

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 761 891**

51 Int. Cl.:

H04N 11/02 (2006.01)
H04N 19/91 (2014.01)
H04N 19/70 (2014.01)
H04N 19/46 (2014.01)
H04N 19/119 (2014.01)
H04N 19/174 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **29.06.2011 E 18152959 (5)**

97 Fecha y número de publicación de la concesión europea: **18.09.2019 EP 3334155**

54 Título: **Codificación y decodificación de imagen**

30 Prioridad:

21.07.2010 US 366215 P

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

21.05.2020

73 Titular/es:

**VELOS MEDIA INTERNATIONAL LIMITED
(100.0%)
Unit 32, the Hyde Building The Park,
Carrickmines
Dublin 18, IE**

72 Inventor/es:

**SJÖBERG, RICKARD y
PRIDDLE, CLINTON**

74 Agente/Representante:

PONS ARIÑO, Ángel

ES 2 761 891 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

Codificación y decodificación de imagen

5

CAMPO TÉCNICO

Las realizaciones presentes generalmente se refieren al procesamiento de imágenes o fotogramas y en particular a la codificación y decodificación de tales imágenes o fotogramas.

10

ANTECEDENTES

H.264, también conocido como MPEG-4 (Motion Picture Expert Group) AVC (Advanced Video Coding), es el estándar de codificación de vídeo de última generación. Es un códec híbrido que se basa en eliminar la redundancia entre fotogramas, lo que se denomina codificación inter, y dentro de las tramas, lo que se denomina codificación intra. La salida del proceso de codificación son datos VCL (Video Coding Layer), que se encapsulan adicionalmente en unidades NAL (Network Abstraction Layer) antes de la transmisión o almacenamiento.

En H.264/MPEG-4 AVC, una imagen de una transmisión de vídeo está compuesta de macrobloques con un tamaño fijo de 16x16 píxeles y la codificación de la imagen se produce macrobloque por macrobloque. Cada imagen de la transmisión de vídeo se divide en uno o más segmentos. Un corte es una pieza decodificable independientemente de la imagen. Esto significa que si se pierde una parte de una imagen, las otras partes de la imagen aún son decodificables. Además, los cortes pueden usarse para codificación y decodificación paralelas, ya que son independientes de otros cortes en la imagen. En H.264/MPEG-4 AVC, deberá producirse un límite de corte entre el borde de dos macrobloques adyacentes, según el orden de codificación.

HEVC (High Efficiency Video Coding) es el sucesor de H.264/MPEG-4 AVC. HEVC tiene como objetivo mejorar sustancialmente la eficiencia de codificación en comparación con H.264/MPEG-4 AVC, es decir, reducir los requisitos de velocidad de bits mientras se mantiene la calidad de la imagen. HEVC está dirigido a pantallas HDTV (High Definition Television) y sistemas de captura de contenido de próxima generación que cuentan con número de fotogramas por segundo escaneados progresivo y resoluciones de pantalla desde QVGA (Quarter Video Graphics Array) (320x240) hasta 1080p y Ultra HDTV (7680x4320), así como una calidad de imagen mejorada.

HEVC permite el uso de las llamadas unidades máximas de codificación (LCU), que son bloques de píxeles que tienen un tamaño que es más grande que los macrobloques de H.264/MPEG-4 AVC para proporcionar una eficiencia de codificación mejorada. Con el fin de manejar grandes áreas homogéneas y pequeñas áreas detalladas en la misma imagen, se ha propuesto una codificación jerárquica para HEVC. Las LCU en una imagen se escanean en un orden predefinido y cada una de estas LCU se puede dividir en unidades de codificación más pequeñas (CU), que a su vez se pueden dividir jerárquicamente de forma cuadrática en una unidad de codificación más pequeña (SCU). Por lo tanto, una imagen puede codificarse como una mezcla de unidades de codificación con diferentes tamaños que van desde las LCU hasta las SCU.

En correspondencia con H.264/MPEG-4 AVC, una imagen de una transmisión de vídeo se puede dividir en uno o más cortes en HEVC. El límite de corte está en HEVC alineado con el borde de dos LCU adyacentes, según un orden predefinido.

Tanto los estándares H.264/MPEG-4 AVC como HEVC requieren la determinación y el uso de direcciones para identificar el primer macrobloque o unidad de codificación de un corte y, por lo tanto, el inicio del corte en la imagen o fotograma. Dichas direcciones, aunque necesarias en el decodificador, agregan sobrecarga a los datos codificados de la imagen. Además, con la introducción de la división jerárquica de la LCU en HEVC surgen nuevos desafíos en relación con la codificación y decodificación de imágenes o fotogramas. Por lo tanto, existe la necesidad de una codificación y decodificación eficiente que pueda manejar las direcciones de los cortes de inicio de una manera eficiente y flexible.

55 RESUMEN

Es un objetivo general proporcionar una gestión eficiente de los cortes en imágenes y fotogramas.

Es un objetivo particular señalar las posiciones de inicio de corte de una manera eficiente.

60

Estos y otros objetivos se consiguen mediante las realizaciones descritas en el presente documento.

Un aspecto de las realizaciones define un procedimiento de codificación de una imagen que comprende dos o más cortes. Se genera una representación de corte codificada para cada corte en la imagen en función de los valores de píxel de los píxeles en el corte. Se asigna un indicador de sector respectivo y se establece para cada uno de los cortes. El primer corte de la imagen tiene un indicador de corte establecido en un primer valor definido, mientras que los cortes

5 restantes tienen su indicador de corte respectivo establecido en un segundo valor definido. Las direcciones de corte que permiten la identificación de la posición de una primera unidad de codificación de un corte y, por lo tanto, el inicio de corte dentro de la imagen se generan para los cortes restantes. Se excluye el primer corte de la imagen. Estas direcciones de corte se incluyen junto con las representaciones de corte codificadas y los indicadores de corte en una representación de imagen codificada de la imagen. Las direcciones de corte son direcciones de longitud fija.

10 Otro aspecto de las realizaciones se refiere a un dispositivo para codificar una imagen que comprende múltiples cortes. Un generador de representación del dispositivo genera una representación de corte codificada respectiva para cada corte en la imagen. El dispositivo comprende un configurador de indicador configurado para establecer un indicador de corte asociado con un primer corte en la imagen a un primer valor, mientras que el(los) indicador(es) de corte de

15 la(s) división(es) restante(s) se establece(n) a un segundo valor definido. Un generador de direcciones genera una dirección de corte respectiva para cada corte de los cortes restantes para permitir la identificación de una posición respectiva de una primera unidad de codificación y el inicio de corte del corte dentro de la imagen. Un administrador de representación genera una representación de la imagen codificada para la imagen que comprende las representaciones de corte codificadas, las direcciones de corte y los indicadores de corte. Las direcciones de corte

20 son direcciones de longitud fija.

Las realizaciones proporcionan una gestión eficiente de cortes dentro de imágenes o fotogramas en términos que proporcionan una manera eficiente de señalar e identificar posiciones de inicio de corte dentro de una imagen o fotograma. Los indicadores de corte de las realizaciones proporcionan una identificación significativamente mejorada

25 de los inicios de corte para los primeros cortes en la imagen, pero sin necesidad de cálculo y señalización de direcciones de corte en el decodificador.

Breve descripción de los dibujos

30 La invención, junto con otros objetivos y ventajas de la misma, puede comprenderse de la mejor manera al referirse a la siguiente descripción, tomándola junto con los dibujos adjuntos, en los que:

la figura 1 es un diagrama de flujo que ilustra un procedimiento de codificación de una imagen según una

35 realización;

la figura 2 ilustra una realización de una imagen dividida en múltiples cortes y que comprende múltiples unidades máximas de codificación (LCU);

40 las figuras 3A y 3B ilustran realizaciones de un inicio de corte que coincide con un borde entre unidades de codificación dentro de una LCU;

la figura 4 ilustra esquemáticamente una realización de un orden de codificación y decodificación para procesar unidades de codificación;

45 la figura 5 es un diagrama de flujo que ilustra pasos adicionales del procedimiento de codificación en la figura 1 según una realización;

la figura 6 ilustra esquemáticamente una realización de una representación de imagen codificada;

50 la figura 7 es un diagrama de flujo que ilustra una realización de decodificar una representación codificada de una imagen;

la figura 8 es un diagrama de flujo que ilustra pasos adicionales del procedimiento de la figura 7 según una

55 realización.

la figura 9 es un diagrama de flujo que ilustra pasos adicionales del procedimiento de la figura 7 según una

60 realización;

la figura 11 es un diagrama de bloques esquemático de un dispositivo para decodificar una representación de

imagen codificada según una realización; y

la figura 12 es un diagrama de bloques esquemático de un terminal de medios según una realización.

5 DESCRIPCIÓN DETALLADA

En todos los dibujos se utilizan los mismos números de referencia para elementos similares o correspondientes.

Las realizaciones generalmente se refieren a la gestión de cortes dentro de imágenes o fotogramas. Con más detalle, las realizaciones proporcionan una forma flexible y algo eficiente de señalar direcciones de inicio de corte. Las realizaciones son aplicables a cualquier codificación y decodificación de imágenes o fotogramas en los que una imagen o fotograma, por ejemplo, una transmisión de vídeo, puede comprender múltiples, es decir, al menos dos, cortes y donde el inicio de los cortes debe señalizarse para el decodificador. Por lo tanto, la realización se puede aplicar a la codificación de imagen o vídeo de última generación, como H.264/MPEG-4 AVC, pero es particularmente aplicable a la codificación y decodificación de imágenes que utilizan la división jerárquica de bloques de píxeles en bloques más pequeños de píxeles. Por lo tanto, las realizaciones son muy adecuadas para usarse en conexión con la codificación de vídeo de alta eficiencia (HEVC), pero no están limitadas a las mismas.

Una característica de las realizaciones es diferenciar entre el primer corte en la imagen o fotograma y los cortes restantes. En la técnica anterior, se determina una dirección de corte durante la codificación para cada corte en la imagen y se incluye en la representación de imagen codificada de la imagen que utilizará un decodificador para identificar el inicio del corte dentro del área de la imagen o fotograma. Dependiendo del tamaño total de la imagen, estas direcciones de corte pueden ser bastante largas y, por lo tanto, agregar una cantidad considerable de sobrecarga a los datos codificados de la imagen. Por ejemplo, una imagen o fotograma H.264/MPEG-4 AVC podría consistir en 1280×960 píxeles. Si el inicio del corte está alineado con los bordes de macrobloque y el inicio del corte se expresa como coordenadas X e Y en relación con un origen fijo, generalmente la esquina superior izquierda de la imagen, la dirección del corte será $\log_2(\text{ceil}(1280/16))+\log_2(\text{ceil}(960/16))=7+6=13$ bits en este sencillo ejemplo. $\text{Ceil}(\)$ denota la función techo definida como $\text{ceil}(x) = \lceil x \rceil$ y genera el entero más pequeño no menor que x . Esto podría no ser una gran cantidad de datos, pero con las velocidades de fotogramas y bits de hoy en día para la decodificación y renderización de vídeo, generalmente se accede a varios cientos de cortes por segundo, por lo que la cantidad de tamaño total de datos de dirección de corte que se debe generar y reenviar al codificador para el decodificador es bastante significativa para una transmisión de vídeo. Además, las direcciones de corte deben ubicarse, recuperarse y procesarse en el decodificador antes de que los datos de píxeles reales de un corte puedan decodificarse y asignarse a la parte correcta de la imagen.

Los problemas mencionados anteriormente con el manejo de direcciones de corte pueden volverse aún más prominentes cuando se emplea la codificación y decodificación jerárquica, como en HEVC, si los inicios de corte pueden alinearse en unidades de codificación de un nivel inferior que las unidades máximas de codificación. La figura 2 ilustra esquemáticamente este concepto.

Una imagen 1, como la de un fotograma en una secuencia de vídeo, se puede dividir en varias unidades máximas de codificación (LCU) 10, también llamadas bloques de árbol de codificación máximos (LCTB) o bloques de codificación máximos (LCB) en la técnica. La LCU 10 es el bloque de píxeles más grande posible que puede manejarse durante la codificación y decodificación, y puede procesarse, por ejemplo, según los bien conocidos modos de codificación/decodificación intra o inter. A su vez, la LCU 10 se puede dividir jerárquicamente en un número de bloques de píxeles más pequeños, en términos de número de píxeles, típicamente unidades de codificación (CU) 20A, 20B, bloques de árbol de codificación (CTB) o bloques de codificación (CB). Estas CU 20A, 20B pueden dividirse a su vez en bloques aún más pequeños 30A, 30B de píxeles de manera jerárquica hasta un bloque de píxeles lo más pequeño posible, denotado bloque de árbol de codificación mínimo (SCTB) de la unidad mínima de codificación (SCU) o bloque mínimo de codificación (SCB).

En claro contraste con las técnicas de la técnica anterior que limitan el posicionamiento del inicio del corte 2, 3 en una imagen 1 para alinearse con el borde entre dos LCU adyacentes 10A, 10B, algunas realizaciones proporcionan un posicionamiento de inicio del corte mucho más flexible al potencialmente permitir que el inicio del corte 4, 5 se alinee con el borde entre cualquier CU adyacente 10A, 10B, 20A, 20B, 30A, 30B en la imagen desde el nivel de SCU hasta el nivel de LCU. Por lo tanto, es posible que el inicio del corte se coloque dentro de una LCU con la última CU del corte anterior y la primera CU del corte actual en la misma LCU. La figura 2 ilustra esto esquemáticamente. El borde entre un primer corte 2 y un segundo corte 3 coincide con el borde de dos LCU 10A, 10B adyacentes, según el orden de procesamiento. El borde entre el segundo corte 3 y un tercer corte 4 se alinea con el borde de dos unidades de codificación 20A, 20B, que tienen un cuarto de tamaño cada una en comparación con una LCU 10. De manera correspondiente, el borde entre el tercer corte 4 y un cuarto corte 5 está en la figura 2 alineado con el borde de dos CU 30A, 30B, cada una de 1/16 de tamaño en comparación con una LCU 10.

