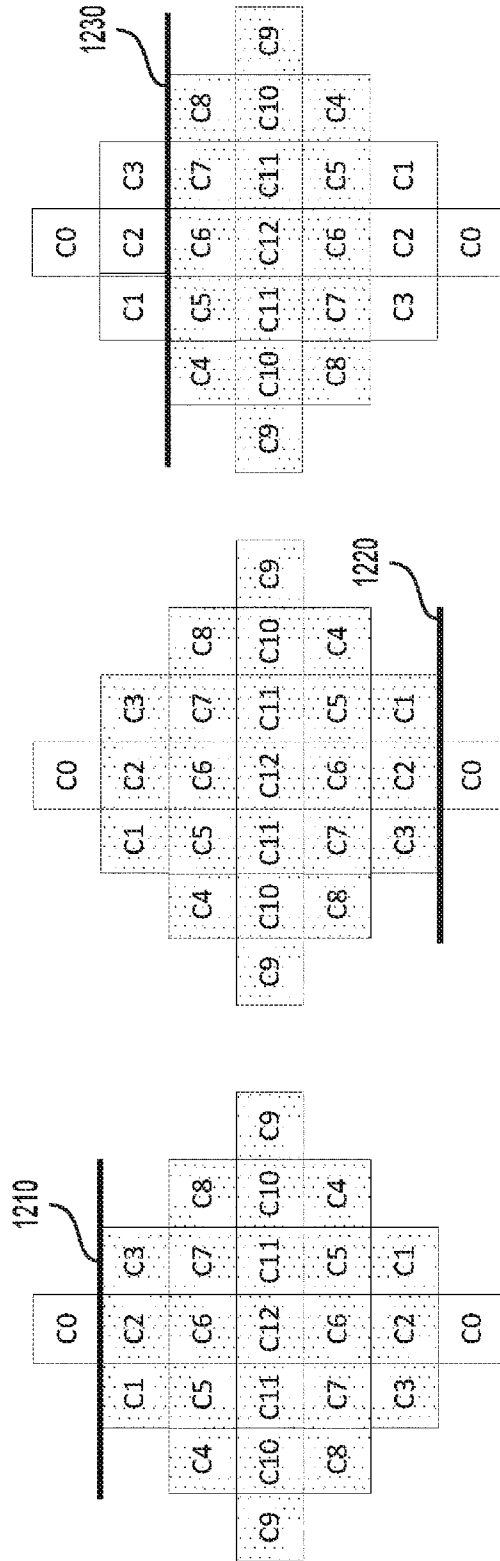


Figura 12A

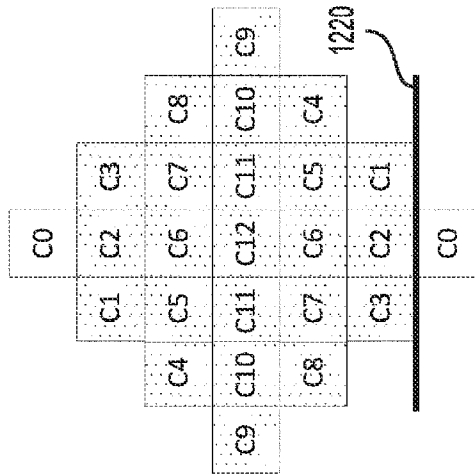


Figura 12B

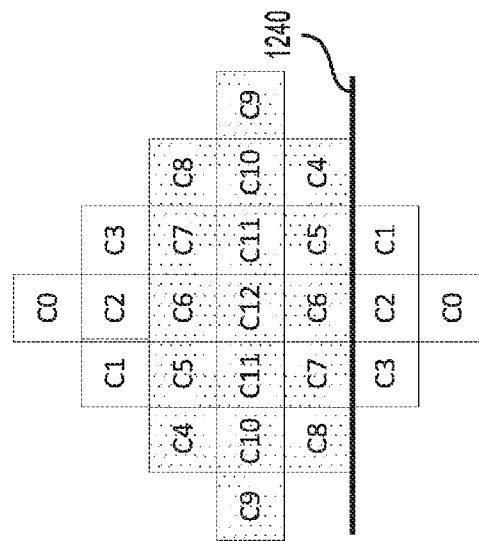


Figura 12D

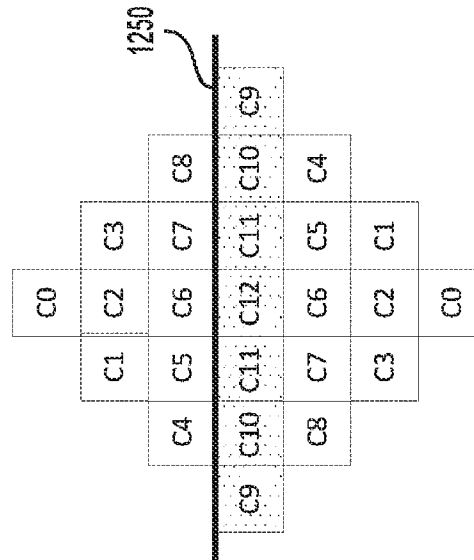


Figura 12E

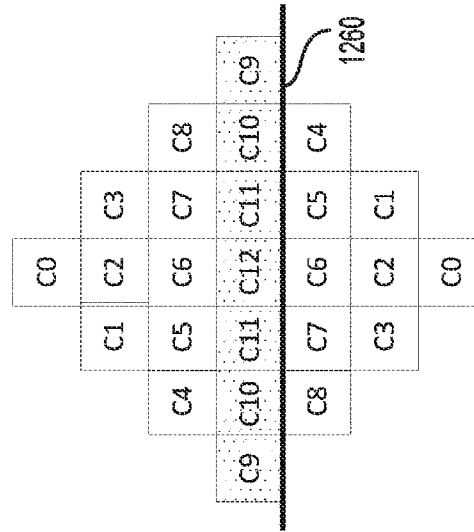


Figura 12F

Figura 12C

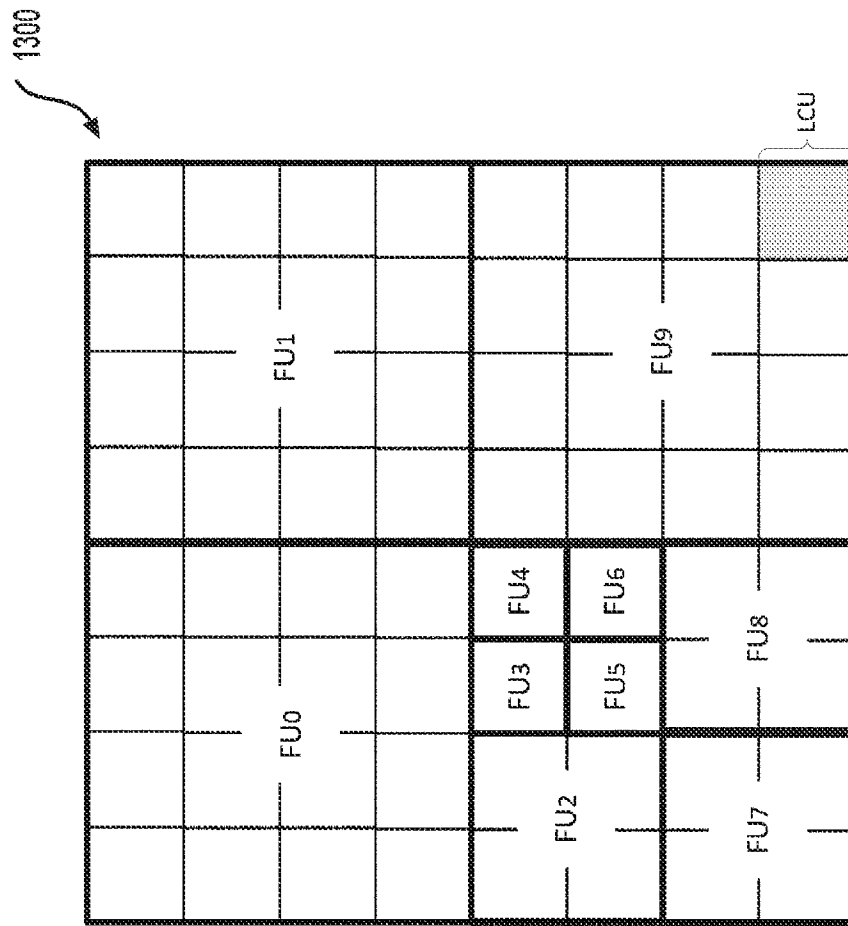


Figure 13

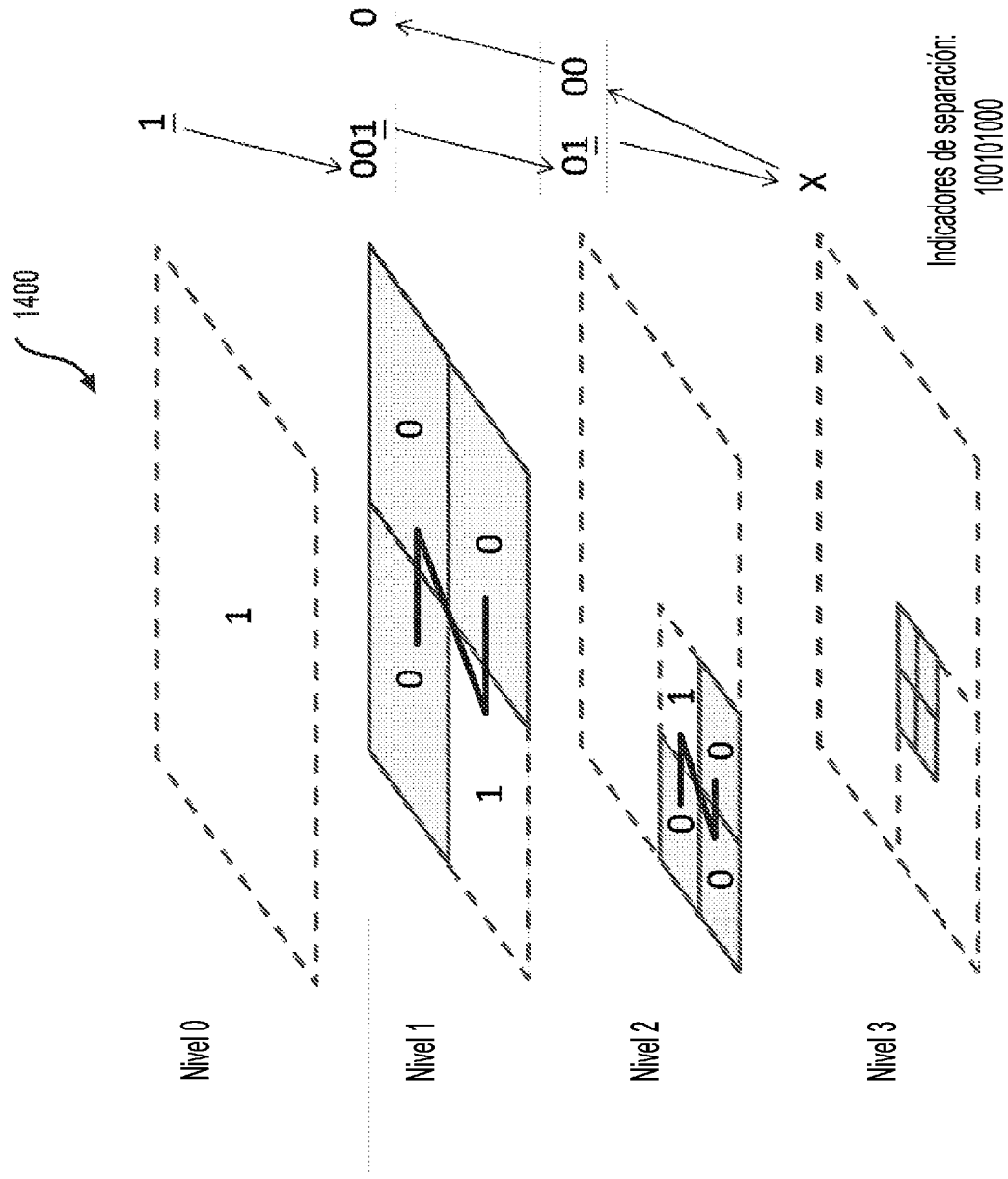


Figura 14

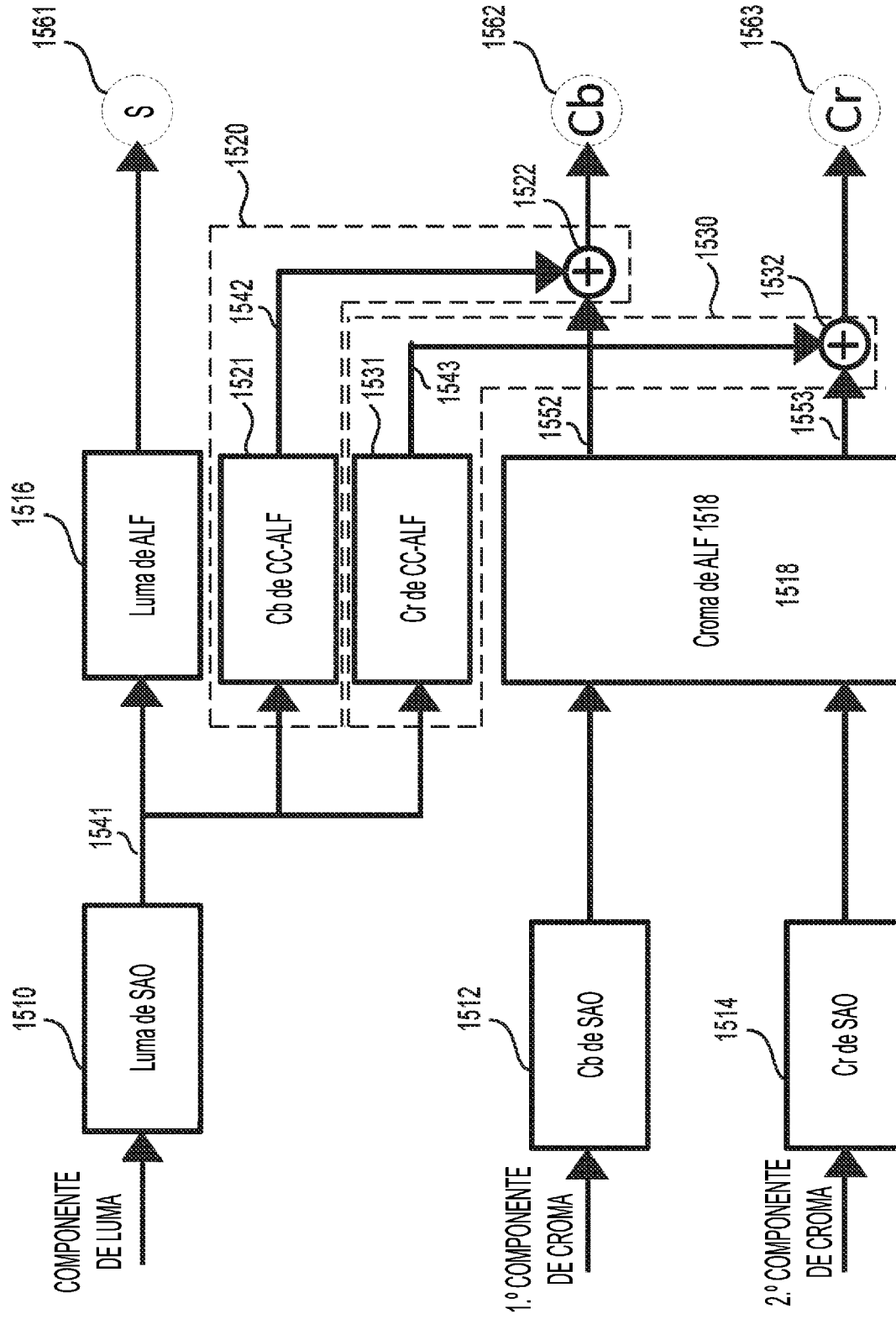


Figura 15

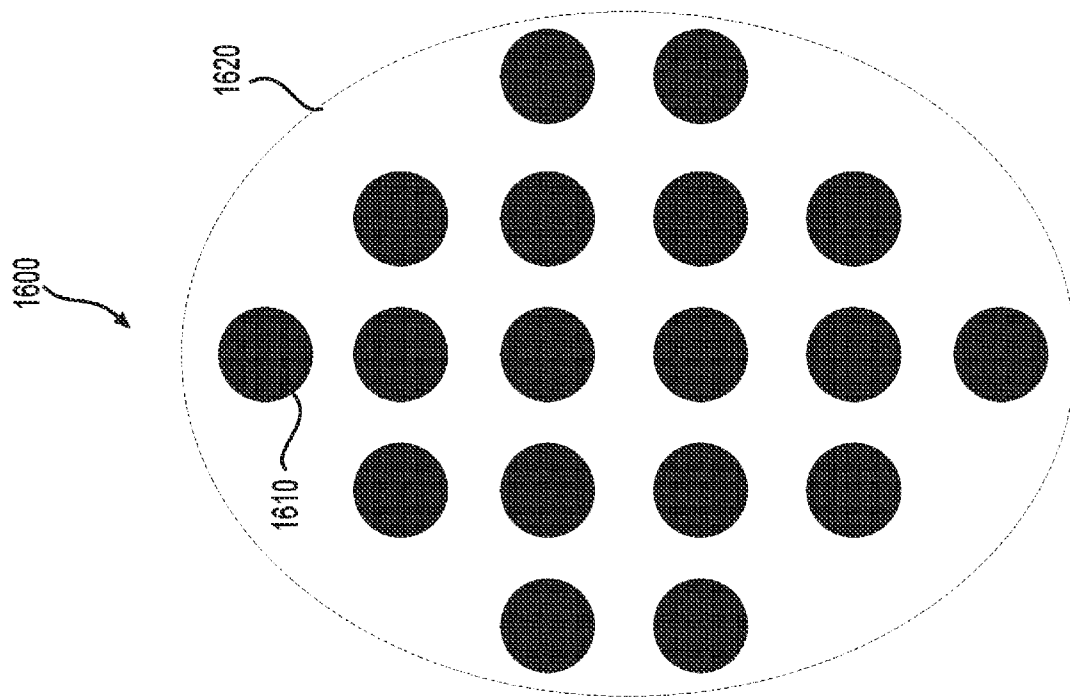


Figura 16

if (slice_cross_component_alf_cb_enabled_flag)	
alf_ctb_cross_component_cb_idc [xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if(slice_cross_component_alf_cb_enabled_flag == 0 alf_ctb_cross_component_cb_idc[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY] == 0)	
if(slice_alf_chroma_idc == 1 slice_alf_chroma_idc == 3) {	
alf_ctb_flag [1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if(alf_ctb_flag[1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY] && aps_alf_chroma_num_alt_filters_minus1 > 0)	
alf_ctb_filter_alt_idx [0][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
}	
if (slice_cross_component_alf_cr_enabled_flag)	
alf_ctb_cross_component_cr_idc [xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if(slice_cross_component_alf_cr_enabled_flag == 0 alf_ctb_cross_component_cr_idc[xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY] == 0)	
if(slice_alf_chroma_idc == 2 slice_alf_chroma_idc == 3) {	
alf_ctb_flag [2][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
if(alf_ctb_flag[2][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY] && aps_alf_chroma_num_alt_filters_minus1 > 0)	
alf_ctb_filter_alt_idx [1][xCtb >> CtbLog2SizeY][yCtb >> CtbLog2SizeY]	ae(v)
}	

