

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 811 234**

51 Int. Cl.:

<b>H04N 19/70</b>	(2014.01)
<b>H04N 19/105</b>	(2014.01)
<b>H04N 19/46</b>	(2014.01)
<b>H04N 19/61</b>	(2014.01)
<b>H04N 19/117</b>	(2014.01)
<b>H04N 19/174</b>	(2014.01)
<b>H04N 19/82</b>	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **28.12.2011 PCT/CA2011/001411**
- 87 Fecha y número de publicación internacional: **05.07.2012 WO12088594**
- 96 Fecha de presentación y número de la solicitud europea: **28.12.2011 E 11853888 (3)**
- 97 Fecha y número de publicación de la concesión europea: **22.07.2020 EP 2659675**

54 Título: **Método para segmentación de imágenes usando columnas**

30 Prioridad:

**28.12.2010 US 201061427569 P**  
**23.12.2011 US 201113336675**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**11.03.2021**

73 Titular/es:

**DOLBY INTERNATIONAL AB (100.0%)**  
**Apollo Building, 3E, Herikerbergweg 1-35**  
**1101 CN Amsterdam Zuid-Oost, NL**

72 Inventor/es:

**HOROWITZ, MICHAEL**

74 Agente/Representante:

**ELZABURU, S.L.P**

**ES 2 811 234 T3**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

## DESCRIPCIÓN

Método para segmentación de imágenes usando columnas

Esta solicitud reivindica la prioridad de la Solicitud de Patente Provisional de EE. UU., número de serie 61/427,569, presentada el 28 de diciembre de 2010, titulada "PICTURE SEGMENTATION USING GENERALIZED SLICES", y de la Solicitud de Patente de EE. UU., número de serie 13/336,675, presentada el 23 de diciembre de 2011, titulada "METHOD AND SYSTEM FOR PICTURE SEGMENTATION USING COLUMNS".

**Campo**

Las realizaciones de la invención se refieren a la codificación de video, y más específicamente a la segmentación de una imagen codificada en una o más columnas y sectores.

**10 Antecedentes**

Las capacidades de video digital se pueden incorporar en una amplia gama de dispositivos, incluidos televisores digitales, sistemas de transmisión digital directa, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras de video, dispositivos de grabación digital, dispositivos de videojuegos, consolas de videojuegos, teléfonos móviles o de radio por satélite, y similares. Los dispositivos de video digital pueden implementar técnicas de compresión de video, como las descritas en estándares como MPEG-2, MPEG-4, ambos disponibles en la Organización Internacional de Normalización ("ISO") 1, ch. de la Voie-Creuse, Apartado de Correos 56, CH-1211 Ginebra 20, Suiza, o [www.iso.org](http://www.iso.org), o ITU-T H.264/MPEG-4, Parte 10, Codificación de video Avanzada ("AVC"), disponible en la Unión Internacional de Telecomunicaciones ("UIT"), Place de Nations, CH-1211 Ginebra 20, Suiza o [www.itu.int](http://www.itu.int), o de acuerdo con otras especificaciones estándares o no estándares, para codificar y/o decodificar información de video digital de manera eficiente.

Un codificador de video puede recibir información de video sin codificar para procesar en cualquier formato adecuado, que puede ser un formato digital conforme a ITU-R BT 601 (disponible en la Unión Internacional de Telecomunicaciones, Place des Nations, 1211 Ginebra 20, Suiza, [www.itu.int](http://www.itu.int), y que se incorpora en la presente memoria como referencia en su totalidad) o en algún otro formato digital. El video no codificado puede organizarse tanto espacialmente en valores de píxeles dispuestos en una o más matrices bidimensionales, así como temporalmente en una serie de imágenes no codificadas, comprendiendo cada imagen no codificada una o más de las matrices bidimensionales de valores de píxeles mencionadas anteriormente. Además, cada píxel puede comprender una serie de componentes separados utilizados para representar el color en formato digital. Un formato común para video no codificado que se introduce a un codificador de video tiene, para cada grupo de cuatro píxeles, cuatro muestras de luminancia que contienen información relativa al brillo/claridad u oscuridad de los píxeles, y dos muestras de crominancia que contienen información de color (por ejemplo, YCrCb 4:2:0).

Una función de los codificadores de video es traducir (más generalmente "transformar") imágenes no codificadas en un flujo de bits, flujo de paquetes, flujo de unidades NAL u otro formato de transmisión adecuado (todos denominados "flujo de bits" en adelante), con objetivos tales como reducir el cantidad de redundancia codificada en el flujo de bits para aumentar así las tasas de transmisión, aumentando la resistencia del flujo de bits para suprimir errores de bits o borrados de paquetes que pueden ocurrir durante la transmisión (conocidos colectivamente como "resistencia al error") u otros objetivos específicos de la aplicación. Las realizaciones de la presente invención han proporcionado al menos una de entre la eliminación o reducción de la redundancia, el aumento de la resistencia al error y la capacidad de implementación de codificadores de video y/o decodificadores asociados en arquitecturas de procesamiento en paralelo.

Una función de los decodificadores de video es recibir como entrada un video codificado en forma de flujo de bits que puede haber sido producido por un codificador de video que cumple con el mismo estándar de compresión de video. El codificador de video luego traduce (más generalmente "transforma") el flujo de bits codificado recibido en información de video no codificada que puede presentarse, almacenarse o manejarse de otra manera.