Sin embargo, la división jerárquica y los inicios de cortes alineados con CU pequeñas, posiblemente incluso las SCU, pueden dar como resultado direcciones de corte aún más largas en comparación con MPEG-4/AVC. Por ejemplo, y el ejemplo discutido anteriormente, una imagen de 1280×960 píxeles puede tener 19200 posiciones de inicio de corte potencial si el inicio de corte puede alinearse con el borde de las SCU de 8×8 píxeles. Si las direcciones de corte están en forma de coordenadas X e Y, tendrían que ser $8+7=15$ bits.

De acuerdo con las realizaciones, «corte» se emplea para denotar una porción independientemente codificable y decodificable de una imagen o fotograma. De este modo, una imagen puede estar compuesta de un solo corte o múltiples, es decir, al menos dos cortes.

La figura 1 es un diagrama de flujo que ilustra un procedimiento de codificación de una imagen o fotograma que comprende N cortes. El parámetro N es un entero positivo igual o mayor que dos. El procedimiento comienza en la etapa S1, que genera una representación de corte codificada para cada corte en la imagen. Esta representación de corte codificada se genera en función de los valores de píxel de los píxeles en el corte según esquemas de codificación bien conocidos, tales como codificación intra o inter.

Los siguientes pasos S2 y S3 establecen los llamados indicadores de corte para los cortes en la imagen. Con más detalle, el paso S2 establece un indicador de corte asociado con el primer segmento de los cortes N en la imagen a un primer valor definido, como 1_{bin} o uno lógico o algún otro símbolo definido, para indicar que el corte actual es el primer corte de la imagen y, por lo tanto, su corte comienza en una posición definida en la imagen, habitualmente la esquina superior izquierda de la imagen. La etapa S3 establece correspondientemente el indicador de corte asociado con cada corte de los cortes $N-1$ restantes de la imagen a un segundo valor definido, como 0_{bin} o un cero lógico o algún otro símbolo definido. Esto significa que los indicadores de corte pueden usarse como información adicional para diferenciar entre el primer corte de una imagen y los cortes restantes.

Estos cortes restantes deberán tener direcciones de corte asociadas para permitir que el decodificador identifique dónde comienza el corte dentro del área de la imagen. Esto no es necesario para el primer corte, que en su lugar tiene una posición de inicio definida dentro de la imagen y preferiblemente comienza en el primer píxel de la imagen en la esquina superior izquierda. Alternativamente, una imagen se puede dividir en regiones en un nivel más alto que los cortes. En tal caso, la posición de inicio definida podría ser el origen de dicha región en la imagen.

La etapa S4, por lo tanto, genera una dirección de corte para cada corte de los cortes $N-1$ restantes, por lo que se excluye el primer corte para el que no se requiera una dirección de corte según las realizaciones. La dirección de corte generada en la etapa S4 para un corte permite la identificación de una posición de una primera unidad de codificación del corte y, por lo tanto, el corte comienza dentro de la imagen. La primera unidad de codificación constituye entonces la primera unidad de codificación del corte y una unidad de codificación precedente, según un orden de procesamiento definido, es la última unidad de codificación del corte anterior en la imagen. La primera unidad de codificación podría ser cualquier bloque de píxeles y en MPEG-4/AVC la primera unidad de codificación es un primer macrobloque del corte. En consecuencia, con HEVC la primera unidad de codificación podría ser una LCU, pero con la división jerárquica puede ser ventajosamente cualquier unidad de codificación desde el tamaño de LCU hasta un tamaño de SCU, a menos que se impongan algunas limitaciones de dónde se pueden encontrar las posiciones de inicio de corte en la imagen, lo que se discute adicionalmente en la presente descripción.

El orden de procesamiento en el que se procesan las unidades de codificación de la imagen, es decir, codificadas y posteriormente decodificadas, podría ser cualquier orden de procesamiento conocido. Un ejemplo de dicho orden de procesamiento es el orden de exploración de ráster o cualquier otro orden de codificación/decodificación, como el orden de Morton o Z, que se trata más adelante en este documento.

Las N representaciones de corte codificadas, las $N-1$ direcciones de corte y los N indicadores de corte se emplean para generar una representación de imagen codificada de la imagen en la etapa S5. La representación de imagen codificada está típicamente en forma de una secuencia o secuencia de bits, aunque podrían usarse otros alfabetos de símbolos además del alfabeto binario y están dentro del alcance de las realizaciones, tales como el alfabeto hexadecimal o decimal. La figura 6 es una ilustración esquemática de una realización de tal representación de imagen codificada. Generalmente, la representación de imagen codificada comprende dos partes principales para cada corte, un encabezado de corte y datos codificados. El encabezado de corte típicamente comprende el indicador de corte establecido para el corte en la etapa S2 o S3, como en la forma de la palabra de código *first_slice_in_pic_flag*. El encabezado de corte 54 de cada segmento restante, excepto el primer corte en la imagen, preferiblemente también comprende la dirección de corte generada en la etapa S4, tal como en forma de una palabra de código *first_cu_in_slice* o la palabra de código *slice_address*. En realizaciones particulares, se puede incluir información adicional en el encabezado de corte 54, que incluye, por ejemplo, el tipo de codificación del corte.

Los datos codificados 56 llevan entonces los datos de imagen codificados de los píxeles en el corte, es decir, las representaciones de corte codificadas generadas en la etapa S1.

La representación codificada 50 también puede comprender opcionalmente o estar asociada con un conjunto de parámetros de imagen (PPS) y/o un conjunto de parámetros de secuencia (SPS) 52. El PPS/SPS 52 podría formar parte de la representación de imagen codificada 50. En tal caso, cada representación de imagen codificada 50 de una secuencia de vídeo podría tener un campo PPS y/o SPS respectivo 52. En una estrategia alternativa, no todas las representaciones de imagen codificadas 50 de la transmisión de vídeo necesitan llevar el campo PPS y/o SPS 52. Por ejemplo, la primera representación de imagen codificada 50 de la transmisión de vídeo podría incluir el campo PPS y/o SPS 52 y luego dichos campos solo se incluyen en otra representación de imagen codificada de la transmisión de vídeo si alguno de los parámetros en el campo PPS y/o SPS 52 se actualiza o cambia. Una variante adicional es señalar el campo PPS y/o SPS 52 fuera de banda con respecto a la presentación de imagen codificada 50. En tal caso, el campo PPS y/o SPS 52 podría enviarse por separado de la representación de imagen codificada 50 pero de tal manera que el decodificador pueda identificar a qué transmisión de vídeo o representación de imagen codificada pertenece el campo PPS y/o SPS 52. Esto se puede lograr mediante la inclusión de un identificador de sesión, transmisión y/o imagen en la representación de imagen codificada 50 y en el campo PPS y/o SPS 52.

Como se mencionó anteriormente, los cortes son unidades independientemente codificables y decodificables de la imagen. Esto significa que la generación de representaciones de corte codificadas en la etapa S1, la configuración del indicador de corte de las etapas S2, S3 y la generación de direcciones de la etapa S4 se pueden realizar en serie o al menos en parte en paralelo para los distintos cortes en la imagen. Una codificación paralela de los cortes típicamente reducirá el tiempo total de codificación de la imagen. El procedimiento de las etapas S1 a S5 generalmente se repite para las imágenes o fotogramas restantes, como por ejemplo, una transmisión de vídeo. Además, la etapa S2 o las etapas S3/S4 se pueden realizar después, antes o al menos parcialmente en paralelo con la etapa S1.

A continuación, las realizaciones de la presente invención se describirán adicionalmente en relación con HEVC como un ejemplo de un estándar de codificación y decodificación de vídeo al que se pueden aplicar las realizaciones. Sin embargo, esto debería verse simplemente como un ejemplo ilustrativo del estándar de codificación/decodificación de imagen o vídeo que puede usarse con las realizaciones y las realizaciones no están limitadas al mismo.

Según HEVC, una imagen o fotograma comprende múltiples LCU que tienen un tamaño seleccionado en términos de número de píxeles. Esto significa que cada LCU de la imagen tiene preferiblemente el mismo número de píxeles. Las LCU podrían ser rectangulares pero preferiblemente son cuadráticas, es decir, comprenden $M \times M$ píxeles, donde M es un entero positivo definido igual o preferiblemente mayor que dos y preferiblemente $M=2^m$, donde m es un entero positivo. Ejemplos no limitativos de valores adecuados de M son 64 o 128. Cada LCU de la imagen puede dividirse jerárquicamente en múltiples CU más pequeñas que tienen tamaños respectivos que son más pequeños que el tamaño seleccionado de las LCU.

En general, dividir jerárquicamente una LCU implica dividir la LCU de forma cuadrática. Como es bien sabido en la técnica, un quadtree es una estructura de datos en árbol en la que cada nodo interno tiene exactamente cuatro hijos. La división jerárquica de la LCU implica dividir el espacio bidimensional de la imagen ocupada por la LCU al subdividirla recursivamente en cuatro cuadrantes o regiones. En una realización preferida, la división recursiva implica la división en cuatro CU de igual tamaño. De acuerdo con las realizaciones, si una unidad de codificación, es decir, ya sea una LCU o una CU más pequeña, se divide, un denominado indicador de unidad de codificación de corte asociado con la unidad de codificación se establece a un valor definido, preferiblemente 1_{bin} o un símbolo lógico o cualquier otro definido, lo que indica que la unidad de codificación se divide jerárquicamente en múltiples CU más pequeñas, preferiblemente cuatro. En consecuencia, si una unidad de codificación divisible, es decir, una unidad de codificación que es más grande que la SCU, no se divide, un indicador de unidad de codificación de corte asociado con la unidad de codificación se establece preferiblemente en 0_{bin} o un cero lógico o cualquier otro símbolo definido. La unidad de codificación «divisible» se refiere aquí a una unidad de codificación que es capaz de dividirse jerárquicamente en múltiples unidades de codificación más pequeñas, preferiblemente cuatro. Generalmente, cualquier unidad de codificación es una unidad de codificación divisible, excepto las SCU. Aunque una unidad de codificación se puede dividir en unidades de codificación más pequeñas, no es necesario dividirla, por ejemplo, si dicha división no mejorase la calidad de codificación de la imagen.

La división jerárquica de las realizaciones procesa preferiblemente LCU por LCU en un orden de procesamiento definido, tal como el orden de exploración de ráster. El orden de exploración de ráster generalmente es de izquierda a derecha y de arriba a abajo. Alternativamente, se podría usar un orden de codificación/decodificación, como el orden de Morton o Z. La figura 4 ilustra los principios del orden de Morton. Si una LCU se divide en, preferiblemente, cuatro CU de igual tamaño, estas CU pueden procesarse luego en un orden de procesamiento para seleccionar si deben dividirse jerárquicamente en, preferiblemente, cuatro CU de igual tamaño o incluso más pequeñas. Este orden de procesamiento podría ser el mismo que cuando se procesan las LCU en la imagen. En una estrategia alternativa, las

LCU se procesan en el orden de exploración de ráster y las CU se procesan en el orden de codificación/decodificación, como el orden de Morton. Las órdenes de procesamiento presentadas anteriormente son simplemente ejemplos de órdenes que pueden usarse y las realizaciones no están limitadas a las mismas.

- 5 Por lo tanto, para cada unidad de codificación se determina si dividir la unidad de codificación en unidades de codificación más pequeñas a menos que la unidad de codificación sea la SCU, que no puede dividirse más jerárquicamente. Cada vez que una unidad de codificación se divide, un indicador de unidad de codificación de corte asociado con la unidad de codificación se establece preferiblemente en uno y si se determina que una unidad de codificación no se divide más en una unidad de codificación más pequeña, su indicador de unidad de codificación de corte asociado se establece preferiblemente en cero. Una SCU normalmente no necesita tener ningún indicador de unidad de codificación de corte asociado, ya que no se puede dividir más.

15 Esta decisión de dividir una unidad de codificación se basa en el proceso de codificación. Por ejemplo, un área de imagen que representa un fondo bastante homogéneo se representa de manera más eficiente utilizando grandes tamaños de CU, como LCU, en comparación con la división del área de imagen en unidades de codificación más pequeñas. Sin embargo, las áreas de imagen con pequeños detalles o muchos de esos detalles generalmente no se pueden representar correctamente si se utilizan unidades de codificación grandes. En tal caso, es más eficiente y preferido desde el punto de vista de la calidad de codificación usar varias CU más pequeñas para el área de la imagen. La selección de si dividir adicionalmente una CU puede realizarse según las técnicas descritas en la técnica y 20 preferiblemente en base a la eficiencia y calidad de codificación.

Los indicadores de la unidad de codificación de corte generados durante la codificación de los cortes en la etapa S1 de la figura 1 se incluyen típicamente en la porción de datos codificados 56 de la representación de imagen codificada 50 como se ilustra en la figura 6.