Figura 17

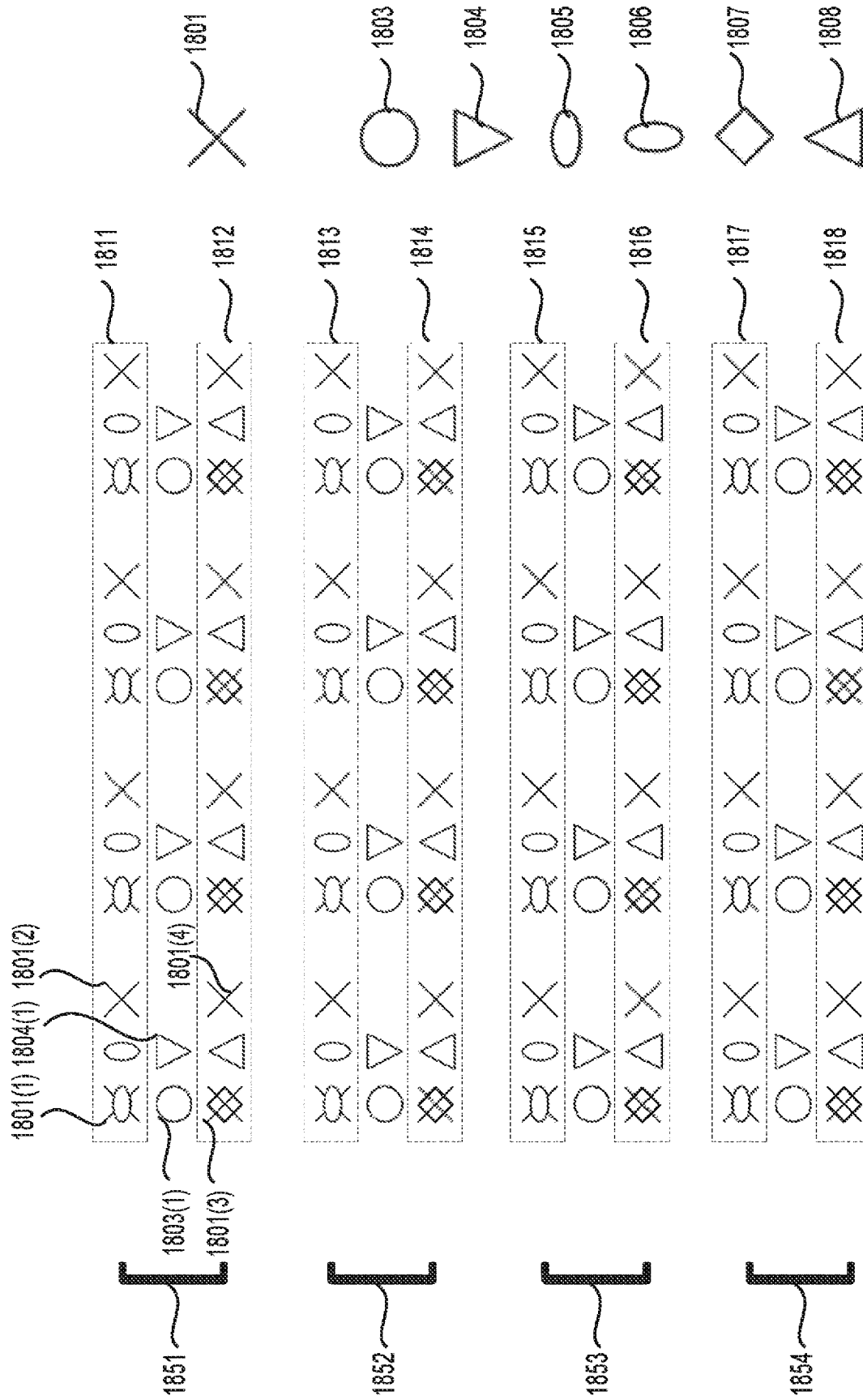


Figure 18A

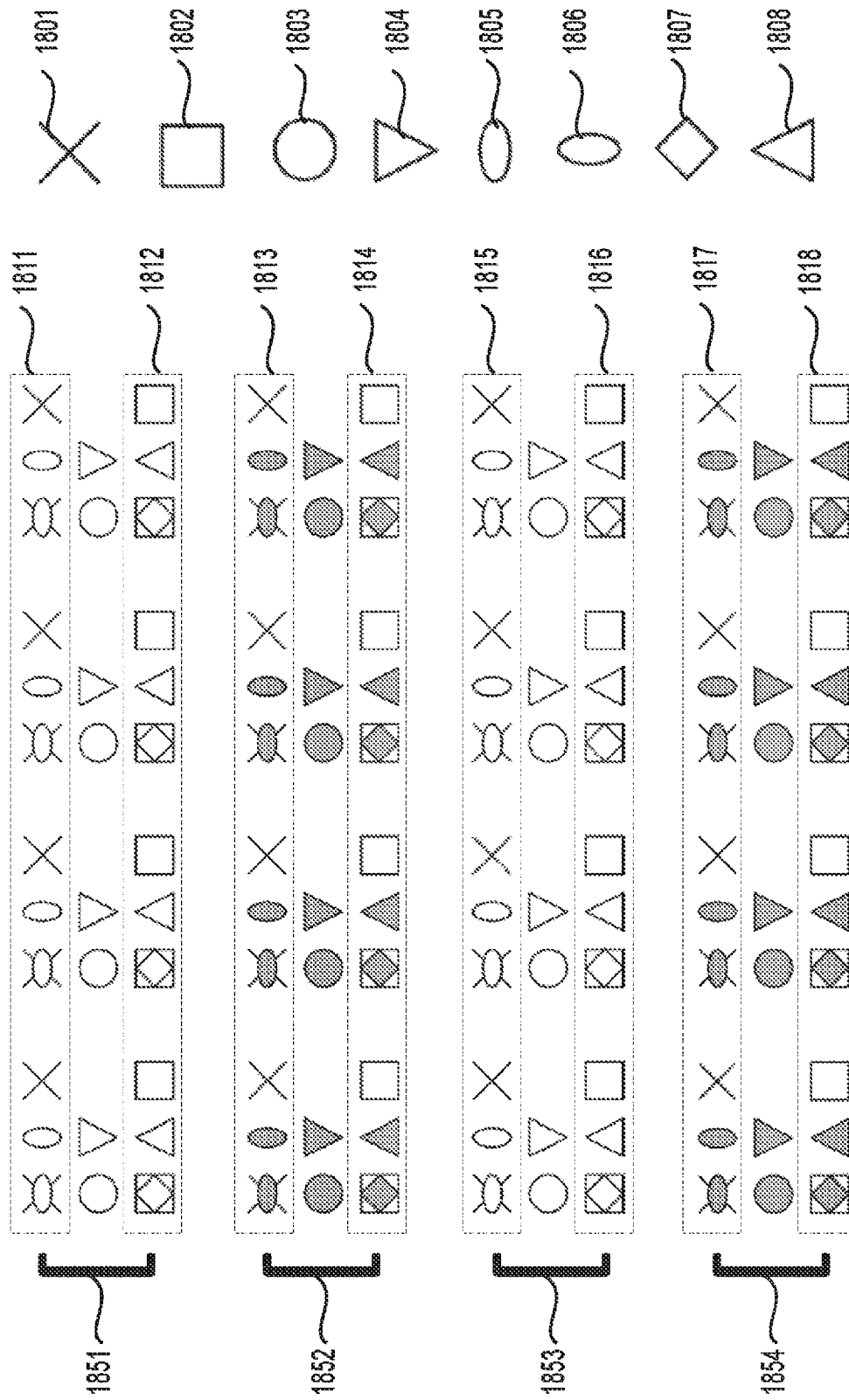


Figura 18B

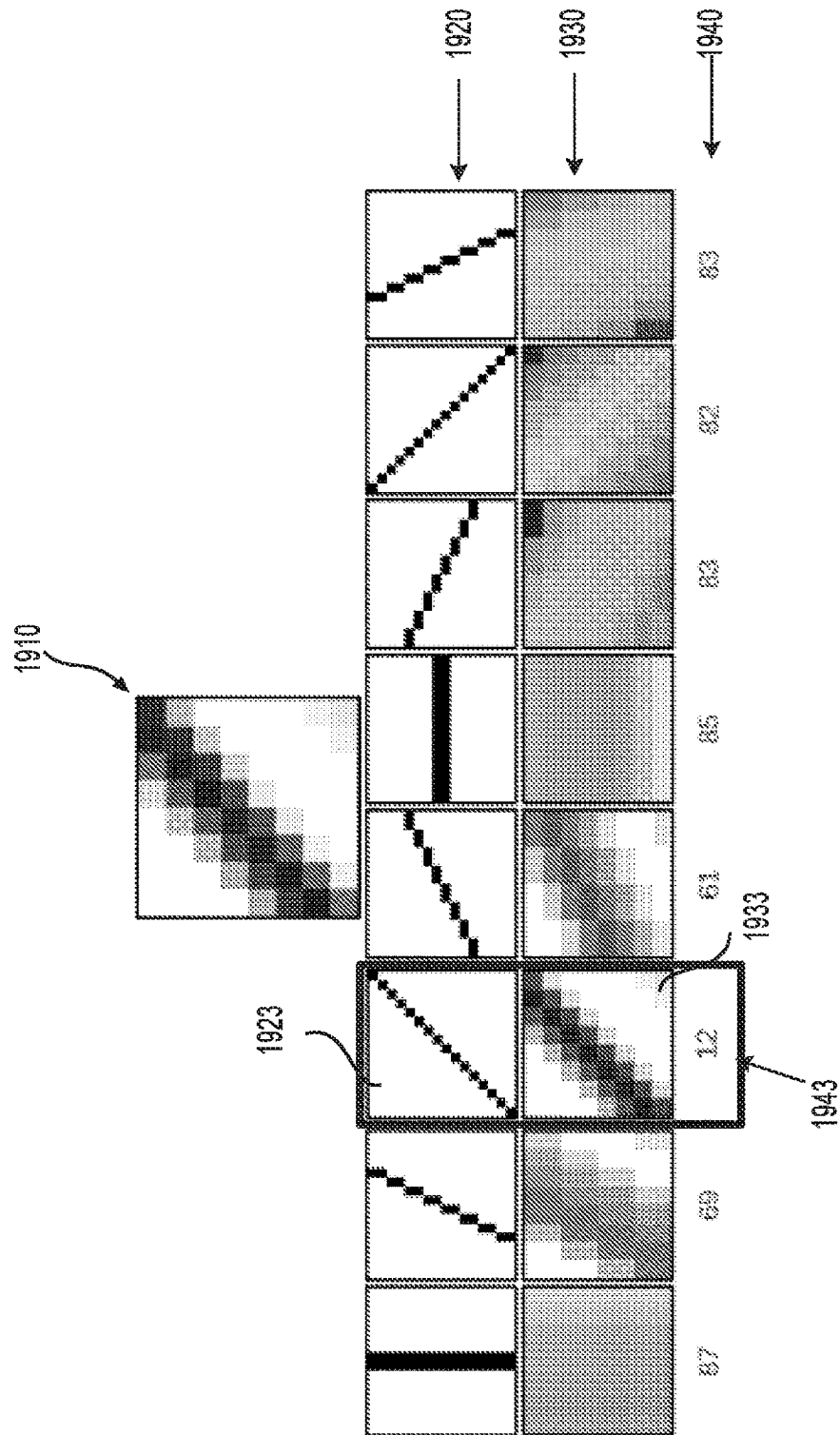


Figura 19

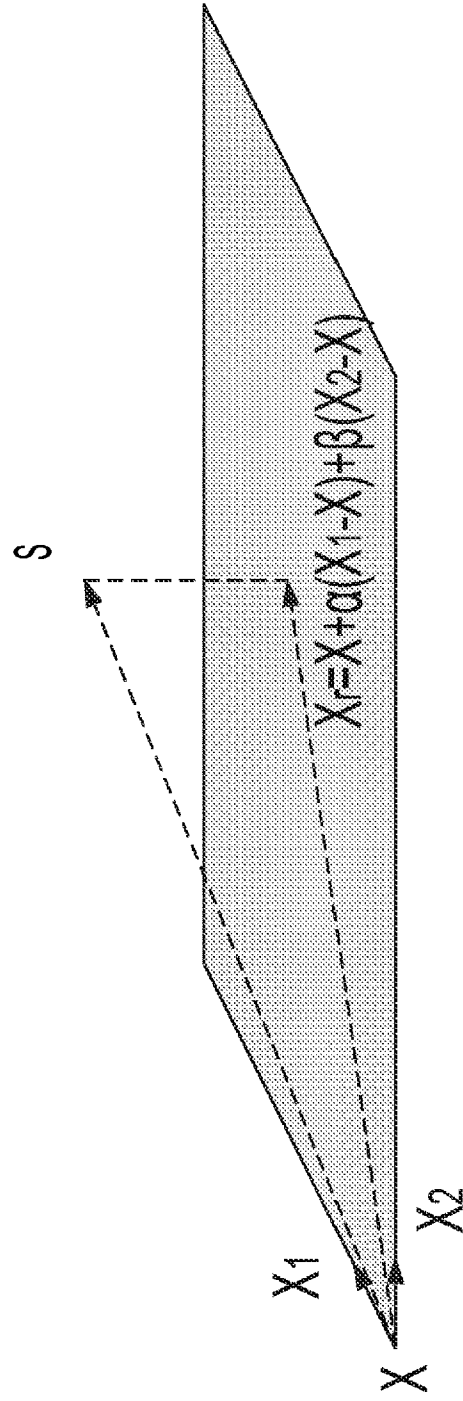


Figura 20

2100

Tipo de SAO	Tipo de desfase adaptativo de muestra que se utilizará	Número de categorías
0	Ninguno	0
1	Desfase de borde del patrón 1-D de 0 grados	4