Tanto los codificadores de video como los decodificadores de video pueden implementarse utilizando la configuración de hardware y/o software, incluidas combinaciones tanto de hardware como de software. Las implementaciones de uno o ambos pueden incluir el uso de componentes de hardware programables, tales como unidades de procesamiento central CPU, de propósito general, como las que se encuentran en ordenadores personales (PC), procesadores integrados, procesadores de tarjetas gráficas, procesadores de señales digitales (DSP), grupos de compuertas programables de campo (FPGA) u otros. Para implementar al menos partes de la codificación o decodificación de video, pueden necesitarse instrucciones, y esas instrucciones pueden almacenarse y distribuirse utilizando uno o más medios legibles por ordenador no transitorios. Las opciones de medios legibles por ordenador incluyen memoria de solo lectura de disco compacto (CD-ROM), memoria de solo lectura de video digital (DVD-ROM), tarjeta de memoria, ROM integrada u otras.

A continuación, se describirán ciertos sistemas, métodos y/o aspectos relacionados con al menos un aspecto amplio de la compresión y descompresión de video, es decir, las operaciones realizadas en un codificador y/o decodificador de video. Un decodificador de video puede realizar todas, o un subconjunto de, las operaciones inversas de las operaciones de codificación. A menos que se indique lo contrario, las técnicas de codificación de video descritas en la presente memoria también pretenden abarcar la inversa de las técnicas de codificación de video descritas (a saber, la

decodificación de video asociada).

La representación digital sin comprimir de video se puede ver como un flujo de muestras, en donde las muestras se pueden procesar mediante el dispositivo de visualización de video en orden de escaneo. Un tipo de límite que ocurre a menudo en este flujo de muestras es el límite entre imágenes en el flujo de muestras. Muchos estándares de compresión de video reconocen este límite y, a menudo, dividen el flujo de bits codificado en estos límites, por ejemplo, mediante la inserción de un encabezado de imagen u otros metadatos al comienzo de cada imagen no codificada.

Para algunas aplicaciones, puede ser ventajoso segmentar la imagen codificada en bloques de datos más pequeños, cuya segmentación puede ocurrir antes, o durante, la codificación. A continuación, se describen dos casos de uso para los que la segmentación de imágenes puede ser ventajosa.

El primer caso de dicho uso implica el procesamiento paralelo. Anteriormente, el video de definición estándar (por ejemplo, 720x480 o 720x576 píxeles) era el formato más grande en uso comercial generalizado. Más recientemente, los formatos de HD (hasta 1920x1080 píxeles), así como 4k (4096x2048 píxeles), 8k (8192x4096 píxeles) y formatos aún más grandes están surgiendo y encontrando uso en una variedad de espacios de aplicación. A pesar del aumento de la potencia informática asequible a lo largo de los años, como resultado de los tamaños de imagen muy grandes asociados con algunos de estos formatos más nuevos y mayores, a menudo es ventajoso aprovechar la eficiencia del procesamiento paralelo en los procesos de codificación y decodificación. La codificación y decodificación paralelas pueden ocurrir en el nivel de instrucción (por ejemplo, usando SIMD), en una tubería donde varias unidades de codificación de video pueden procesarse en diferentes etapas simultáneamente, o en una base de estructura grande donde las colecciones de subunidades de codificación de video se procesan mediante motores de cálculo separado como entidades separadas (p. ej., un procesador multinúcleo de propósito general). La última forma de procesamiento paralelo requiere segmentación de imagen.

El segundo caso de dicho uso implica la segmentación de imágenes para crear un flujo de bits adecuado para el transporte eficiente a través de redes de paquetes. Los códecs cuyo video codificado se transporta a través de IP u otros protocolos de red de paquetes pueden estar sujetos a una restricción de tamaño de unidad de transmisión máxima ("MTU"). A veces es ventajoso que el tamaño de sector codificado sea tal que el paquete resultante que contiene el sector codificado sea lo más cercano posible al tamaño de MTU sin exceder ese tamaño, a fin de mantener alta la relación de carga útil/sobrecarga de paquetes, evitando la fragmentación (y la mayor probabilidad de pérdida resultante) por la red.

El tamaño de MTU difiere ampliamente de una red a otra. Por ejemplo, el tamaño MTU de muchas conexiones a Internet puede establecerse por el tamaño MTU más pequeño de la infraestructura de red comúnmente utilizada en Internet, que a menudo corresponde a limitaciones en Ethernet y puede ser de aproximadamente 1500 bytes.

El número de bits en una imagen codificada depende de muchos factores, tales como las dimensiones de la imagen de origen, la calidad deseada, la complejidad del contenido en términos de idoneidad para la predicción, la eficiencia de codificación del estándar de codificación de video y otros factores. Sin embargo, incluso con configuraciones de calidad y complejidad de contenido moderadas, para secuencias de resolución de HD y superiores, el tamaño de una imagen codificada promedio excede fácilmente el tamaño de MTU. Por ejemplo, un codificador de videoconferencia puede requerir aproximadamente 2 Mbits/s para codificar una secuencia de video 720p60. Esto da como resultado un tamaño de imagen codificado promedio de aproximadamente 33000 bits o 4125 bytes, que es considerablemente más que los aproximadamente 1500 bytes del tamaño MTU de Internet. A resoluciones más altas, el tamaño promedio de la imagen aumenta a valores significativamente superiores al tamaño MTU de Internet. Suponiendo una relación de compresión similar a la del ejemplo anterior de 720p60, un video de 4096 x 2048 (4k) a 60 fps (4kp60) puede requerir más de 300.000 bits, o 25 paquetes de tamaño MTU para cada imagen de video codificada.