25 La figura 5 es un diagrama de flujo que ilustra etapas adicionales del procedimiento de codificación de la figura 1. El procedimiento comienza en la etapa opcional S10, donde se determina una granularidad jerárquica para la imagen. La granularidad jerárquica define un nivel jerárquico para las alineaciones de bordes de corte dentro de la imagen. Este nivel jerárquico define y limita el tamaño de una unidad de codificación direccionable mínima posible en la que se puede alinear el inicio de un segmento en la imagen. El nivel jerárquico y la granularidad jerárquica determinada definen así el número máximo de posiciones de inicio de corte que están potencialmente disponibles en la imagen y en las que se puede colocar un inicio de corte. Esto significa que la granularidad jerárquica define el número de CU direccionables dentro de la imagen, donde el inicio de un corte se puede alinear entre el borde de una CU direccionable y una CU anterior en la imagen, según un orden de procesamiento definido.

35 Por ejemplo, una LCU que tiene un tamaño seleccionado de 64×64 píxeles podría tener una granularidad de corte que define un nivel de granularidad de 0 a 3 con un tamaño de SCU de 8×8 píxeles. En tal caso, un nivel de granularidad de 0 indica que los inicios de corte solo se pueden alinear con los bordes entre las LCU. Con una imagen de 1280×960 píxeles, esto implica $20 \times 15 = 300$ posibles posiciones de inicio de corte. Si el nivel de granularidad es 1, la unidad de codificación más pequeña posible en la que se puede alinear un inicio de corte es 32×32 píxeles con un total de $40 \times 30 = 1200$ posibles posiciones de inicio de corte. En consecuencia, un nivel de granularidad de 2 significa que hay $80 \times 60 = 4800$ posibles posiciones de inicio de corte, ya que el inicio de corte puede alinearse con CU de 16×16 píxeles o más. Finalmente, un nivel de granularidad de 3 indica que los inicios de corte pueden alinearse con los bordes de la SCU, lo que da un total de $160 \times 120 = 19200$ posibles posiciones de inicio de corte.

45 Una razón por la que uno quisiera tener la posibilidad de seleccionar entre estos casos de 220 a 14400 posibles posiciones de inicio de corte en el presente ejemplo, es que cuantas más posiciones de inicio de corte estén disponibles en la imagen, más largas serán la direcciones de corte necesarias, lo que aumenta la sobrecarga de los datos de la imagen codificada. Por lo tanto, si no hay demandas específicas sobre los tamaños de corte objetivo, como 50 ajustar los datos de corte en un solo paquete de datos IP, podría ser ventajoso limitar el número de posiciones de inicio de corte en una imagen para reducir así la cantidad de datos de dirección que necesita ser generados y transmitidos al decodificador.

La siguiente etapa S11 determina la longitud de la dirección de corte para los cortes $N-1$ en la imagen, donde la 55 longitud es en términos del número de bits u otros símbolos de la dirección de corte. La longitud de la dirección de corte depende de la cantidad de posibles posiciones de inicio de corte y la cantidad de unidades de codificación direccionables dentro de la imagen. En el caso de MPEG-4/AVC, este número de posiciones de inicio de corte depende del tamaño de la imagen, ya que el inicio de corte solo se puede alinear en los bordes de macrobloque. Esto significa que, dado el tamaño total de la imagen, el número de posibles posiciones de inicio de corte se puede calcular dado el tamaño fijo de macrobloque. La longitud de la dirección de corte se puede calcular a partir de este número, como $\log_2(P)$ o $\log_2(P-1)$, donde P representa el número de posibles posiciones de inicio de corte y, por lo tanto, el número 60 total de posibles direcciones de corte en la imagen. El tamaño de la imagen se incluye habitualmente en un campo de

encabezado asociado con las representaciones de imagen codificadas o se puede encontrar en el campo PPS o SPS 52 mencionado anteriormente o asociado con la representación de imagen codificada 50, ver figura 6.

En HEVC, la longitud de la dirección de corte se determina preferiblemente en la etapa S11 en función de la 5 granularidad jerárquica determinada en la etapa S10 para la imagen. La granularidad jerárquica se puede utilizar para definir el tamaño de las unidades de codificación direccionables y, por lo tanto, el tamaño de unidad de codificación más pequeño posible en el que se puede alinear un inicio de corte. Esta información de granularidad se emplea preferiblemente junto con información del tamaño de la imagen o del número total de LCU en la imagen, para determinar la longitud de la dirección de corte en la etapa S11.

10 En una realización, el tamaño de las LCU en la imagen podría predefinirse y, por lo tanto, ser conocido por el codificador y el decodificador. Por ejemplo, el tamaño de la LCU podría ser de 128×128 píxeles o 64×64 píxeles. Por lo tanto, no se necesita determinación ni señalización del tamaño de LCU. En consecuencia, el tamaño de las SCU en la imagen podría estar predefinido. Ejemplos de tales tamaños de SCU fijos y predefinidos que pueden emplearse son 15 16×16 píxeles u 8×8 píxeles.

En realizaciones alternativas, el proceso de codificación puede determinar adicionalmente el tamaño de LCU y/o el tamaño de SCU a emplear para las imágenes actuales. Esto podría ser beneficioso para adaptar así estos tamaños de LCU y/o SCU a las características particulares de la imagen actual. Por ejemplo, para algunas imágenes que son 20 básicamente una vista de fondo homogénea uniforme, se podrían preferir tamaños de LCU y SCU más grandes y conducir a una codificación más eficiente en comparación con imágenes con muchos detalles pequeños donde se podrían preferir tamaños de LCU y SCU más pequeños.

En una realización, el tamaño de LCU y/o el tamaño de SCU se determinan, por lo tanto, durante la codificación, tal 25 como en base a los valores de píxeles de la imagen. Una notificación del tamaño de LCU determinado y/o una notificación del tamaño de SCU determinado se asocia entonces con la representación de imagen codificada. La asociación de la(s) notificación(es) y la representación de imagen codificada se puede realizar según diversas realizaciones. Por ejemplo, las notificaciones se pueden incluir en la representación de imagen codificada. Una alternativa es incluir las notificaciones en el PPS o SPS.

30 El tamaño de la SCU podría definirse según el parámetro *log2_min_coding_block_size_minus3* y preferiblemente calculando el parámetro *Log2MinCUSize* como $Log2MinCUSize = log2_min_coding_block_size_minus3 + 3$. Este parámetro *Log2MinCUSize* luego se emplea como representación del tamaño de la SCU y da el tamaño de la SCU $MinCUSize = (1 \ll Log2MinCUSize)$, donde \ll denota un operador de desplazamiento a la izquierda. Dependiendo del 35 valor del parámetro *Log2MinCUSize* y, por lo tanto, en el parámetro *log2_min_coding_block_size_minus3*, el tamaño de la SCU podría ser 8 o 16 como ejemplos ilustrativos.

El tamaño de la LCU se determina preferiblemente en relación con el tamaño de la SCU definiendo el parámetro *log2_diff_max_min_coding_block_size*. Más detalladamente, el parámetro *Log2MaxCUSize* se calcula como 40 $Log2MaxCUSize = log2_min_coding_block_size_minus3 + 3 + log2_diff_max_min_coding_block_size$. Este parámetro *Log2MaxCUSize* se emplea como representación del tamaño de la LCU y proporciona el tamaño de la LCU $MaxCUSize = (1 \ll Log2MaxCUSize)$. Dependiendo del valor del parámetro *Log2MaxCUSize* y, por lo tanto, en los parámetros *log2_min_coding_block_size_minus3* y *log2_diff_max_min_coding_block_size* el tamaño de la LCU podría ser 64 o 128 como ejemplos ilustrativos.

45 Las notificaciones del tamaño de la SCU y el tamaño de la LCU podrían ser los parámetros *log2_min_coding_block_size_minus3* y *log2_diff_max_min_coding_block_size*.

En una realización alternativa, el tamaño de la LCU no se determina en relación con el tamaño de la SCU. Esto significa 50 que no se necesita ningún parámetro de SCU para determinar el tamaño de la LCU.

La dirección del corte generada para cada corte, excepto por el primer corte de la imagen, podría definir la posición del inicio de corte y la primera CU del corte como un número simple. Las diferentes posiciones de inicio de corte posibles y las unidades de codificación direccionables se numeran de cero en adelante. Por ejemplo, una imagen de 55 1280×960 píxeles tiene 4800 posiciones de inicio de corte únicas si la granularidad jerárquica define que el tamaño de la unidad de codificación más pequeña posible en la que se puede alinear un inicio de corte en la imagen es de 16×16 píxeles. Estas posiciones podrían numerarse desde 0 hasta 4799, lo que requeriría direcciones de corte de 13 bits.

60 Una alternativa es manejar las coordenadas X e Y por separado. Con el ejemplo anterior, la coordenada X es de 0 a 79 y la coordenada Y es de 0 a 59, lo que requiere 7 más 6 bits para las direcciones de corte.

Una alternativa adicional es determinar la dirección de corte para que las coordenadas de la LCU y las coordenadas de la sub-LCU puedan recuperarse a partir de ellas. En tal caso, las coordenadas de una posición de una LCU dentro de la imagen se determinan para un corte. El inicio de corte y la primera CU del corte se colocan en la imagen dentro de esta LCU. Las coordenadas están entonces en relación con un origen global o punto de inicio, típicamente la esquina superior izquierda de la imagen. Las coordenadas de la LCU podrían ser las coordenadas de la LCU en relación con este origen global, como en términos de un número de LCU o en términos de coordenadas X e Y como se mencionó anteriormente. También se determinan las coordenadas de la posición de la primera CU y, por lo tanto, el inicio del corte dentro de la LCU. Estas coordenadas son entonces relativas a un origen local o punto de inicio, habitualmente la esquina superior izquierda de la LCU. Estas coordenadas sub-LCU también podrían estar en forma de un número o en términos de coordenadas X e Y.

La dirección de división se genera luego en base a las coordenadas de la LCU y las coordenadas de la sub-LCU y la granularidad jerárquica. La granularidad jerárquica se emplea al definir las coordenadas de la sub-LCU restringiendo y definiendo las posibles posiciones de inicio para el corte y la primera CU del corte dentro de la LCU.

En una realización, se genera una primera representación o LCU en base a las coordenadas de LCU determinadas y se genera una segunda representación o sub-LCU en base a las coordenadas de sub-LCU. La dirección de corte podría comprender estas dos representaciones. Alternativamente, la dirección de corte se genera de tal manera que las coordenadas de LCU y las coordenadas de sub-LCU se pueden determinar o calcular a partir de la dirección de corte.

La derivación de las coordenadas LCU y sub-LCU podría realizarse según lo siguiente como un ejemplo no limitativo pero ilustrativo.

La granularidad jerárquica determinada en la etapa S10 está definida por la palabra de código *slice_granularity*. *Slice_granularity* es típicamente un valor de 2 bits que va desde $00_{bin}=0$ hasta un máximo de $11_{bin}=3$. Esto permite cuatro niveles jerárquicos distintos. Si solo se necesitan dos niveles jerárquicos *slice_granularity* podría en cambio ser un valor de 1 bit. En consecuencia, para más de cuatro niveles jerárquicos es necesario un parámetro *slice_granularity* de 3 bits o más largo. Alternativamente, la codificación de longitud variable es posible para señalar la granularidad jerárquica.

La palabra de código *slice_granularity* se define opcionalmente para que no sea mayor que el mínimo de otras dos palabras de código determinadas durante la codificación de imagen o transmisión de vídeo: $Log2MaxCUSize-4$ y $log2_diff_max_min_coding_block_size$. La palabra de código *slice_granularity* luego, durante la decodificación, se emplea para calcular el parámetro *SliceGranularity* como $SliceGranularity=(slice_granularity<<1)$.

La dirección de corte generada durante la codificación es la palabra de código *slice_address*. Esta palabra de código define la dirección del corte en la resolución de granularidad de corte en la que comienza el corte. La longitud de la dirección del corte, es decir *slice_address*, es como se mencionó anteriormente determinada en función de la granularidad jerárquica. En una realización particular, la longitud de la *slice_address* en términos de número de bits es igual a $ceil(log_2(NumLCUsInPicture) + SliceGranularity)$.

El parámetro *NumLCUsInPicture* define el número total de LCU en la imagen y se determina en función del tamaño de la imagen y en función del tamaño de las LCU, que es fijo o determinado como se menciona anteriormente. *NumLCUsInPicture* entonces puede incluirse en la representación de imagen codificada o asociarse a ella, como en un campo PPS o SPS. Alternativamente, un decodificador puede calcular por sí mismo el parámetro *NumLCUsInPicture* basado en el tamaño de la LCU ($log2_min_coding_block_size_minus3$ y $log2_diff_max_min_coding_block_size$) y el tamaño total de la imagen, que se señala al decodificador o se asocia con la representación de imagen codificada.

La parte de la LCU de la dirección de corte según un orden de procesamiento, como el orden de exploración de ráster, se define como $LCUAddress=(slice_address>>SliceGranularity)$, donde $>>$ denota un operador de desplazamiento a la derecha.

La parte sub-LCU de la dirección de corte según un orden de procesamiento, como el orden de Morton, se calcula como $GranularityAddress=slice_address - (LCUAddress<<SliceGranularity)$.

La dirección de corte se determina en función de la *LCUAddress* y la *GranularityAddress* como $SliceAddress=(LCUAddress<<(log2_diff_max_min_coding_block_size<<1)) + (GranularityAddress<<((log2_diff_max_min_coding_block_size<<1) - SliceGranularity))$.