2	Desfase de borde del patrón 1-D de 90 grados	4
3	Desfase de borde del patrón 1-D de 135 grados	4
4	Desfase de borde del patrón 1-D de 45 grados	4
5	bandas centrales, desfase de banda	16
6	bandas laterales, desfase de banda	16

Figura 21

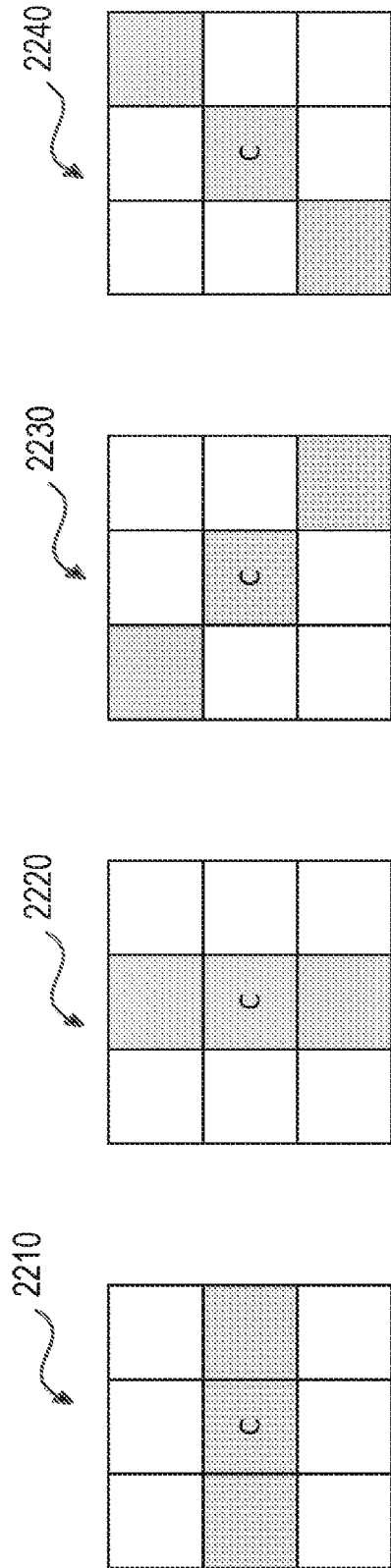


Figura 22

2300

Categoría	Condición
1	$c < 2$ vecinos
2	$c < 1$ vecino && $c == 1$ vecino
3	$c > 1$ vecino && $c == 1$ vecino
4	$c > 2$ vecinos
0	Ninguna de las anteriores

Figura 23

2400

sao_offset_vlc(rx, ry, cldx) {	Descriptor
sao_type_idx[cldx][rx][ry]	ue(v)
if(sao_type_idx[cldx][rx][ry] == 5) {	
sao_band_position[cldx][rx][ry]	u(5)
for(i = 0; i < 4; i++)	
sao_offset[cldx][rx][ry][i]	se(v)
} else if(sao_type_idx[cldx][rx][ry] != 0)	
for(i = 0; i < 4; i++)	
sao_offset[cldx][rx][ry][i]	ue(v)
}	