En muchos estándares de codificación de video, un segmento de imagen (o, al menos, una forma de segmento de imagen) se conoce como un "sector". En la siguiente descripción, cualquier tipo de fragmentación de imagen codificada (por ejemplo, basada en el estándar de codificación de video) que rompa cualquier forma de predicción en imagen u otro mecanismo de codificación se puede denominar generalmente como un "sector". Como tal, estructuras como el Grupo de Bloques ("GOB") en la Rec. H.261 de ITU.T o en la Rec. H.263 de ITU (disponible en la ITU; véase antes para H.264), sectores en H.264 o la familia de estándares MPEG, cada uno puede constituir un "sector", ya que este término se usa aquí en todo momento. Sin embargo, las unidades de fragmentación de RFC3984 o las particiones de datos de H.264 pueden no constituir un "sector", ya que este término se usa aquí en todo momento, incluso si subdividen el flujo de bits de una imagen codificada en bloques de datos más pequeños, porque no se rompen en la predicción de imagen u otro mecanismo de codificación.

Una ventaja de usar sectores sobre mecanismos de segmentación inconscientes de medios, como, por ejemplo, los proporcionados por IP en la capa de enrutamiento, es que los sectores son al menos en cierta medida decodificables de forma independiente (como se trata a continuación con más detalle). Por lo tanto, la pérdida de un sector no necesariamente hace que los otros sectores de una imagen codificada sean inutilizables o no decodificables. Dependiendo de la implementación de un mecanismo de fragmentación, la pérdida de un fragmento, por el contrario, puede inutilizar muchos otros fragmentos.

Muchos o todos los mecanismos de predicción o mecanismos de codificación incorporados en imagen pueden romperse por la decodificación de un encabezado de imagen (o equivalente). Si esos mecanismos de predicción se rompen también por la detección de un encabezado de sector puede depender del estándar de compresión de video y del tipo de sector en uso.

5 En H.264, las imágenes de video individuales se pueden segmentar en uno o más sectores, de modo que se adaptan a las aplicaciones que requieren o utilizan de otro modo imágenes que se particionan como parte del proceso de codificación/decodificación. Los sectores en H.264 pueden ser decodificables independientemente con respecto a la predicción del vector de movimiento, la predicción intra, los estados CA-VLC y CABAC, y otros aspectos del estándar H.264. Aunque esta independencia de decodificación puede generar aumentos en la resistencia al error, no permitir la predicción antes mencionada a través de los límites de sector puede tender a reducir la eficiencia de codificación.

10 En H.263, un codificador de video tiene más flexibilidad para seleccionar qué mecanismos de predicción se rompen mediante el uso de sectores o GOB con encabezados GOB no vacíos. Por ejemplo, hay un bit incluido en el encabezado de la imagen, seleccionable cuando se usa el Anexo R, que le indica al decodificador que no se produce ninguna predicción a través de los límites de sector/GOB. Sin embargo, si el bit no se establece, los vectores de movimiento pueden apuntar fuera del sector actual, lo que potencialmente "importa" valores de muestra que se utilizan para la compensación de movimiento dentro del sector actual. Además, el filtrado en bucle puede incorporar valores de muestra fuera del sector.

15 En la mayoría o en todos los estándares de codificación de video existentes, con la posible excepción del orden flexible de macrobloques ("FMO") utilizado como parte de H.264, los macrobloques dentro de los sectores se ordenan en orden de exploración de ráster. En consecuencia, cuando las secuencias de video con tamaños de imagen grandes se dividen en sectores que abarcan solo un porcentaje relativamente pequeño de todos los macrobloques en la imagen, los sectores tienden a alargarse cuando se visualizan espacialmente.

20 La FIG. 1 muestra una imagen 100 ejemplar que se rompe en sectores de acuerdo con la técnica anterior. La imagen 100 ejemplar tiene una matriz 101 de macrobloques de 6 x 4, cuyos límites se indican a través de líneas finas. La imagen 100 se divide en dos sectores 102, 103, con el límite 104 de sector entre los dos sectores 102, 103 indicado por una línea en negrita. El primer sector 102 contiene 10 macrobloques en orden de escaneo, específicamente, macrobloque 1 a 10. El segundo sector 103 contiene los 14 macrobloques restantes en la matriz 101 (es decir, macrobloques 11 a 24). Los números en los macrobloques (por ejemplo, el número '11' en el macrobloque 105) son las direcciones de los macrobloques según el orden de escaneo.

25 El flujo de bits 106 representa la imagen codificada correspondiente a la imagen 100, y puede incluir uno o más conjuntos 107 de parámetros como ejemplo de una estructura de sintaxis de alto nivel, que puede incluir elementos de sintaxis relevantes para más de uno de los sectores codificados de la imagen 100. El conjunto o conjuntos 107 de parámetros pueden ir seguidos de uno o más sectores, comprendiendo cada uno de estos sectores un encabezado 108, 110 de sector correspondiente y los datos 109, 111 de sector correspondiente, respectivamente. Por consiguiente, en este ejemplo, el encabezado 108 del sector puede estar asociado con los datos 109 del sector y puede corresponder al sector 102 en la matriz 101, mientras que el encabezado 110 del sector puede estar asociado con los datos 111 del sector y puede corresponder al sector 103. Los encabezados 108, 110 de sector pueden incluir información tal como la dirección del primer macrobloque de ese sector respectivo, de acuerdo con el orden de escaneo. Por ejemplo, el segundo sector 103 cuando se codifica en el flujo de bits 106 comienza con el encabezado 110 de sector que incluye una primera dirección de macrobloque de '11', que designa la dirección del macrobloque 105.