- La dirección de corte generada para los cortes restantes, excluyendo el primer corte de la imagen, podría ser una dirección de longitud fija, donde la longitud de la dirección se fija para una imagen y depende del tamaño de una unidad de codificación más pequeña posible en la que se inicie un corte en la imagen se puede alinear y el tamaño total de la imagen. Una alternativa sería utilizar una representación de longitud variable. Un ejemplo de un código de longitud variable que se puede usar es la codificación universal de longitud variable (UVLC) como se menciona en Lee and Kuo, Complexity Modeling of H.264/AVC CAVLC/UVLC Entropy Decoders, IEEE International Symposium on Circuits and Systems (ISCAS2008), 2008, pp. 1616-1619. Brevemente, UVLC utiliza el código Exp-Golomb (EG). El código EG para un valor entero sin signo C es $[P \text{ zeros}][1][\text{info}]$, donde $P = \text{floor}(\log_2(C+1))$ and $\text{info} = C + 1 - 2^P$.
- 10 La dirección de corte no solo define la posición de la primera CU y, por lo tanto, el inicio de un corte, sino que además define el tamaño de la CU más grande posible que puede ocupar la posición en la imagen definida por la dirección de corte. Esto significa que este tamaño depende de la posición determinada por la dirección de corte. Sin embargo, la dirección de corte proporciona el tamaño de la CU más grande posible que puede ocupar la posición, el tamaño de la primera CU no necesita ser igual al tamaño de esta CU más grande posible que puede ocupar la posición. Las figuras 15 3A y 3B ilustran este concepto. En las figuras, los números de referencia 2, 3 denotan dos cortes distintos en una imagen y la línea en negrita define el borde entre los dos sectores 2, 3. El borde de corte se produce en estos ejemplos dentro del área de la imagen ocupada por una LCU 10. El número de referencia 20 denota una CU obtenida para una granularidad de 1 cuando la LCU 10 se divide jerárquicamente en cuatro CU 20. Con una granularidad de 2, este CU 20 se puede dividir jerárquicamente en cuatro CU 30 más pequeñas. En las figuras 3A y 3B que muestran un caso 20 con una granularidad de 3, una CU 30 se puede dividir en cuatro SCU 40.

En la figura 3A, la primera CU del segmento 3 está referenciada por 30, mientras que en la figura 3B está referenciada por 40B. Los números de referencia 40 (figura 3A) y 40A (figura 3B) denotan una CU precedente de la LCU 10 según el orden de procesamiento definido, que en este ejemplo es el orden de Morton. En ambas figuras 3A y 3B, el inicio 25 del corte y la posición de las primeras CU 30, 40B son las mismas, aunque las primeras CU 30, 40B tienen tamaños distintos en los dos ejemplos. Sin embargo, la dirección de corte es habitualmente la misma en ambos casos y el tamaño de la CU 30 máxima posible que puede ocupar la posición relevante es el mismo. Los dos casos se pueden diferenciar al complementar la dirección de corte con información adicional en términos de los llamados indicadores de unidad de codificación de corte.

30 En un ejemplo, suponga que el tamaño de una LCU es de 128×128 píxeles y que el tamaño correspondiente de una SCU es de 16×16 píxeles. Además, suponga que las LCU 10 de las figuras 3A y 3B constan de dos cortes 2, 3, entonces la representación codificada podría definirse como:

Slice_header_syntax() // corte 2 en la figura 3A – incluye la información de dirección

split_coding_unit_flag=1 // dividir LCU de 128x128 en cuatro CU de 64x64

split_coding_unit_flag=0 // primera CU de 64x64 no se sigue dividiendo

code of first 64x64 CU // la representación codificada de los valores de píxel de la primera CU de 64x64

split_coding_unit_flag=1 // dividir segunda CU de 64x64 en cuatro CU de 32x32

split_coding_unit_flag=0 // primera CU de 32x32 no se sigue dividiendo

code of first 32x32 CU // la representación codificada de los valores de píxel de la primera CU de 32x32

split_coding_unit_flag=1 // dividir segunda CU de 32x32 en cuatro SCU, no se necesitan más indicadores de unidad de codificación de corte

code of first SCU // la representación codificada de los valores de píxel de la primera SCU

code of second SCU // la representación codificada de los valores de píxel de la segunda SCU

code of third SCU // la representación codificada de los valores de píxel de la tercera SCU

code of fourth SCU // la representación codificada de los valores de píxel de la cuarta SCU

split_coding_unit_flag=1 // dividir tercera CU de 32x32 en cuatro SCU, no se necesitan más indicadores de unidad de codificación de corte

code of first SCU // la representación codificada de los valores de píxel de la primera SCU

code of second SCU // la representación codificada de los valores de píxel de la segunda SCU

code of third SCU // la representación codificada de los valores de píxel de la tercera SCU

code of fourth SCU // la representación codificada de los valores de píxel de la cuarta SCU

split_coding_unit_flag=1 // dividir cuarta CU de 32x32 en cuatro SCU, no se necesitan más indicadores de unidad de codificación de corte

code of first SCU // la representación codificada de los valores de píxel de la primera SCU

code of second SCU // la representación codificada de los valores de píxel de la segunda SCU

code of third SCU // la representación codificada de los valores de píxel de la tercera SCU

code of fourth SCU // la representación codificada de los valores de píxel de la cuarta SCU

split_coding_unit_flag=1 // dividir tercera CU de 64x64 en cuatro CU de 32x32

split_coding_unit_flag=1 // dividir primera CU de 32x32 en cuatro SCU, no se necesitan más indicadores de unidad de codificación de corte

code of first SCU // la representación codificada de los valores de píxel de la primera SCU

code of second SCU // la representación codificada de los valores de píxel de la segunda SCU

code of third SCU // la representación codificada de los valores de píxel de la tercera SCU

code of fourth SCU // la representación codificada de los valores de píxel de la CU 40 anterior de la figura 3A

Slice_header_syntax() // corte 3 de la figura 3A – incluye información de dirección

split_coding_unit_flag=0 // el indicador de unidad de codificación de corte de la primera CU se establece en cero ya que el tamaño de la primera CU es igual al tamaño de la CU máxima posible, que puede ocupar la posición en la imagen definida por la información de dirección generada. La CU máxima posible en esta dirección es 32x32 y no se necesitan indicadores de corte que se dividan en 32x32. En cambio, el tamaño 32x32 se deriva de la dirección y la granularidad.

code of first SCU // la representación codificada de los valores de píxel de la CU 30 de la figura 3ª

split_coding_unit_flag=1 // dividir CU de 32x32 en cuatro SCU, no se necesitan más indicadores de unidad de codificación de corte

code of first SCU // la representación codificada de los valores de píxel de la primera SCU

code of second SCU // la representación codificada de los valores de píxel de la segunda SCU

code of third SCU // la representación codificada de los valores de píxel de la tercera SCU

code of fourth SCU // la representación codificada de los valores de píxel de la cuarta SCU

split_coding_unit_flag=1 // dividir CU de 32x32 en cuatro SCU, no se necesitan más indicadores de unidad de codificación de corte

code of first SCU // la representación codificada de los valores de píxel de la primera SCU

code of second SCU // la representación codificada de los valores de píxel de la segunda SCU

code of third SCU // la representación codificada de los valores de píxel de la tercera SCU

code of fourth SCU // la representación codificada de los valores de píxel de la cuarta SCU

split_coding_unit_flag=0 // no se sigue dividiendo la CU de 64x64

code of 64x64 CU // la representación codificada de los valores de píxel de la primera CU de 64x64

En la realización ilustrada en la figura 3B, el código para el primer corte 2 sería el mismo que el anterior, mientras que para el segundo corte 3 el código sería:

5

Slice_header_syntax() // corte 3 de la figura 3B – incluye información de dirección

split_coding_unit_flag=1 // el indicador de la unidad de codificación de corte de la primera CU se establece en uno ya que el tamaño de la primera CU es más pequeño que el tamaño de la CU máxima posible que puede ocupar la posición en la imagen definida por la información de dirección generada, no se necesitan más indicadores de unidad de codificación de corte.

code of first CU // la representación codificada de los valores de píxel de la primera CU 40B de la figura 3B

code of second SCU // la representación codificada de los valores de píxel de la segunda SCU
code of third SCU // la representación codificada de los valores de píxel de la tercera SCU
code of fourth SCU // la representación codificada de los valores de píxel de la cuarta SCU
split_coding_unit_flag=1 // dividir CU de 32x32 en cuatro SCU, no se necesitan más indicadores de unidad de codificación de corte
code of first SCU // la representación codificada de los valores de píxel de la primera SCU
code of second SCU // la representación codificada de los valores de píxel de la segunda SCU
code of third SCU // la representación codificada de los valores de píxel de la tercera SCU
code of fourth SCU // la representación codificada de los valores de píxel de la cuarta SCU
split_coding_unit_flag=1 // dividir CU de 32x32 en cuatro SCU, no se necesitan más indicadores de unidad de codificación de corte
code of first SCU // la representación codificada de los valores de píxel de la primera SCU
code of second SCU // la representación codificada de los valores de píxel de la segunda SCU
code of third SCU // la representación codificada de los valores de píxel de la tercera SCU
code of fourth SCU // la representación codificada de los valores de píxel de la cuarta SCU
split_coding_unit_flag=0 // no se sigue dividiendo la CU de 64x64
code of 64x64 CU // la representación codificada de los valores de píxel de la primera CU de 64x64

La figura 7 es un diagrama de flujo que ilustra un procedimiento de decodificación de una representación de imagen codificada de una imagen que comprende múltiples cortes. El procedimiento comienza en la etapa S20, donde se recupera un indicador de corte asociado con un corte de la presentación de imagen codificada, preferiblemente desde un encabezado de corte asignado al corte actual en la presentación de imagen codificada. Una siguiente etapa S21 genera una representación decodificada de valores de píxel de los píxeles en el corte basándose en una representación de corte codificada asociada con el corte y recuperada de la representación de imagen codificada, típicamente de la porción de datos codificados de la misma. Las representaciones decodificadas de los valores de píxel se generan según técnicas de decodificación bien conocidas, como los modos de decodificación inter e intra.

El valor de píxel tal como se usa en el presente documento denota cualquier valor de una propiedad de píxel asignada a un píxel. En implementaciones típicas para HEVC, el valor de píxel es un valor de color. Se conocen distintos formatos de color en la técnica y pueden usarse según las realizaciones. Por ejemplo, un valor de color podría comprender componentes de luminancia y crominancia, típicamente un valor de luminancia y dos componentes de crominancia. Por lo tanto, un valor de píxel podría ser un valor de luminancia de un píxel, un valor de crominancia de un píxel o, de hecho, valores de luminancia y crominancia. Otro ejemplo de un formato de color común es el llamado formato RGB, que significa rojo-verde-azul. Un valor de color comprende un valor rojo, verde y azul. Un valor de píxel podría ser un valor RGB, un valor rojo, un valor verde o un valor azul. También se conocen variantes del formato RGB, como RGBA, y se pueden usar según las realizaciones.

De hecho, las realizaciones no necesariamente tienen que limitarse al uso de valores de color como valores de píxel, sino que también pueden aplicarse a otros valores de píxel conocidos, incluidos valores de escala de grises, valores normales, es decir, valores de coordenadas X, Y, Z, etc.

El indicador de corte recuperado en la etapa S20 se emplea luego en la etapa S22 para determinar si el corte actual es el primer corte en la imagen y, por lo tanto, no tiene ninguna dirección de corte asociada o si el corte actual no es el primer corte y, por lo tanto, se requiere una dirección de corte para el corte.

Si la etapa S22 determina que el corte actual es efectivamente el primer corte, por ejemplo, cuando el indicador de corte tiene un valor de uno, el procedimiento continúa a la etapa S23. La etapa S23 simplemente asigna los valores de píxeles generados en la etapa S21 a una primera porción de la imagen que comienza con el inicio de corte definido en la imagen, habitualmente la esquina superior izquierda de la imagen. Los valores de píxel generalmente se asignan a los píxeles en un orden de procesamiento definido, como el orden de exploración de ráster o Morton mencionado anteriormente. En una realización típica aplicada a HEVC, las unidades de codificación más pequeñas que la LCU se procesan en el orden de Morton, mientras que las LCU de la imagen se procesan en el orden de exploración de ráster. Esto implica que la decodificación comienza con la primera LCU del corte y luego, si esta LCU se divide en CU más

pequeñas, estas CU más pequeñas se decodifican en el orden de Morton. Una vez que la LCU ha sido decodificada, el proceso continúa con la siguiente LCU según el orden de exploración de ráster y cualquier CU más pequeña de esta próxima LCU se decodifica en el orden Morton.

5 En una realización particular, el paso S22 se realiza de hecho antes del paso S21 para determinar que el corte actual es, de hecho, el primer corte de la imagen. Luego, los datos codificados de la presentación de imagen codificada que pertenecen al corte actual se decodifican y se asignan a unidades de codificación de píxeles por unidad de codificación. Esto significa que las etapas S21 y S23 forman un bucle que avanza a través de las distintas CU del corte y decodifica cada CU una por una y asigna el valor de píxel a la CU de píxeles por CU según el orden de procesamiento mencionado
10 anteriormente.

Si, en cambio, el paso S22 determina que el corte actual no es el primer corte de la imagen en función del valor del indicador de corte asociado, como tener un valor de cero, el procedimiento continúa en la etapa S24. La etapa S24 recupera la dirección de corte de la presentación de imagen codificada, habitualmente del encabezado del corte. La
15 dirección de corte se emplea para identificar el inicio de corte dentro de la imagen y, por lo tanto, la parte de la imagen que pertenece al corte. La siguiente etapa S25 luego asigna valores de píxel a los píxeles en la porción identificada de la imagen para generar de ese modo un corte descodificado.