Figura 24

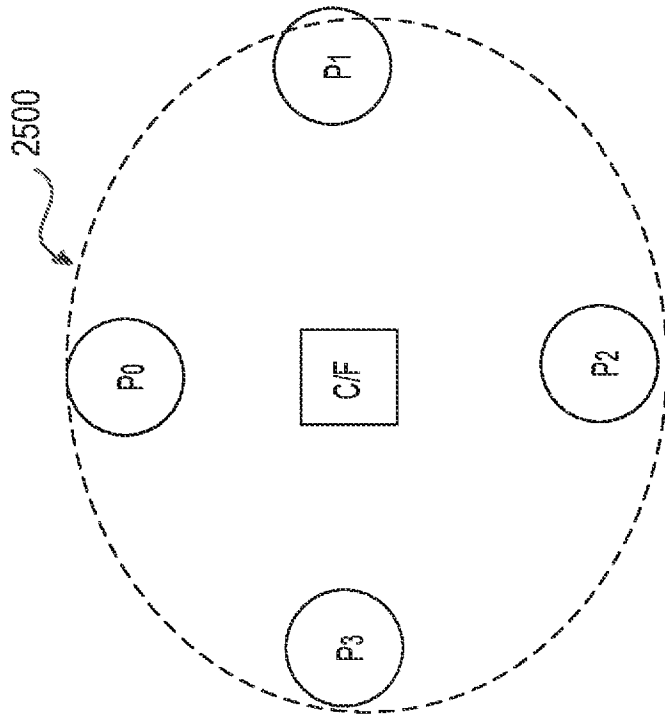


Figura 25

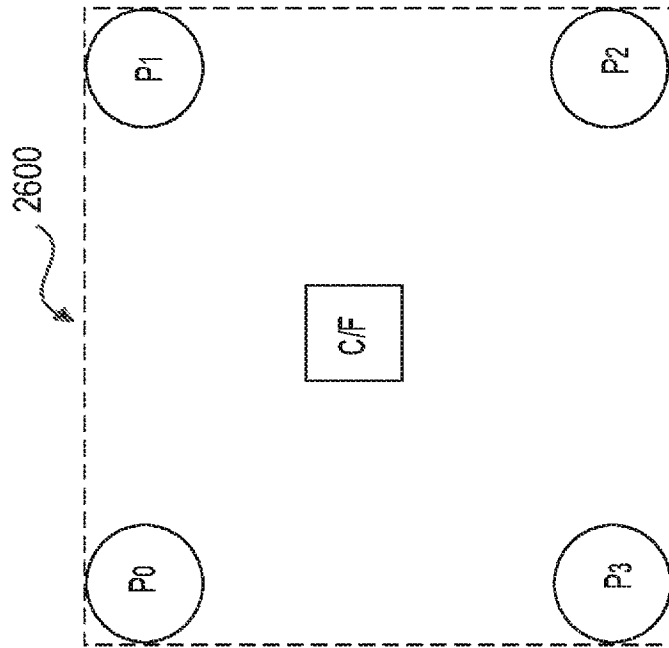
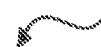


Figura 26

 2700

Combinación	d0	d1	d2	d3	Desfase
0	-1	-1	-1	-1	s0
1	-1	-1	-1	0	s1
2	-1	-1	-1	1	s2
3	-1	-1	0	-1	s3
4	-1	-1	0	0	s4
5	-1	-1	0	1	s5
6	-1	-1	1	-1	s6
7	-1	-1	1	0	s7
8	-1	-1	1	1	s8
9	-1	0	-1	-1	s9
10	-1	0	-1	0	s10
11	-1	0	-1	1	s11
12	-1	0	0	-1	s12
13	-1	0	0	0	s13
14	-1	0	0	1	s14
15	-1	0	1	-1	s15
16	-1	0	1	0	s16
17	-1	0	1	1	s17
18	-1	1	-1	-1	s18
19	-1	1	-1	0	s19
20	-1	1	-1	1	s20
21	-1	1	0	-1	s21
22	-1	1	0	0	s22
23	-1	1	0	1	s23
24	-1	1	1	-1	s24
25	-1	1	1	0	s25
26	-1	1	1	1	s26

Figura 27A

 2700

Combinación	d0	d1	d2	d3	Desfase
27	0	-1	-1	-1	s27
28	0	-1	-1	0	s28
29	0	-1	-1	1	s29
30	0	-1	0	-1	s30
31	0	-1	0	0	s31
32	0	-1	0	1	s32
33	0	-1	1	-1	s33
34	0	-1	1	0	s34
35	0	-1	1	1	s35
36	0	0	-1	-1	s36
37	0	0	-1	0	s37
38	0	0	-1	1	s38
39	0	0	0	-1	s39
40	0	0	0	0	s40
41	0	0	0	1	s41
42	0	0	1	-1	s42
43	0	0	1	0	s43
44	0	0	1	1	s44
45	0	1	-1	-1	s45
46	0	1	-1	0	s46
47	0	1	-1	1	s47
48	0	1	0	-1	s48
49	0	1	0	0	s49
50	0	1	0	1	s50
51	0	1	1	-1	s51
52	0	1	1	0	s52
53	0	1	1	1	s53

Figura 27B

 2700

Combinación	d0	d1	d2	d3	Desfase
54	1	-1	-1	-1	s54
55	1	-1	-1	0	s55
56	1	-1	-1	1	s56
57	1	-1	0	-1	s57
58	1	-1	0	0	s58
59	1	-1	0	1	s59
60	1	-1	1	-1	s60
61	1	-1	1	0	s61
62	1	-1	1	1	s62
63	1	0	-1	-1	s63
64	1	0	-1	0	s64
65	1	0	-1	1	s65
66	1	0	0	-1	s66
67	1	0	0	0	s67
68	1	0	0	1	s68
69	1	0	1	-1	s69
70	1	0	1	0	s70
71	1	0	1	1	s71
72	1	1	-1	-1	s72
73	1	1	-1	0	s73
74	1	1	-1	1	s74
75	1	1	0	-1	s75
76	1	1	0	0	s76
77	1	1	0	1	s77
78	1	1	1	-1	s78
79	1	1	1	0	s79
80	1	1	1	1	s80

Figura 27C

2800

Combinación	d0	d1	d2	d3
1	1	1	-1	-1
2	1	-1	-1	1
3	-1	1	1	-1
4	-1	-1	1	1
5	0	0	0	0