30 Como se puede ver en la FIG. 1, los sectores 102 y 103 son algo alargados en el sentido de que cada uno de los sectores 102 y 103 abarcan más macrobloques horizontalmente (es decir, 6 macrobloques) que verticalmente (es decir, 2 a 3 macrobloques). Los sectores alargados, tales como los sectores 102 y 103, tienden a contener diversos contenidos de imagen como resultado de la gran distancia de extremo a extremo horizontalmente. Además, los sectores alargados tienden a tener proporciones bajas de área de sector a perímetro/límite de sector. La combinación de sectores que contienen contenido de imágenes diversas con proporciones de área a perímetro/límite relativamente bajas puede ser desventajosa desde una perspectiva de eficiencia de codificación cuando se compara con un sector que abarca un área más cuadrada de una imagen, tal como cuadrados u otras figuras geométricas cercanas a un cuadrado. Los sectores con esta propiedad geométrica pueden denominarse en adelante sectores "compactos" dentro de esta descripción.

35 También, muchas herramientas de codificación de entropía que tienen propiedades bidimensionales, tales como la codificación de vectores de movimiento o modos de predicción intra, pueden optimizarse para relaciones de aspecto de imagen cuadrada. Por ejemplo, en H.264, la codificación de un vector de movimiento horizontal de una longitud dada cuesta aproximadamente el mismo número de bits que la codificación de un vector de movimiento vertical de la misma longitud. En consecuencia, estas herramientas de codificación pueden producir una mejor compresión para sectores compactos que para sectores "alargados", tales como los sectores 102 y 103 mostrados en la FIG. 1.

40 La eficiencia mejorada de codificación para sectores compactos puede surgir aún más del hecho de que el contenido homogéneo, que es más probable que se encuentre en un sector compacto, puede codificarse de manera más eficiente en comparación con el contenido relativamente diverso que es más probable que se encuentre en un sector alargado. Como regla general, aunque no necesariamente absoluta, es más probable que el contenido de la imagen sea

homogéneo en un sector compacto porque la distancia espacial desde el centro hasta los límites del sector es menor, en promedio, para un sector compacto que para un sector alargado. Además, tener una relación de límite de área de sector a sector más alta para sectores compactos significa que generalmente se pueden romper menos mecanismos de predicción en una imagen dada, dando como resultado una mayor eficiencia de codificación.

5 En H.264, FMO permite que el codificador de video produzca efectivamente sectores rectangulares definiendo grupos de sectores rectangulares. FMO es una herramienta de codificación muy generalizada que fue diseñada para abordar varios problemas encontrados en la codificación de video. Sin embargo, desde un punto de vista práctico, FMO tiende a percibirse como que tiene un grado relativamente alto de complejidad de implementación, lo que da como resultado una adopción algo limitada como un aspecto de la compresión de video estándar. Una herramienta de codificación más simple que puede lograr una eficiencia de codificación mejorada, así como la codificación y decodificación paralelas, puede abordar o mejorar uno o más de los problemas de complejidad asociados con una implementación completa de FMO.

10 El problema de los sectores alargados también puede aparecer en un caso extremo en muchos esquemas de codificación basados en MPEG-2. Por ejemplo, en la codificación MPEG-2, a menudo sucede que cada fila individual de macrobloques dentro de una imagen se codifica en un sector, rompiendo efectivamente cualquier mecanismo de predicción de imagen en la dimensión vertical dentro de la imagen.

15 El modo de sector rectangular es uno de los dos modos secundarios especificados en el Anexo K de H.263, siendo otro el "modo de sector de orden de escaneo", que tiene propiedades similares a los sectores de H.264 tratados anteriormente. Los sectores rectangulares según lo previsto en H.263 pueden ofrecer una o más de las ventajas descritas anteriormente que proporcionan sectores compactos. Sin embargo, H.263 requiere que las dimensiones (específicamente el ancho) de cada sector deben ser transportadas en su encabezado correspondiente, lo que conduce a la ineficiencia de codificación, por ejemplo, en aplicaciones en las que los tamaños de sector en la dimensión horizontal no cambian de imagen a imagen. Además, el Anexo K de H.263 no especifica un ancho de sector mínimo que efectivamente evitaría el uso de sectores alargados verticalmente. Los sectores alargados verticalmente pueden introducir dificultades de implementación y en cualquier caso no proporcionarían las ventajas de eficiencia de codificación deseadas que, por las razones tratadas anteriormente para los sectores alargados horizontalmente, pueden proporcionarse mediante el uso de sectores más compactos.

20 Restringir el sector para que tenga una forma rectangular también puede ser desventajoso en ciertos casos. En primer lugar, los sectores rectangulares pueden funcionar de manera subóptima en aplicaciones para las cuales los flujos de bits usan protocolos de transporte sujetos a una MTU. Por ejemplo, los paquetes pueden fragmentarse si el número de bits dentro de un paquete dado excede el límite de MTU impuesto en el flujo de bits, lo que puede ser indeseable desde al menos el rendimiento de la red y las perspectivas de resistencia al error. Por el contrario, si el número de bits dentro de un paquete dado está muy por debajo del límite de MTU, entonces la relación del número de bits en los encabezados de transporte y sector resulta relativamente grande en comparación con el número de bits en la carga útil del paquete, lo que conduce a ineficiencias de codificación. Requerir que los sectores sean de forma rectangular limita la capacidad del codificador para controlar con precisión el número de bits en el sector codificado para evitar las desventajas mencionadas anteriormente.

25 En segundo lugar, los sectores rectangulares pueden funcionar de manera inferior a la óptima en aplicaciones que utilizan codificación y/o decodificación paralela. Cuando se codifica y/o decodifica en paralelo, generalmente es ventajoso dividir una imagen en diferentes partes de modo que cada parte de la imagen requiera aproximadamente la misma cantidad de potencia computacional para codificar. Al dividir la imagen de esta manera, cada parte de la imagen puede por ello codificarse con casi la misma latencia para reducir o minimizar el retraso entre los tiempos de codificación de diferentes partes de la imagen. Un codificador restringido a utilizar sectores rectangulares puede no ser capaz de controlar con precisión la cantidad de capacidad de CPU requerida para codificar y/o decodificar cada sector y, por lo tanto, evitar esta desventaja potencial.