De manera similar a las etapas S21 y S23 anteriores, las etapas S22 y S24 se pueden realizar antes de las etapas
20 S21 y S25 para concluir primero que el corte actual no es el primero y luego identificar y leer la dirección del corte de la representación de imagen codificada. Posteriormente, el inicio de corte se identifica en función de la dirección del corte y se puede iniciar la decodificación de los datos codificados para el corte. La decodificación puede continuar CU por CU y luego asignar los valores de píxeles decodificados a los píxeles en la CU actual antes de continuar a la siguiente CU según el orden de procesamiento.

25 En una estrategia alternativa, la etapa S22 se realiza antes de la etapa S21. Por lo tanto, las etapas S22 investigan si el indicador de corte está configurado o no y luego proceden generando la representación decodificada de los valores de píxeles y asignan como se indica en la etapa S23 o primero recuperan y usan la información de dirección en la etapa S24 para identificar qué porción de la imagen para asignar los valores de píxel en la etapa S25.

30 Una vez que todos los datos codificados de un corte han sido decodificados y asignados a la porción de píxel identificada para el corte en la etapa S23 o S25, el procedimiento finaliza o continúa adicionalmente al siguiente corte de la imagen actual a otro corte de otra imagen en una transmisión de vídeo. En tal caso, el procedimiento de la figura 7 se repite para este próximo u otro corte.

35 Sin embargo, en realizaciones preferidas que reducen el tiempo total de decodificación de una imagen, se pueden decodificar múltiples cortes en paralelo. En tal caso, el procedimiento de la figura 7 se realiza para cada uno de estos cortes y, por lo tanto, se toma una decisión de la etapa S22 para cada uno de los cortes en función del indicador de corte respectivo del corte a decodificar.

40 La figura 8 es un diagrama de flujo que ilustra pasos adicionales del procedimiento de la figura 7. El procedimiento continúa desde la etapa S22 de la figura 7, que concluyó que el corte actual no es el primer corte de la imagen según su indicador de corte asociado. Una siguiente etapa S30 recupera información de la granularidad jerárquica para la representación de imagen codificada. Como se discutió anteriormente en el presente documento, la información de
45 granularidad se puede incluir en la representación de imagen codificada y luego se recupera de ella en la etapa S30. Alternativamente, la información de granularidad puede haberse incluido en una representación de imagen codificada recibida previamente relacionada con una misma transmisión de vídeo. En tal caso, la información de granularidad se recupera de la misma y se almacena para su uso posterior al decodificar las siguientes representaciones de imágenes codificadas. La información de granularidad también podría haberse enviado por separado de cualquier representación
50 de imagen codificada, como un campo PPS o SPS separado. Los identificadores de sesión, imagen o transmisión podrían usarse para identificar la información de granularidad relevante para la representación de imagen codificada actual.

El siguiente paso opcional S31 recupera información del número de LCU en la imagen actual. Esta información
55 simplemente podría identificar el número de tales LCU o podría ser utilizada por el decodificador para calcular el número de LCU. Por ejemplo, la palabra de código *NumLCUsInPicture* podría recuperarse de la representación de imagen codificada o de la información del encabezado global, como los campos PPS o SPS. Alternativamente, *NumLCUsInPicture* se calcula en función de la información del tamaño total de la imagen, tal como se recupera de la representación de la imagen codificada o del encabezado global, y la información del tamaño de la LCU, por ejemplo,
60 las palabras de código *log2_min_coding_block_size_minus3* y *log2_diff_max_min_coding_block_size* anteriormente mencionadas.

Una siguiente etapa S32 determina la longitud de la dirección de corte del corte actual en función de la información de la granularidad jerárquica y preferiblemente en función del número de LCU en la imagen. En una realización particular, la longitud de la dirección de corte se define como $\text{ceil}(\log_2(\text{NumLCUsInPicture}) + \text{SliceGranularity})$. Por lo tanto, en una realización particular, la longitud de la dirección de corte se determina en función de la información de la granularidad jerárquica y en función de la información del número de LCU en la imagen actual. El parámetro *SliceGranularity* se obtiene preferiblemente directamente de la información de granularidad *slice_granularity* como $\text{SliceGranularity} = (\text{slice_granularity} \ll 1)$.

El procedimiento continúa luego con la etapa S24 de la figura 7, donde la dirección de corte del corte actual se recupera de la presentación de imagen codificada en función de la información de la longitud de la dirección de corte determinada en la etapa S32. Por lo tanto, esta longitud se emplea para identificar qué bits o símbolos de la representación de imagen codificada, que define la dirección del corte al definir la longitud de la dirección de corte, que preferiblemente tiene un punto de inicio fijo en el encabezado del corte, pero donde el punto final depende de la longitud de la dirección de corte.

En el caso de que las presentes realizaciones se apliquen sobre H.264/MPEG-4 AVC, no hay información de granularidad disponible y, por lo tanto, se puede omitir la etapa S30. La etapa S31 recupera información o permite la determinación del número de macrobloques en la imagen y dónde se emplea esta información en la etapa S32 para determinar la longitud de la dirección de corte.

La figura 9 es un diagrama de flujo que ilustra una realización particular de identificación de la posición de la primera unidad de codificación del corte y, por lo tanto, el inicio del corte dentro de la imagen. El procedimiento continúa desde la etapa S24 de la figura 7. Una siguiente etapa S40 determina una dirección de LCU que representa una posición de una LCU dentro de la imagen en la que están presentes la primera CU y el inicio de corte. La etapa S40 emplea la dirección de corte para determinar la dirección de LCU. Por ejemplo, el parámetro *LCUAddress* se puede determinar como $\text{slice_address} \gg \text{SliceGranularity}$, donde *slice_address* representa la dirección de corte. Una siguiente etapa S41 determina correspondientemente una dirección de sub-LCU que representa una posición de la primera CU dentro de la LCU identificada en la etapa S40. Esta dirección sub-LCU también se determina en función de la dirección de corte. Por ejemplo, el parámetro *GranularidadDirección* se determina como $\text{slice_address} - (\text{LCUAddress} \ll \text{SliceGranularity})$.

Las partes LCU y sub-LCU se pueden emplear para calcular la dirección de corte final como $(\text{LCUAddress} \ll \log_2(\text{diff_max_min_coding_block_size} \ll 1)) + (\text{GranularidadDirección} \ll ((\log_2(\text{diff_max_min_coding_block_size} \ll 1) - \text{SliceGranularity}))$ que se emplea en la etapa S42 para identificar la parte de la imagen que pertenece al corte actual. Por lo tanto, esta parte comienza con el inicio de corte y la primera CU identificada en función de la dirección de corte y luego continúa según el orden de procesamiento a través de la imagen hasta que todos los datos codificados del corte hayan sido decodificados y asignados a las CU de la imagen.

En realizaciones alternativas, la dirección de corte recuperada de la representación de imagen codificada se emplea directamente para identificar el inicio de corte y la primera unidad de codificación. La dirección de corte podría corresponder al número de posiciones de inicio del corte o CU direccionables en las que se inicia el corte. La dirección de corte puede ser un índice en una lista de todas las direcciones posibles en orden de codificación/decodificación. Alternativamente, las coordenadas X e Y se derivan de la dirección de corte y se emplean para ubicar el inicio del corte. Una variante adicional es recuperar o calcular a partir de la dirección de corte las coordenadas LCU y las coordenadas sub-LCU como se describió anteriormente en este documento.

La dirección de corte no solo define la posición de la primera CU del corte y el inicio del corte, sino que también define el tamaño de la primera CU. Por lo tanto, el tamaño de la primera CU se determina en base al menos en parte a la dirección de corte. Más detalladamente, la dirección de corte dicta el tamaño máximo posible en términos de número de píxeles que puede tener la primera CU. Esto significa que la primera CU puede tener un tamaño igual a este tamaño máximo posible o un tamaño más pequeño que el tamaño máximo posible. En el último caso, se utiliza además un indicador de unidad de codificación de corte además de la dirección de corte para definir el tamaño correcto de la primera CU, que se trata adicionalmente en este documento.

Por ejemplo, la primera CU puede asociarse con un indicador de unidad de codificación de corte incluida en la representación de imagen codificada, habitualmente en la porción de datos codificados. El valor del indicador de unidad de codificación de corte se emplea junto con la dirección de corte para definir el tamaño correcto de la primera CU. Por lo tanto, si el indicador de unidad de codificación de corte se establece en un valor definido, preferiblemente uno, el tamaño de la primera CU es más pequeño que el tamaño de la CU máxima posible que puede ocupar la posición dentro de la imagen definida en función de la dirección de corte, ver figura 3B. Sin embargo, si el indicador de la unidad de codificación de corte se establece en otro valor definido, preferiblemente cero, el tamaño de la primera CU es igual al tamaño de la CU máxima posible que puede ocupar la posición en la imagen definida por la dirección

de corte, ver figura 3A.

Es posible que la primera CU esté asociada con múltiples indicadores de unidad de codificación de corte. Por ejemplo, si el tamaño de la CU más grande posible es 32×32 píxeles, mientras que el tamaño de la primera CU es 8×8 píxeles con un tamaño de LCU y SCU de 64×64 píxeles y 8×8 píxeles, el código sería:

split_coding_unit_flag=1 // 32×32 CU se divide en 16×16 CUs

split_coding_unit_flag=1 // la primera 16×16 CU se divide en SCU, no se necesita más indicador de unidad de codificación de corte ya que ahora hemos alcanzado el tamaño objetivo de la primera CU y este también es el tamaño de la SCU, lo que implica que ahora es posible una división adicional.

En algunas realizaciones, el tamaño de la primera CU puede determinarse únicamente en función de la dirección de corte sin utilizar ningún indicador de unidad de codificación de corte como información adicional. Esto es posible cuando el tamaño de la CU máxima posible que puede ocupar la posición dentro de la imagen definida en función de la dirección de corte es igual al tamaño de la SCU. En tal caso, no es posible dividir esta CU máxima posible ya que de hecho es una SCU.

La figura 10 es un diagrama de bloques esquemático de un codificador o dispositivo 100 para codificar una imagen que comprende múltiples cortes. El dispositivo 100 comprende un generador de representación 110 configurado para generar una representación de corte codificada respectiva para cada corte en la imagen en base a los valores de píxel de los píxeles en el corte. El generador de representación 110 realiza esta codificación de píxeles según esquemas de codificación conocidos, tales como codificación inter o intra. Un configurador de indicador 120 del dispositivo 100 está configurado para configurar un indicador de corte asociado con el corte. Si el corte actual es el primer corte de la imagen, el configurador de indicador 120 establece el indicador de sección en un primer valor definido, como uno, mientras que para los cortes restantes en la imagen, el indicador de corte respectivo se establece en un segundo valor definido, como cero.

Un generador de dirección 130 genera una dirección de corte respectiva para cada corte, excepto el primer corte de la imagen, es decir, para cada corte con un indicador de corte establecido en cero. La dirección de corte generada por el generador de direcciones 130 permite la identificación de una posición de una primera CU del corte dentro de la imagen y, por lo tanto, la posición de inicio del corte dentro de la imagen.

El dispositivo 100 también comprende un administrador de representación 140 configurado para incluir las representaciones de corte codificadas respectivas del generador de representación 110, los indicadores de corte del configurador de indicador 120 y las direcciones de corte del generador de dirección de corte 130 en una representación de imagen codificada de la imagen. En una realización particular, el indicador de corte se proporciona en la representación codificada antes de las direcciones de corte. En tal caso, el análisis es posible ya que el indicador de corte decide si hay un campo de dirección de corte o no en la representación de corte codificada.

En una realización, el generador de dirección 130 genera la dirección de corte en función de la granularidad jerárquica determinada para la imagen por el dispositivo 100. En tal caso, se puede implementar un determinante de longitud opcional 150 en el dispositivo 100 para emplear la granularidad jerárquica con el fin de determinar la longitud de la dirección de corte y, por lo tanto, el número de bits que debe contener la dirección de corte. El determinante de longitud 150 adicionalmente también usa preferiblemente información del número total de LCU en la imagen cuando determina la longitud de la dirección de corte, donde este número total de LCU puede calcularse como se desvela anteriormente en este documento. En otra realización, se omite el determinante de longitud 150 y el propio generador de dirección 130 determina la longitud de la dirección de corte.

En el caso de la codificación AVC H.264/MPEG-4, el determinante de longitud 150 determina preferiblemente la longitud de la dirección de corte en función del número de macrobloques en la imagen, que puede calcularse en función de la información del tamaño total de la imagen.

El generador de direcciones 130 luego usa esta información de la longitud cuando genera la dirección de corte. En una realización particular, el generador de direcciones 130 determina las coordenadas de una posición de LCU dentro de la imagen y las coordenadas de una posición de sub-LCU dentro de la LCU como se describe anteriormente en este documento. La dirección de corte podría entonces comprender representaciones de las representaciones de estas posiciones de LCU y sub-LCU o determinarse a partir de ellas.

El dispositivo 100 se emplea ventajosamente para codificar múltiples cortes en paralelo con el fin de reducir el tiempo de codificación total de una imagen y de una transmisión de vídeo.