Figura 28

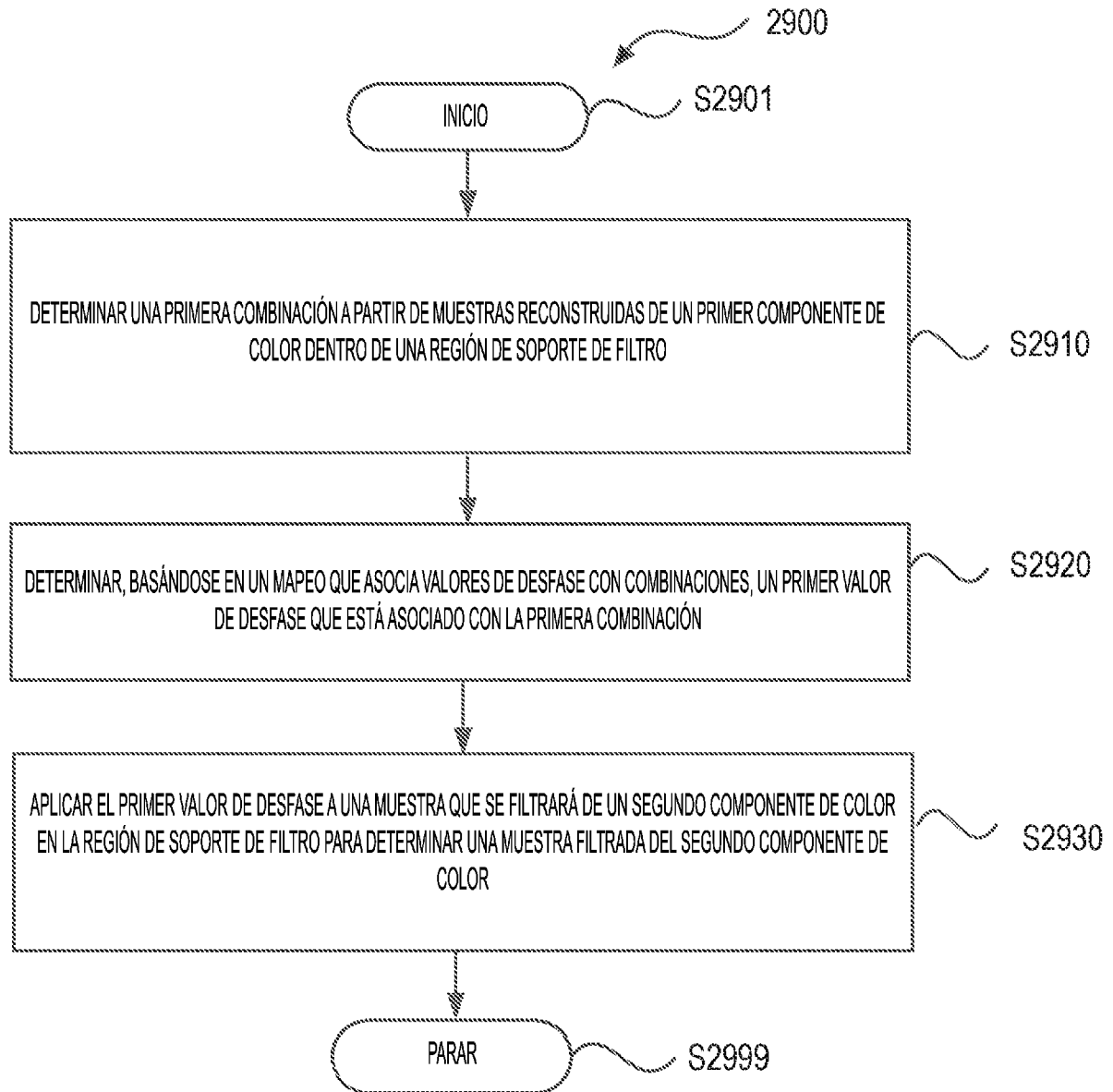


Figura 29

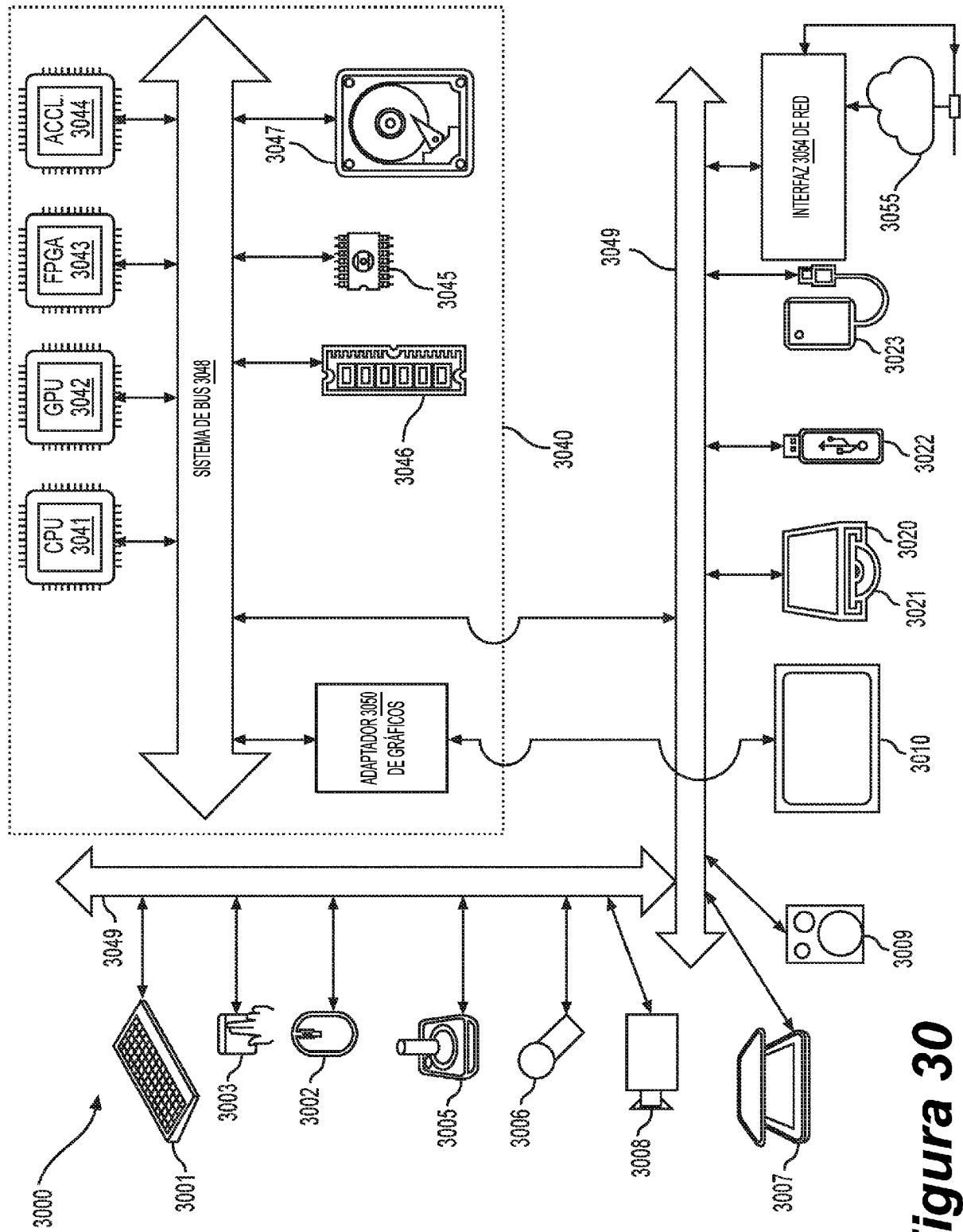


Figura 30