30 Para facilitar la decodificación paralela de sectores que pertenecen a la misma imagen codificada, un decodificador generalmente asignará sectores de imagen codificados a los diversos procesadores, núcleos de procesador, u otros mecanismos de decodificación que operan independientemente puestos a disposición del decodificador para decodificación paralela. Sin el uso de FMO, esta fue una tarea generalmente difícil, en algunos casos extremadamente difícil, para los estándares de codificación de video anteriores a manejar, ya que esos estándares anteriores permitirían demasiada flexibilidad en la generación de flujo de bits. Por ejemplo, en H.264, es posible que una imagen se codifique en un solo sector y otra imagen en docenas de sectores dentro del mismo flujo de bits. Si la paralelización se produce a nivel de sector, cuando una imagen se codifica en un único sector, el procesador asignado para decodificar esa imagen deberá aprovisionarse para manejar su decodificación en su totalidad. Como resultado, sin imponer restricciones fuera del estándar de codificación de video, puede haber una ventaja comparativamente pequeña al construir decodificadores paralelos si cada procesador de decodificación necesitará ser provisionado para poder manejar una imagen completa, en cualquier caso.

35 La codificación de sector utilizada en muchos codificadores MPEG-2 se considera ampliamente como el resultado de un acuerdo para utilizar una especificación informal de Cable Labs que sugirió un esquema de segmentación de un sector por fila de macrobloques. La aceptación generalizada de esta especificación informal fue finalmente obtenida. Si bien

puede haber sido valioso en dicho esquema de segmentación cuando los primeros productos MPEG-2 estuvieron disponibles, alrededor de 1995, hoy en día, las diversas restricciones asociadas con la especificación histórica pueden limitar significativamente la eficiencia de codificación, aunque la paralelización de la decodificación de (al menos con codificación SD) las imágenes no han sido un problema relativo durante al menos una década.

- 5 Por lo tanto, existe la necesidad de un método y un sistema mejorados para la segmentación de imágenes que aborde, mejore o proporcione una alternativa útil a los inconvenientes existentes de los codificadores de video, tanto en términos de coincidencia de tamaño MTU como de decodificación paralela. Por consiguiente, se desea una solución que aborde, al menos en parte, los anteriores y otros inconvenientes.

10 El documento US 20060115001 describe un método para indicar tamaño, forma y ubicación de una región dentro de una imagen digital que es dividida en un conjunto de bloques. Un valor para al menos un parámetro de tamaño, que es indicativo de un número de los bloques dentro de dicha región es definido, y un valor para al menos un parámetro de evolución de forma, que es indicativo de un orden de selección de los bloques en dicha región es seleccionado. Luego preferiblemente los valores para dicho al menos un parámetro de tamaño y dicho al menos un parámetro de evolución de forma, son codificados en un flujo de bits de una secuencia de video para indicar tamaño, forma y ubicación de la región dentro de la imagen.

15 En "Flexible macroblock ordering as a content adaptation tool in H.264/AVC" ("Ordenación flexible de macrobloques como herramienta de adaptación de contenido"), Proc. SPIE, Vol. 6015 (2005), Dhondt et al., presenta el uso de ordenación flexible de macrobloques, utilizando su idea de combinar juntos macrobloques que no son contiguos en un sector. En lugar de crear un patrón de dispersión, que es útil cuando se transmiten los datos sobre una red propensa a los errores, dividen la imagen en varias regiones de interés y una región restante de desinterés. El documento ZHOU (TI) M: "Sub-picture based raster scanning coding order for HEVC UHD video coding" ("Orden de codificación de escaneo de ráster basado en imagen secundaria para codificación de video HEVC UHD"), 2. JCT-VC MEETING: 21-7-2010 - 28-7-2010; GINEBRA: (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16): URL:HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE, nº JCTVC-B062,-25 Julio 2010 (2010-07-25), describe la codificación de video basada en columnas.

#### Resumen

De acuerdo con aspectos de esta invención, se ha proporcionado un método para codificar una imagen según la reivindicación 1 y un método para decodificar según la reivindicación 8.

30 Aspectos de la invención se refieren en uno o más aspectos generales a técnicas para segmentar una imagen codificada en columnas y sectores.

En algunos ejemplos, se puede definir una pluralidad de columnas en uno o más elementos de sintaxis colocados en una o más estructuras de sintaxis de alto nivel, tales como encabezados de sector o conjuntos de parámetros. Las columnas así definidas pueden tener cualquier ancho entre, por ejemplo, un Bloque de Árbol Codificado (CTB, como se define, por ejemplo, en WD4: (B. Bross et al., "WD4: Working Draft 4 of High-Efficiency Video Coding", disponible en [http://wftp3.itu.int/av-arch/jctvc-site/2011\\_07\\_F\\_Torino/](http://wftp3.itu.int/av-arch/jctvc-site/2011_07_F_Torino/)) y el tamaño horizontal completo de la imagen, cuando se mide en CTB.

En algunos ejemplos, un límite de columna entre columnas adyacentes puede romper uno o más mecanismos de predicción o mecanismos de filtrado de bucle del códec de video. Los mecanismos de predicción o de filtro de bucle rotos por límites de columna pueden ser iguales o diferentes de los rotos por límites de sector.

40 En algunos ejemplos, la granularidad para calcular y especificar el ancho de columna puede ser una Unidad de Codificación Más grande (LCU), y un bloque de árbol codificado (CTB) puede ser del mismo tamaño que una LCU.