El dispositivo 100 podría implementarse al menos parcialmente en software. En tal realización, el dispositivo 100 se

implementa como un producto de programa de ordenador almacenado en una memoria y cargado y ejecutado en un ordenador, procesador o microprocesador especialmente adaptado, tal como una unidad central de procesamiento (CPU). El software incluye elementos de código de programa de ordenador o porciones de código de software que efectúan la operación de al menos el generador de representación 110, el configurador de indicador 120, el generador de direcciones 130, el administrador de representación 140 y el determinante de longitud opcional 150. El programa puede almacenarse en su totalidad o en parte, en uno o más medios de almacenamiento de datos o medios legibles por ordenador volátiles adecuados, como RAM, o uno o más medios de almacenamiento de datos o medios legibles por ordenador no volátiles, como discos magnéticos, CD-ROM, discos DVD, discos duros, en memoria ROM o flash. Los medios de almacenamiento de datos pueden ser medios locales de almacenamiento de datos o se proporcionan de forma remota, como en un servidor de datos. De este modo, el software puede cargarse en la memoria operativa de un ordenador o sistema de procesamiento equivalente para que lo ejecute un procesador. El ordenador/procesador no tiene que estar dedicado a ejecutar solo las funciones descritas anteriormente, sino que también puede ejecutar otras tareas de software. Un ejemplo no limitativo de código de programa utilizado para definir el dispositivo 100 incluye un código de datos múltiples de instrucción única (SIMD).

Alternativamente, el dispositivo 100 puede implementarse en hardware. Existen numerosas variantes de elementos de circuitos que se pueden usar y combinar para lograr las funciones de las unidades del dispositivo 100. Dichas variantes están abarcadas por las realizaciones. Ejemplos particulares de implementación de hardware del dispositivo 100 es la implementación en hardware de procesador de señal digital (DSP) y tecnología de circuito integrado, que incluye tanto circuitos electrónicos de propósito general como circuitos específicos de aplicaciones.

La figura 11 es un diagrama de bloques esquemático de una realización de un decodificador o dispositivo 200 para decodificar una representación codificada de una imagen que comprende múltiples cortes. El dispositivo 200 comprende un recuperador de representación 210 configurado para recuperar un indicador de corte asociado con un corte para decodificar desde la representación de imagen codificada, habitualmente desde un encabezado de corte en la representación de imagen codificada. Se proporciona un generador de representación 220 en el dispositivo 200 para generar una representación decodificada de los valores de píxel de los píxeles en el corte basándose en la representación de imagen codificada. El generador de representación 220 genera los valores de píxel de acuerdo con técnicas conocidas, tales como esquemas de decodificación modo intra o inter.

Un recuperador de dirección 230 se vuelve operable si el indicador de corte recuperado para un corte actual por el recuperador de representación 210 tiene un segundo valor definido, tal como cero, que indica que el corte no es el primer corte en la imagen. El recuperador de dirección 230 lee y recupera una dirección de corte asociada con el corte desde la representación de imagen codificada, tal como desde un encabezado de corte en la representación de imagen codificada. Un asignador de valores 240 asigna valores de píxeles obtenidos del generador de representación 220 a medida que el corte se decodifica a píxeles en una parte del corte identificado en función de la dirección de corte recuperada por el recuperador de dirección 230.

Si el indicador de corte recuperado por el recuperador de representación para un corte actual tiene un primer valor definido, como uno, el asignador de valores 240 puede identificar directamente la porción del corte a la que se deben asignar los valores de píxeles del generador de representación 220. Esta es generalmente la primera parte de la imagen en el orden de codificación/decodificación, como la parte superior izquierda. Por lo tanto, en tal caso no se necesita una dirección de corte para identificar esta primera parte de la imagen.

En el caso de una implementación HEVC y si se asigna información de granularidad jerárquica a los datos de imagen codificados, se puede proporcionar un recuperador de información de granularidad opcional 250 en el dispositivo 200 para recuperar información de una granularidad jerárquica aplicable a un corte actual para decodificar. El recuperador de información de granularidad 250 podría recuperar la información de granularidad de la representación de imagen codificada o de un campo de encabezado global, tal como el campo PPS o SPS, asociado con la representación de imagen codificada. La información de granularidad recuperada por el recuperador de información de granularidad 250 es empleada por un determinante de longitud opcional 260 para determinar la longitud de la dirección de corte y así determinar el número de bits que el recuperador de dirección 230 debería leer para recuperar la dirección de corte. Alternativamente, este determinante de longitud 260 puede omitirse y el recuperador de dirección 230 mismo determina la longitud de la dirección basándose en la información de granularidad.

Un recuperador de información de unidad de codificación opcional 270 se implementa ventajosamente en el dispositivo 200 para recuperar información de un número total de LCU en la imagen desde la representación de imagen codificada, tal como desde un campo de encabezado global, campo PPS o SPS. Esta información podrían ser las mencionadas anteriormente *log2_min_coding_block_size_minus3* y *log2_diff_max_min_coding_block_size*, que permiten que el recuperador de información de la unidad de codificación 270 calcule el número de LCU en la información de imagen dada del tamaño total de la imagen, que preferiblemente también está disponible en la representación de imagen codificada o en un campo de encabezado global archivado, campo PPS o SPS.

El determinante de longitud 260 determina ventajosamente la longitud de la dirección de corte basándose en la información de granularidad del recuperador de información de granularidad 250 y el número total de LCU según lo determinado por el recuperador de información de la unidad de codificación 270.

5

En una realización, el recuperador de dirección 230 está configurado para determinar una primera representación o dirección de LCU de las coordenadas de una posición de una LCU dentro de la imagen en función de la dirección de corte si el corte actual no es el primer corte de la imagen. El recuperador de dirección 230 también determina preferiblemente una segunda representación o dirección de sub-LCU de las coordenadas de una posición de una primera unidad de codificación de corte y, por lo tanto, el inicio del corte dentro de la LCU. Las direcciones de la LCU y la sub-LCU son luego empleadas por el recuperador de direcciones 230 para identificar la parte de la imagen que pertenece al corte actual en función de la dirección LCU y la dirección sub-LCU como se describe en la presente descripción.

10

15 Por ejemplo, el recuperador de dirección 230 puede determinar el parámetro $LCUAddress=slice_address \gg SliceGranularity$ en función de la dirección del segmento (*slice_address*) y en función de la información de la granularidad jerárquica (*SliceGranularity*). La dirección de la sub-LCU se determina preferiblemente como $GranularityAddress=slice_address - (LCUAddress \ll SliceGranularity)$ en función de la dirección de corte (*slice_address*), la información de la granularidad jerárquica (*SliceGranularity*) y la dirección de la LCU.

20

El generador de representación 220 determina preferiblemente el tamaño de la primera CU en el corte en términos de número de píxeles basado al menos en parte en la dirección de corte. La dirección de corte define el tamaño de la CU máxima posible que puede ocupar la posición definida por la dirección de corte en la imagen. En una realización, el tamaño de la primera CU está determinado por el generador de representación 220 basado únicamente en la dirección de corte. Esto es posible cuando el tamaño de la primera CU es igual al tamaño de la SCU y no es posible dividir más la CU. En otras realizaciones, el generador de representación 220 usa adicionalmente uno o más indicadores de unidad de codificación de corte incluidos en la representación de imagen codificada junto con la dirección de corte para determinar el tamaño de la primera CU. Si un solo indicador de unidad de codificación de corte es igual a cero o algún otro valor definido, el tamaño de la primera CU es igual al tamaño de la CU máxima que pueda ocupar la posición dentro de la imagen definida por la dirección de corte. Si el indicador de unidad de codificación de corte es, en cambio, igual a uno u otro valor definido, el tamaño de la primera CU es menor que, preferiblemente un cuarto del tamaño de la CU máxima posible que pueda ocupar la posición dentro de la imagen definida por la dirección de corte.

25

30

Por ejemplo, si el tamaño de la CU máxima posible en la coordenada inicial es de 32×32 píxeles (con un tamaño de LCU de 64×64 píxeles y un tamaño de SCU de 8×8 píxeles), los indicadores de unidad de codificación de corte serían: *split_coding_unit_flag*=0 para un tamaño de 32×32 píxeles de la primera CU *split_coding_unit_flag*=1 *split_coding_unit_flag*=0 para un tamaño de 16×16 píxeles de la primera CU *split_coding_unit_flag*=1 *split_coding_unit_flag*=1 para un tamaño de 8×8 píxeles de la primera CU

35

40

El dispositivo 200 podría implementarse al menos parcialmente en software. En tal realización, el dispositivo 200 se implementa como un producto de programa de ordenador almacenado en una memoria y cargado y ejecutado en un ordenador, procesador o microprocesador especialmente adaptado, tal como una unidad central de procesamiento (CPU). El software incluye elementos de código de programa de ordenador o porciones de código de software que efectúan la operación de al menos el recuperador de representación 210, el generador de representación 220, el recuperador de dirección 230, el asignador de valor 240, el recuperador de información de granularidad opcional 250, el determinante de longitud opcional 260 y el recuperador de información de unidad de codificación opcional 270. El programa puede almacenarse en su totalidad o en parte, en uno o más medios de almacenamiento de datos o medios legibles por ordenador volátiles adecuados, como RAM, o uno o más medios de almacenamiento de datos o medios legibles por ordenador no volátiles, como discos magnéticos, CD-ROM, discos DVD, discos duros, en memoria ROM o flash. Los medios de almacenamiento de datos pueden ser medios locales de almacenamiento de datos o se proporcionan de forma remota, como en un servidor de datos. De este modo, el software puede cargarse en la memoria operativa de un ordenador o sistema de procesamiento equivalente para que lo ejecute un procesador. El ordenador/procesador no tiene que estar dedicado a ejecutar solo las funciones descritas anteriormente, sino que también puede ejecutar otras tareas de software. Un ejemplo no limitativo de código de programa utilizado para definir el dispositivo 200 incluye un código de datos múltiples de instrucción única (SIMD).

45

50

55

Alternativamente, el dispositivo 200 puede implementarse en hardware. Existen numerosas variantes de elementos de circuitos que se pueden usar y combinar para lograr las funciones de las unidades del dispositivo 200. Dichas variantes están abarcadas por las realizaciones. Ejemplos particulares de implementación de hardware del dispositivo 200 es la implementación en hardware de procesador de señal digital (DSP) y tecnología de circuito integrado, que incluye tanto circuitos electrónicos de propósito general como circuitos específicos de aplicaciones.

60

La figura 12 es un diagrama de bloques esquemático de un terminal de medios 300 que aloja un dispositivo 200 para decodificar una representación codificada de una imagen. El terminal de medios 300 puede ser cualquier dispositivo que tenga funciones de decodificación de medios que opere en una transmisión de bits codificada, tal como una transmisión de video de fotogramas codificados para decodificar los fotogramas y hacer que los datos de video estén disponibles. Ejemplos no limitativos de tales dispositivos incluyen teléfonos móviles y otros reproductores multimedia portátiles, ordenadores, decodificadores, consolas de juego, etc. El terminal multimedia 300 comprende una memoria 320 configurada para una representación codificada de una imagen, como fotogramas codificados. La representación codificada puede haber sido generada por el propio terminal de medios 300. En tal caso, el terminal de medios 300 comprende preferiblemente un motor o grabador de medios junto con un codificador conectado, tal como el dispositivo para codificar una imagen de la figura 10. Alternativamente, las representaciones codificadas son generadas por algún otro dispositivo y transmitidas de forma inalámbrica o transmitidas por cable al terminal de medios 300. El terminal de medios 300 comprende entonces un transceptor 310 (transmisor y receptor) o un puerto de entrada y salida para lograr la transferencia de datos.

La representación codificada se lleva desde la memoria 320 al dispositivo 200 para decodificar, tal como el dispositivo ilustrado en la figura 11. El dispositivo 200 luego decodifica la representación codificada en una imagen decodificada o como fotogramas decodificados. Los datos decodificados se proporcionan a un reproductor de medios 330 que está configurado para procesar los datos de imagen decodificados o los fotogramas en datos que se puedan visualizar en una pantalla 340 o conectar al terminal de medios 300.

En la figura 12, se ha ilustrado que el terminal de medios 300 comprende tanto el dispositivo 200 para decodificar como el reproductor de medios 330. Sin embargo, esto debería verse simplemente como un ejemplo ilustrativo pero no limitativo de una realización de implementación para el terminal de medios 300. También son posibles implementaciones distribuidas donde el dispositivo 200 y el reproductor de medios 330 están provistos en dos dispositivos físicamente separados y están dentro del alcance del terminal de medios 300 como se usa en la presente invención. La pantalla 340 también podría proporcionarse como un dispositivo separado conectado al terminal de medios 300, donde está teniendo lugar el procesamiento de datos real.

Las realizaciones descritas anteriormente se deben entender como unos pocos ejemplos ilustrativos de la presente invención. Se entenderá por los especialistas en la materia que pueden hacerse diversas modificaciones, combinaciones y cambios en las realizaciones sin apartarse del alcance de la presente invención. En particular, las soluciones de partes diferentes en las diferentes realizaciones se pueden combinar en otras configuraciones, cuando sea técnicamente posible. El alcance de la presente invención se define, sin embargo, por las reivindicaciones adjuntas.

35

REIVINDICACIONES

1. Un procedimiento para codificar una imagen (1) que comprende dos o más cortes (2-5), dicho procedimiento comprende:
- 5 generar (S1), para cada corte (2-5) de dichos cortes (2-5), una representación de corte codificada basada en valores de píxeles de píxeles en dicho corte (2-5);
 establecer (S2) un indicador de corte asociado con un primer corte (2) de dichos dos o más cortes (2-5) en un primer valor definido para indicar que el primer corte (2) es el primer corte de la imagen (1) y de ese modo definir una dirección de inicio de corte asociada en la imagen (1);
 10 ajustar (S3), para cada corte (3-5) de los cortes restantes (3-5) de dichos dos o más cortes (2-5), un indicador de corte asociado con dicho corte (3-5) a un segundo valor definido;
 generar (S4), para cada corte (3-5) de dichos cortes restantes (3-5), una dirección de corte que permite la identificación de una posición de una primera unidad de codificación (10B, 20B, 30B, 40B) de dicho corte (3-5)
 15 dentro de dicha imagen (1); y
 generar (S5) una representación de imagen codificada (50) de dicha imagen (1) que comprende dichas representaciones de cortes codificadas, dichas direcciones de cortes y dichos indicadores de cortes;
- caracterizado porque** dichas direcciones de corte se señalan con un código de longitud fija.
- 20 2. El procedimiento según la reivindicación 1, que comprende además determinar (S11) una longitud de dichas direcciones de corte en términos de número de bits como uno de $\text{ceil}(\log_2(P))$ y $\text{ceil}(\log_2(P-1))$, donde $\text{ceil}()$ denota la función techo definida como $\text{ceil}(x) = \lceil x \rceil$ y genera el número entero más pequeño, no menor que x , y P indica un número total de direcciones de corte posibles en dicha imagen (1).
- 25 3. El procedimiento según la reivindicación 1 o 2, en el que dicha imagen (1) comprende múltiples unidades de codificación más grandes (10), que tienen un tamaño seleccionado en términos de número de píxeles y se pueden dividir jerárquicamente en múltiples unidades de codificación (20, 30, 40) que tienen tamaños respectivos que son más pequeños que dicho tamaño seleccionado, dicho procedimiento comprende además determinar (S11) una longitud de dichas direcciones de corte en términos de número de bits basándose en una granularidad jerárquica seleccionada para dicha imagen (1), dicha granularidad jerárquica define un nivel jerárquico para la alineación del borde de corte dentro de dicha imagen (1), donde dicho nivel jerárquico define un tamaño de la unidad de codificación direccionable mínima posible (30) en la que se puede alinear el inicio de un corte (3) en dicha imagen (1).
- 30 4. El procedimiento según la reivindicación 3, en el que determinar (S11) dicha longitud de dichas direcciones de corte comprende determinar (S11) dicha longitud de dichas direcciones de corte en función de dicha granularidad jerárquica y en función de un número total de unidades máximas de codificación (10) en dicha imagen (1).
- 35 5. Un dispositivo (100) para codificar una imagen (1) que comprende dos o más cortes (2-5), dicho dispositivo (100) comprende:
- un generador de representación (110) configurado para generar, para cada corte (2-5) de dichos cortes (2-5), una representación de corte codificada basada en valores de píxeles de píxeles en dicho corte (2-5);
 45 un configurador de indicador (120) configurado para establecer un indicador de corte asociado con un primer corte (2) de dichos corte (2-5) a un primer valor definido para indicar que el primer corte (2) es el primer corte de la imagen (1) y de ese modo definir una dirección de inicio de corte asociada en la imagen (1) y establecer, para cada corte (3-5) de los cortes restantes (3-5) de dichos cortes (2-5), un indicador de corte asociado con dicho corte (3-5) a un segundo valor definido;
 50 un generador de direcciones (130) configurado para generar, para cada corte (3-5) de dichos cortes restantes (3-5), una dirección de corte que permite la identificación de una posición de una primera unidad de codificación (10B, 20B, 30B, 40B) de dicho corte (3-5) dentro de dicha imagen (1); y
 un administrador de representación (140) configurado para generar una representación de imagen codificada (50) de dicha imagen (1) que comprende dichas representaciones de corte codificadas, dichas direcciones de corte y dichos indicadores de corte; **caracterizado porque** dichas direcciones de corte se señalan con un código de longitud fija.
- 55 6. El dispositivo según la reivindicación 5, que comprende además un determinante de longitud (150) configurado para determinar una longitud de dichas direcciones de corte en términos de número de bits como uno de $\text{ceil}(\log_2(P))$ y $\text{ceil}(\log_2(P-1))$, donde $\text{ceil}()$ denota la función techo definida como $\text{ceil}(x) = \lceil x \rceil$ y genera el entero mínimo no menor que x y P denota un número total de posibles direcciones de corte en dicha imagen (1).
- 60

7. El dispositivo según la reivindicación 5 o 6, en el que dicha imagen (1) comprende múltiples unidades máximas de codificación (10), que tienen un tamaño seleccionado en términos de número de píxeles y pueden dividirse jerárquicamente en múltiples unidades de codificación (20, 30,40) que tienen tamaños respectivos que son más pequeños que dicho tamaño seleccionado, dicho dispositivo (100) comprende además un determinante de longitud 5 (150) configurado para determinar una longitud de dichas direcciones de corte en términos de número de bits en función de una granularidad jerárquica seleccionada para dicha imagen (1), dicha granularidad jerárquica define un nivel jerárquico para la alineación de borde de corte dentro de dicha imagen (1), donde dicho nivel jerárquico define un tamaño de la unidad de codificación direccionable mínima posible (30) en la que se puede alinear un inicio de un corte (3) en dicha imagen (1).
- 10 8. El dispositivo según la reivindicación 7, en el que dicho determinante de longitud (150) está configurado para determinar dicha longitud de dichas direcciones de corte basándose en dicha granularidad jerárquica y en un número total de unidades máximas de codificación (10) en dicha imagen (1).
- 15 9. Un terminal de medios (300) que comprende un dispositivo para codificar una imagen según una cualquiera de las reivindicaciones 5 a 8.

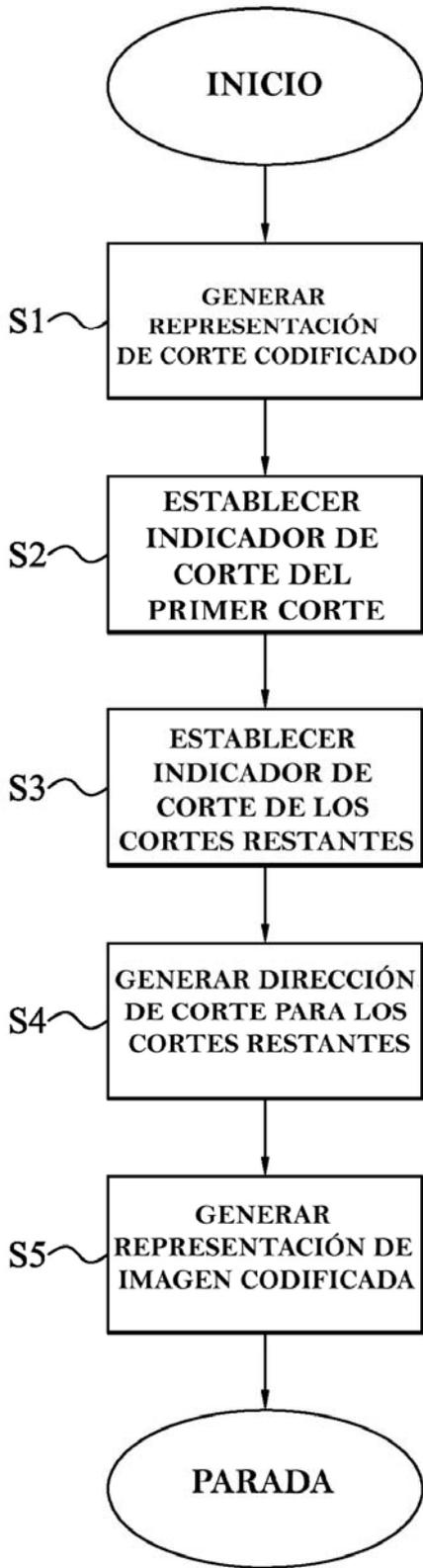


Fig. 1

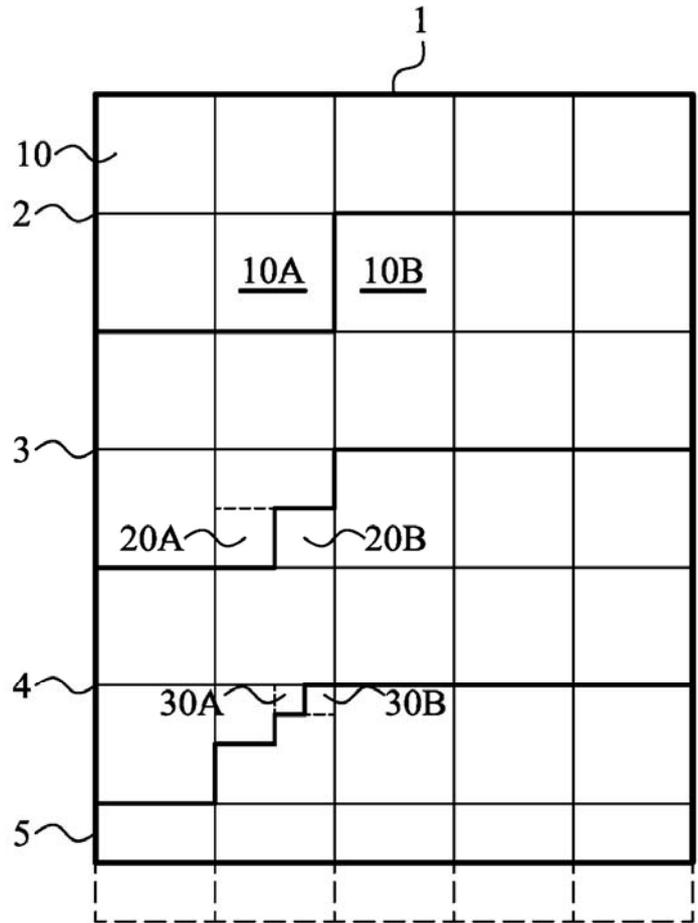


Fig. 2

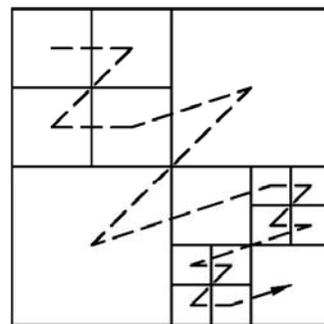


Fig. 4

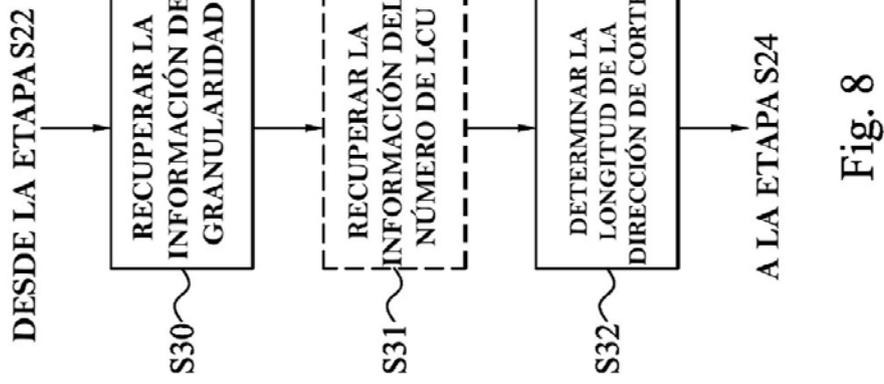
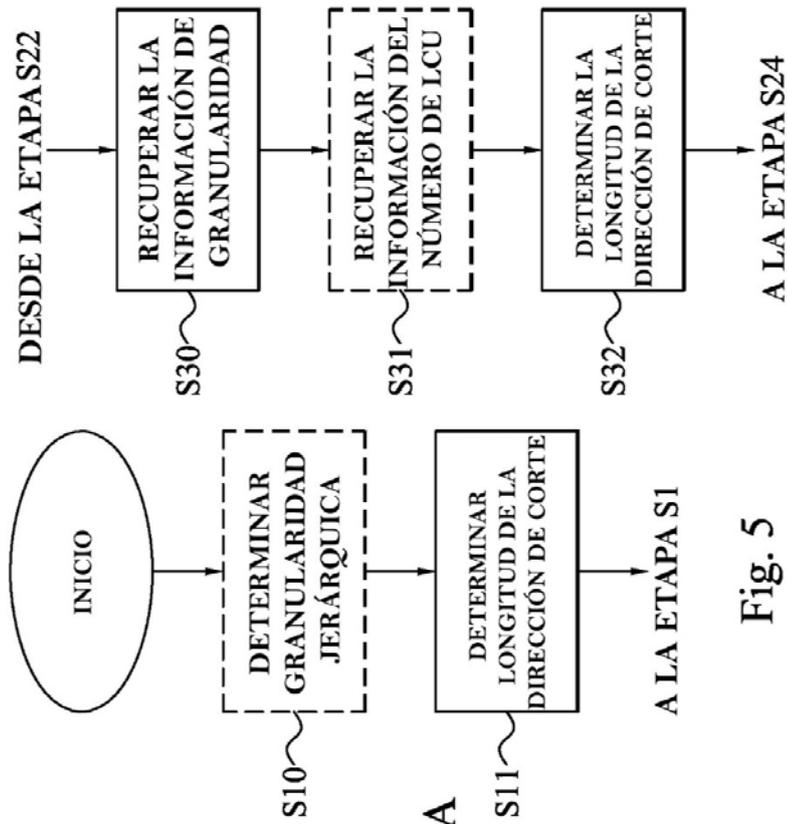
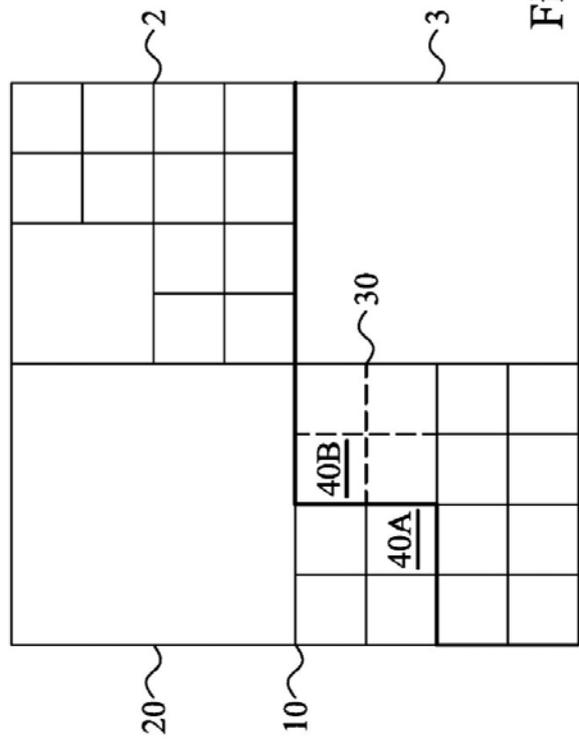
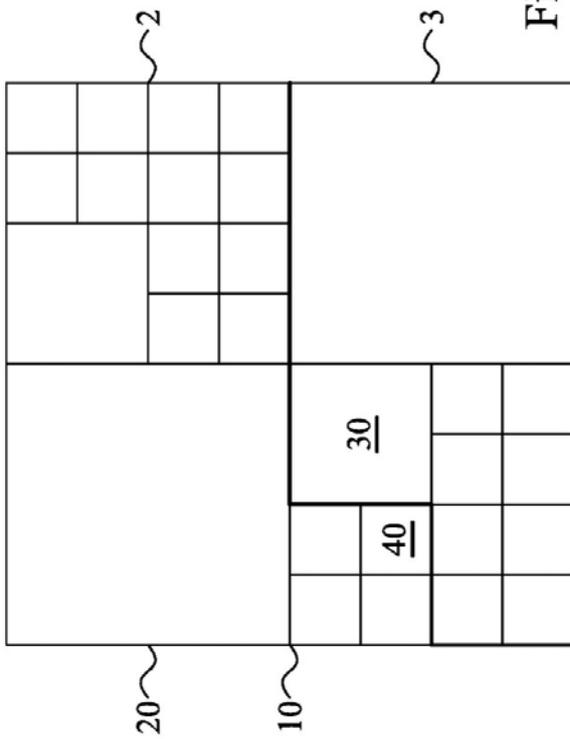