En algunos ejemplos, N las columnas pueden ser definidas por un elemento N de sintaxis que indica el número de columnas en una imagen, y que puede ir seguido por N-1 elementos de sintaxis que indican los anchos correspondientes de cada una de las N columnas. El ancho de una columna designada, por ejemplo, la columna situada más a la derecha, se puede calcular, por ejemplo, restando la suma de los anchos de las N-1 columnas más a la izquierda del ancho de la imagen.

En algunos ejemplos, cada columna en la imagen puede tener un ancho igual (excepto quizás para una columna predefinida, tal como la columna más a la derecha, que puede ser de menor ancho) y un elemento W de sintaxis puede usarse para indicar dicho ancho igual de cada columna (excepto la columna predefinida). El elemento W de sintaxis puede colocarse, por ejemplo, en una estructura de elemento de sintaxis de alto nivel tal como un conjunto de parámetros. En tales casos, el ancho de una columna predefinida se puede determinar en función de una operación de división que implica W y el ancho de la imagen, representada por N, por lo que el ancho de la columna predefinida se puede tomar como el resto cuando N se divide por W.

55 En algunos ejemplos, cada columna en la imagen puede tener un ancho igual (excepto quizás para una columna predefinida, tal como la columna más a la derecha, que puede ser de menor ancho, en caso de que el ancho de una imagen en CTB no sea un entero divisible de N), y un elemento N de sintaxis puede usarse para indicar el número de

columnas de igual ancho (excluyendo quizás a la columna predefinida). En tales casos, el ancho de todas las N columnas puede determinarse dividiendo el ancho de la imagen en CTB por N, igualando entonces el ancho de la columna predefinida el resto de la operación de división.

5 En algunos ejemplos, un elemento N de sintaxis puede usarse para indicar el número de columnas en una imagen. El ancho W igual de columna, en unidades de LCU, es igual al ancho de la imagen, en unidades de LCU, dividido por N.

En algunos ejemplos, si el ancho de la imagen no es un número entero divisible por el ancho W igual de columna, entonces una columna predefinida, por ejemplo, la columna situada más a la derecha, puede tener un ancho que es el resto de una operación de división entre el ancho de la imagen (medido, por ejemplo, en CTB) y W-1.

10 En algunos ejemplos, un sector puede abarcar uno o más CTB ubicados cada uno dentro de una sola columna, lo que permite la coincidencia de tamaño de MTU, mientras que aún permite la paralelización a través de columnas.

En algunos ejemplos, un sector puede abarcar los CTB que se encuentran en más de una columna.

15 En algunos ejemplos, el ancho de una columna se puede ajustar en un encabezado de sector o una estructura de sintaxis comparable, permitiendo así "equilibrar la carga" de los recursos computacionales en el codificador. Tal ajuste del ancho de columna puede ser ventajoso si la complejidad del contenido es diferente en diferentes áreas horizontales en una secuencia de imagen de origen.

En algunos ejemplos, se puede usar un elemento de sintaxis de alto nivel para fijar el tamaño de todas las columnas de una imagen, permitiendo así una asignación eficiente de cada columna a un único procesador o equivalente en un escenario de codificación o decodificación paralela.

20 En algunos ejemplos, se puede imponer una restricción para el ancho horizontal mínimo y/o máximo de una columna. Tal restricción puede en algunos casos ayudar a la paralelización y puede mejorar la eficiencia de codificación. La restricción puede, por ejemplo, codificarse en la especificación de nivel de un estándar, o en un elemento de sintaxis de alto nivel, o se puede adaptar optimizando una función que puede implicar el tamaño de MTU y los bits promedio por CTB (que puede derivarse de la tasa de bits, la tasa de fotograma y el tamaño de fotograma en CTB).

25 En algunas realizaciones, se puede usar un orden de escaneo de CTB para decodificar y/o codificar, según el cual los CTB se pueden procesar en una secuencia ordenada, en un primer nivel, por columna de izquierda a derecha en todas las columnas de la imagen, y un segundo nivel por debajo del primer nivel, por CTB de izquierda a derecha y de arriba a abajo en todos los CTB dentro de una columna dada.

30 En un aspecto amplio, se proporciona un método para codificar una imagen de video que comprende una pluralidad de bloques de árbol codificados (CTB) dispuestos en al menos dos filas y en al menos una columna de CTB. El método puede implicar codificar un ancho de columna correspondiente de cada al menos una columna, cada ancho de columna mayor que cero y una suma de todos los anchos de columna iguales a una dimensión horizontal de la imagen de video, y codificar al menos un sector de la imagen de video. En algunos casos, el al menos uno de los sectores puede incluir al menos dos CTB ubicados en al menos dos filas de CTB.

35 En otro aspecto amplio, se proporcionan uno o más medios legibles por ordenador no transitorios en los que se almacenan instrucciones dispuestas para permitir que uno o más procesadores realicen un método para codificar video comprendiendo una imagen de video una pluralidad de bloques de árbol codificados (CTB) dispuestos en al menos dos filas y al menos una columna de CTB. El método puede incluir codificar un ancho de columna correspondiente de cada al menos una columna, cada ancho de columna mayor que cero y una suma de todos los anchos de columna iguales a una dimensión horizontal de la imagen de video, y codificar al menos un sector de la imagen de video. En algunos casos, el al menos un sector puede incluir al menos dos CTB ubicados en al menos dos filas de CTB.

En algunos ejemplos, de acuerdo con cualquiera de los dos aspectos anteriores, el método puede incluir además romper al menos una forma de predicción o filtrado en bucle en los límites de columna entre columnas adyacentes de CTB.