Fig. 6

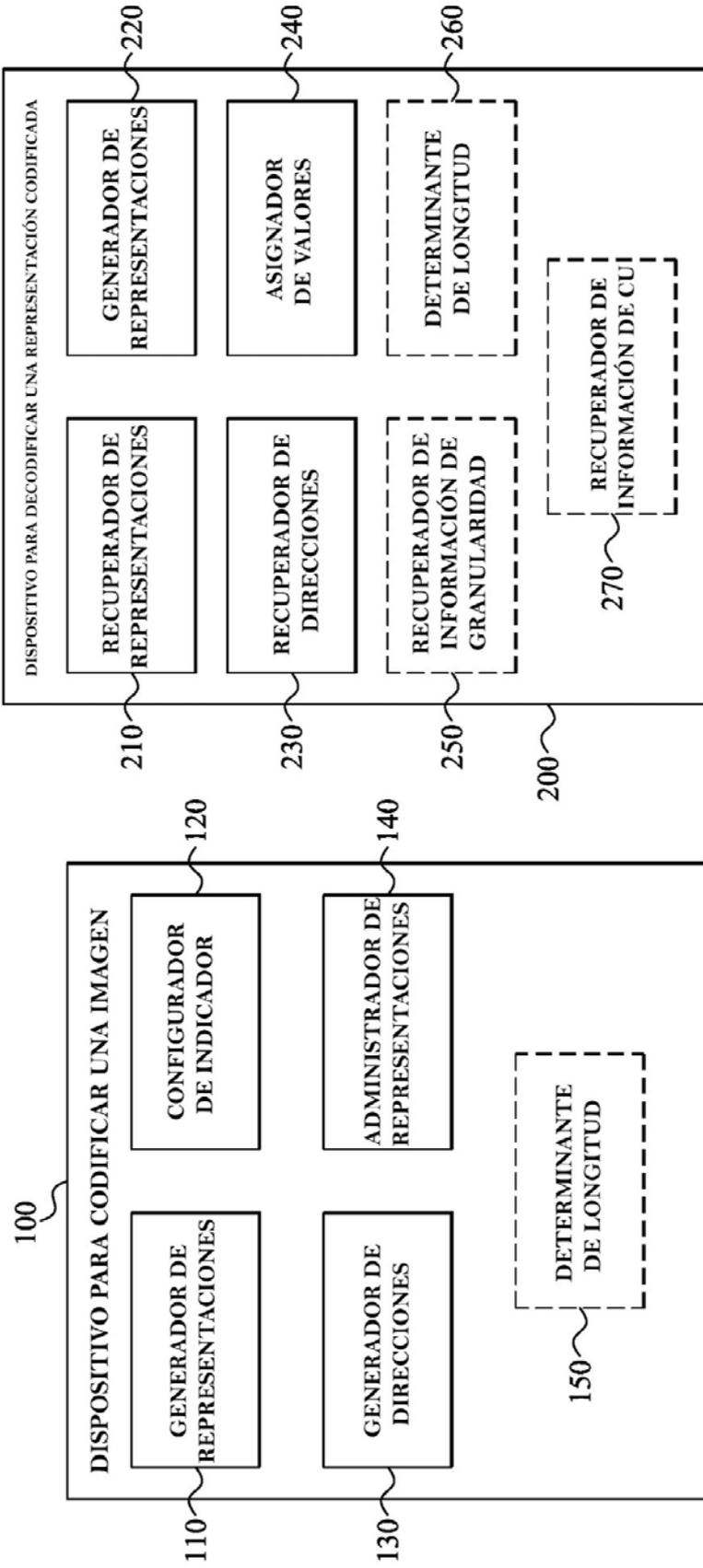


Fig. 11

Fig. 10

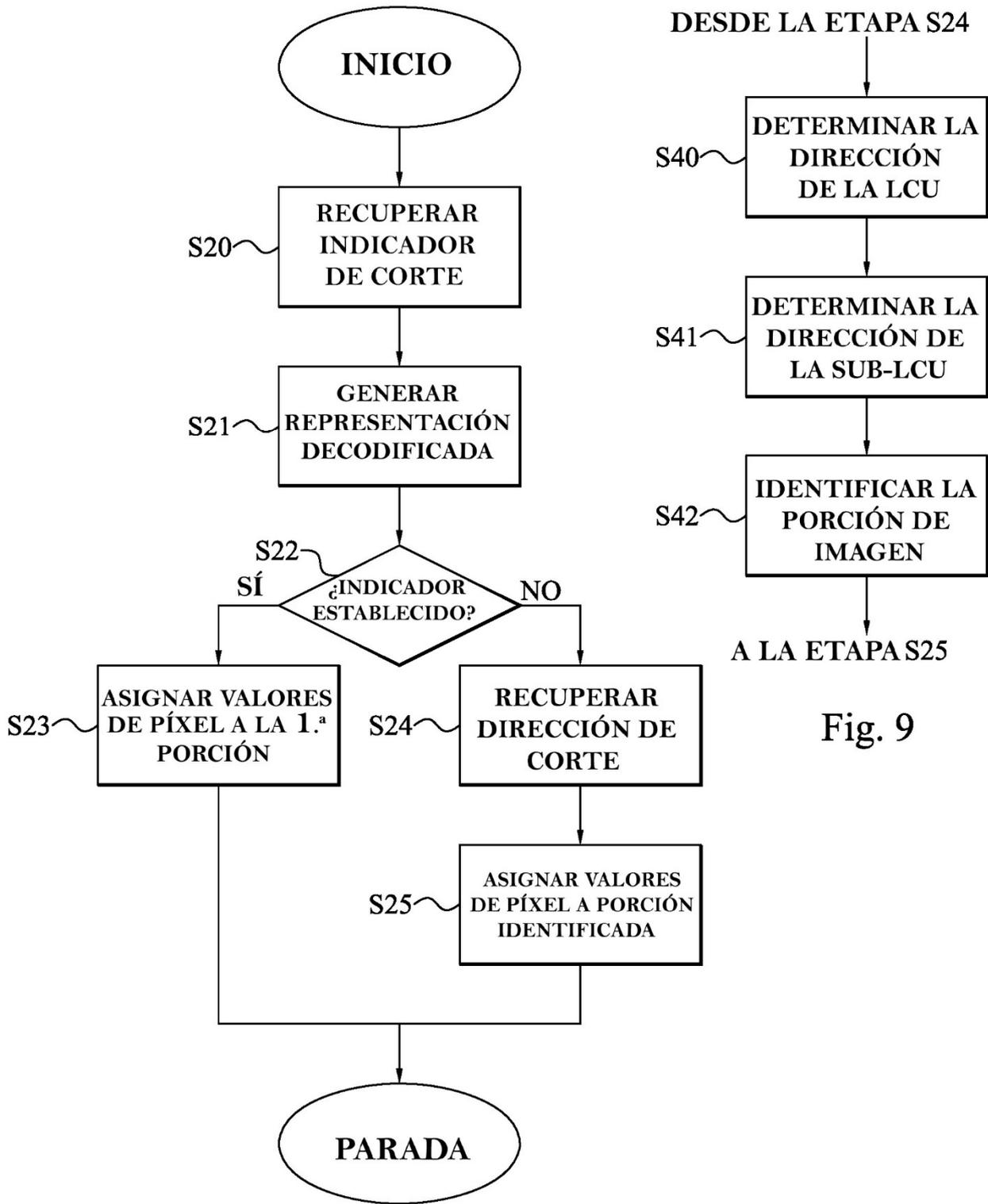


Fig. 7

Fig. 9

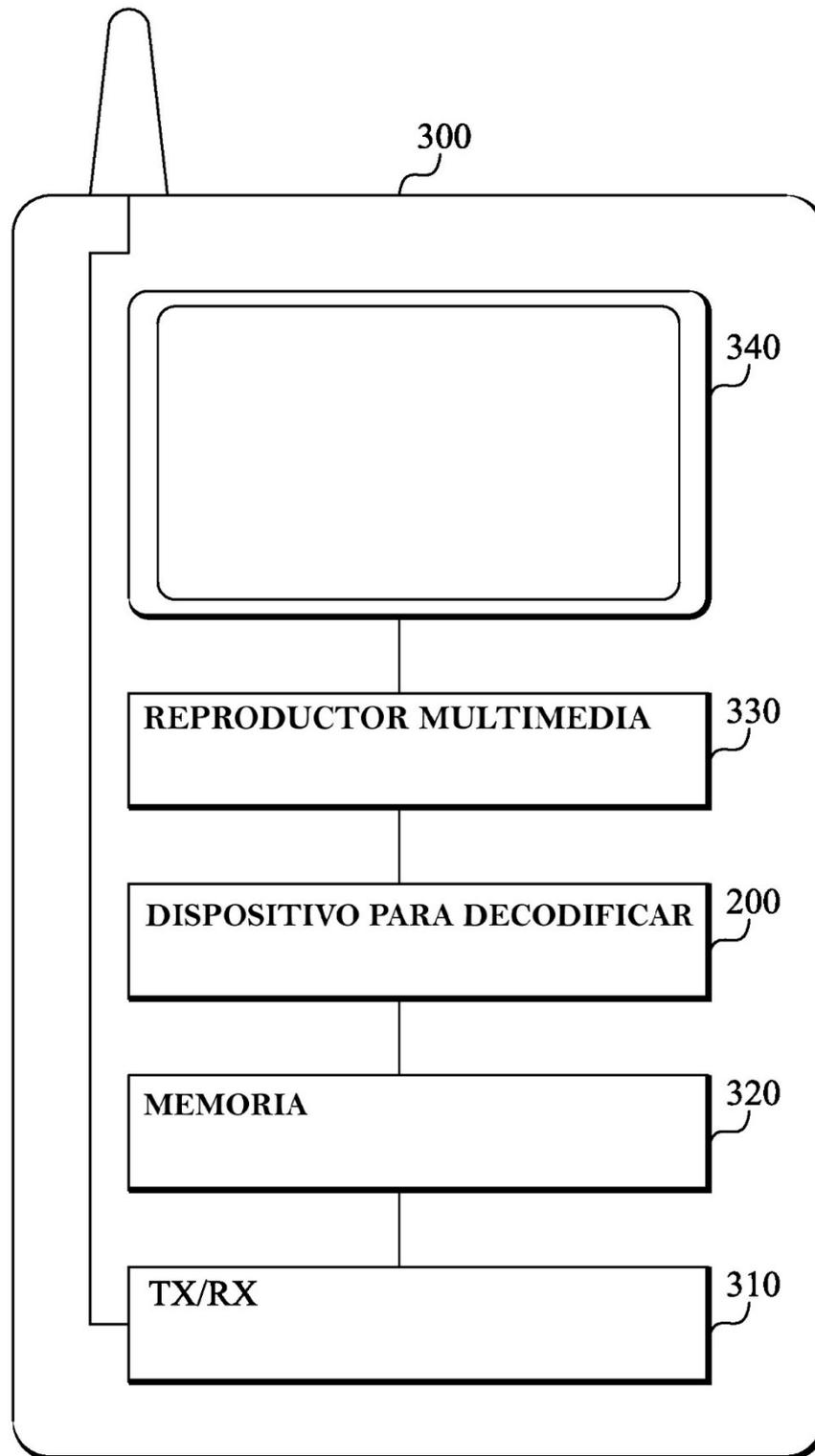


Fig. 12