En algunos ejemplos, de acuerdo con cualquiera de los dos aspectos anteriores, el método puede incluir además codificar al menos una de varias columnas y un ancho de una columna en un conjunto de parámetros.

45 En algunos ejemplos, según cualquiera de los dos aspectos anteriores, el ancho de columna correspondiente de al menos una columna puede ser variable entre al menos dos filas de CTB en la imagen de video. En tales realizaciones, dicha variabilidad puede controlarse codificando el ancho de columna correspondiente en un encabezado de sector.

50 En algunos ejemplos, según cualquiera de los dos aspectos anteriores, el método puede incluir además codificar la pluralidad de CTB siguiendo un orden de escaneo de CTB, según el cual la pluralidad de CTB se codifica secuencialmente por columna de izquierda a derecha a través de una pluralidad de columnas, y secuencialmente por CTB de izquierda a derecha y de arriba a abajo dentro de cada una de la pluralidad de columnas.

Aún en otro aspecto más amplio, se proporciona un método para decodificar una imagen de video codificada que comprende una pluralidad de bloques de árbol codificados (CTB) dispuestos en al menos dos filas y en al menos una columna de CTB. El método puede incluir obtener al menos uno de (i) un ancho de columna correspondiente de cada al

menos una columna y (ii) varias columnas, derivando el ancho de columna correspondiente de cada al menos una columna (si no se obtiene a partir de la imagen de video codificada) y decodificar al menos un sector de la imagen de video codificada. En algunos casos, el al menos un sector puede incluir al menos dos CTB ubicados en al menos dos filas de CTB.

- 5 En aún otro aspecto más amplio, se proporciona uno o más medios legibles por ordenador no transitorios en los que se almacenan instrucciones dispuestas para permitir que uno o más procesadores realicen un método para decodificar video comprendiendo una imagen de video codificada una pluralidad de bloques de árbol codificados (CTB) dispuestos en al menos dos filas y al menos una columna de CTB. El método puede incluir obtener al menos uno de (i) un ancho de columna correspondiente de cada al menos una columna y (ii) varias columnas, derivando el ancho de columna correspondiente de cada al menos una columna (si no se obtiene a partir de la imagen de video codificada) y decodificar al menos un sector de la imagen de video codificada. En algunos casos, el al menos un sector puede incluir al menos dos CTB ubicados en al menos dos filas de CTB.

En algunos ejemplos, según cualquiera de los dos aspectos anteriores, el método puede incluir además romper al menos una forma de predicción o filtrado en bucle en los límites de columna entre columnas adyacentes de CTB.

- 15 En algunos ejemplos, según cualquiera de los dos aspectos anteriores, al menos uno del ancho de columna correspondiente de la al menos una columna y el número de columnas pueden obtenerse a partir de un conjunto de parámetros.

- 20 En algunos ejemplos, según cualquiera de los dos aspectos anteriores, el ancho de columna correspondiente de al menos una columna puede ser variable entre al menos dos filas de CTB en la imagen de video codificada. En tales realizaciones, dicha variabilidad puede controlarse codificando un ancho de columna en un encabezado de sector.

En algunos ejemplos, según cualquiera de los dos aspectos anteriores, el método puede incluir además decodificar la pluralidad de CTB siguiendo un orden de escaneo de CTB, según el cual la pluralidad de CTB se decodifica secuencialmente por columna de izquierda a derecha a través de una pluralidad de columnas, y secuencialmente por CTB de izquierda a derecha y de arriba a abajo dentro de cada una de la pluralidad de columnas.

- 25 En otro aspecto amplio aún, se proporciona un decodificador de video para una imagen de video codificada que comprende una pluralidad de bloques de árbol codificados (CTB) dispuestos en al menos dos filas y en al menos una columna de CTB. El decodificador de video puede incluir una pluralidad de decodificadores de columna, con cada decodificador de columna configurado para decodificar sectores de la imagen de video codificada ubicada dentro de al menos una columna de CTB asignados a ese decodificador de columna.

- 30 En algunos ejemplos, según el aspecto anterior, cada sector puede estar ubicado completamente dentro exactamente de una columna.

En algunos ejemplos, según el aspecto anterior, cada decodificador de columna puede ser un proceso que se ejecuta en al menos uno de uno de un núcleo de un procesador multinúcleo, o de un multiprocesador.

### Breve descripción de los dibujos

- 35 Varias características y ventajas de las realizaciones de la presente invención resultarán evidentes a partir de la siguiente descripción detallada, tomada en combinación con los dibujos adjuntos, en los que:

La FIG. 1 es un diagrama que ilustra una imagen ejemplar que tiene sectores de orden de escaneo y direcciones de macrobloques;

La FIG. 2 es un diagrama que ilustra columnas y sectores según una realización de la invención;

- 40 La FIG. 3 es un diagrama que ilustra columnas y sectores (local de una columna dada) según una realización de la invención;

La FIG. 4 es un diagrama de flujo que ilustra el funcionamiento de un decodificador ejemplar según una realización de la invención;

La FIG. 5 es un diagrama que ilustra la modificación del ancho de columna según una realización de la invención;

- 45 La FIG. 6 es un diagrama de flujo que ilustra el funcionamiento de un decodificador ejemplar cuando se decodifica un sector según una realización de la invención;

La FIG. 7 es un diagrama de bloques de un decodificador con dos procesadores de columna según una realización de la invención; y,

- 50 La FIG. 8 es un diagrama de bloques que ilustra una implementación basada en un sistema de procesamiento de datos (por ejemplo, un ordenador personal ("PC")) según una realización de la invención.

Se observará que, a lo largo de los dibujos adjuntos, las características similares se identifican por números de referencia similares.

### Descripción detallada de realizaciones

5 En la siguiente descripción, se exponen detalles para proporcionar una comprensión de la invención. En algunos casos, cierto software, circuitos, estructuras y métodos no han sido descritos o mostrados en detalle para no oscurecer la invención. El término "sistema de procesamiento de datos" se usa en la presente memoria para referirse a cualquier máquina para procesar datos, incluidos los sistemas informáticos, dispositivos inalámbricos y las disposiciones de red descritas en la presente memoria. Las realizaciones de la presente invención pueden implementarse en cualquier lenguaje de programación de ordenador siempre que el sistema operativo del sistema de procesamiento de datos proporcione las facilidades que pueden soportar los requisitos de estas realizaciones. Las realizaciones de la presente invención también pueden implementarse en hardware o en una combinación de hardware y software.

Las realizaciones de la presente invención se refieren a la segmentación de imágenes en compresión de video utilizando columnas y sectores.

15 Con referencia a la FIG. 2, se muestra una segmentación ejemplar 200 de una imagen 201 según realizaciones de la invención. La imagen 201 se divide en una pluralidad de Bloques de Árbol Codificados o macrobloques (en adelante denominados "CTB"). En este ejemplo, la imagen 201 tiene dimensiones de 6x4 CTB. Los límites de los CTB se muestran generalmente como líneas divisorias. La naturaleza de los CTB y sus dimensiones específicas son variables y pueden modificarse de acuerdo con diferentes realizaciones. Aunque la descripción siguiente supone que los CTB incluidos en la imagen 201 son cuadrados y de igual tamaño (como es común para macrobloques en H.264, por ejemplo), ciertas realizaciones de la invención pueden no requerir tal propiedad y pueden ser operables con CTB de diferentes tamaños, siempre que en tales realizaciones sea posible definir una o más columnas horizontales utilizando esos CTB de tamaño desigual. Por ejemplo, las realizaciones de la invención pueden funcionar con pares de macrobloques como se usa en la codificación entrelazada H.264, o con el modo de actualización de resolución reducida de H.263. Las realizaciones también pueden ser operables si hubiera CTB de, por ejemplo, 32x32, 16x16 y 8x8 muestras en la misma imagen.

25 Dos columnas 202, 203 se definen en la imagen 201 y se muestran divididas por un límite de columna representado en negrita, línea continua 204. El ancho de cualquier columna puede ser un múltiplo entero del tamaño de un CTB (o un múltiplo entero del tamaño del CTB más pequeño posible) utilizado dentro de la imagen 201. Las columnas pueden tener un ancho igual o desigual dentro de la imagen 201. Como se muestra, la columna 202 abarca los CTB 1 a 8, y la columna 203 abarca los CTB 9 a 24. La columna 202 tiene dos CTB de ancho, mientras que la columna 203 tiene cuatro CTB de ancho. Sin embargo, las realizaciones descritas no se limitan a este número de columnas ni a los anchos de columna ejemplares mostrados en la FIG. 1.

35 También se muestran dos sectores definidos en la imagen 201 y delineados por la línea 205 de límite discontinua en negrita. El primer sector abarca los CTB con direcciones CTB 1 a 14. El segundo sector abarca los CTB con direcciones 15 a 24. Los CTB 1-8 del primer sector se encuentran en la columna 202, mientras que los CTB 9-14 del primer sector se encuentran en la columna 203. Por lo tanto, el primer sector cubre los CTB que forman parte de las columnas 202 y 203. El segundo sector incluye los CTB 15-24, todos ubicados en la columna 203. En consecuencia, el orden de escaneo de los CTB dentro de la imagen 201 ha cambiado desde el orden de escaneo por imagen, como se muestra en la FIG. 1, a un orden de escaneo por columna, como se muestra en la FIG. 2. De acuerdo con el orden de escaneo por columna, los CTB de la columna 202 se enumeran primero en un orden de escaneo local de columna (de izquierda a derecha, de arriba a abajo en toda la columna 202), antes de enumerar los CTB de la columna 203 (por un orden de escaneo local de columna definido de manera similar). Si/donde sea apropiado, el orden de escaneo por columna que se muestra en la FIG. 2 se puede extender a imágenes para las que se han definido más de dos columnas.

45 El flujo de bits codificado 206 representa la imagen 201 y puede contener un conjunto 207 de parámetros (o puede incluir una referencia a un conjunto de parámetros que se ha descodificado previamente, ya que múltiples imágenes pueden referirse al mismo conjunto de parámetros). El conjunto 207 de parámetros puede contener información 208 de frontera de columna, que identifica directa o indirectamente el ancho de una o más columnas, por ejemplo, columnas 202 y 203, como se describe más adelante. Cada uno de los sectores codificados puede comprender un encabezado de sección correspondiente 209, 211 y los datos de sector correspondientes 210, 212, respectivamente.

50 Segmentación de imágenes con sectores que abarcan los límites de las columnas, como se muestra en la FIG. 2, puede proporcionar una o más ventajas, tales como que el tamaño de todos los sectores codificados (con la posible excepción del último sector de una imagen dada) se puede ajustar para que coincida con el tamaño 213 de MTU tanto como sea posible, al menos parcialmente tenga en cuenta que los sectores codificados pueden no incluir partes de CTB codificados. En el ejemplo de la FIG. 2, los dos sectores codificados (comprendiendo cada uno los encabezados 209, 211 de sectores respectivos y los datos 210, 212 de sectores) se muestran aproximadamente del mismo tamaño para indicar que estos sectores codificados transportan aproximadamente el mismo número de bits, que puede ser un número menor, pero cercano al tamaño 213 de MTU. El tamaño 213 de MTU se muestra como un intervalo, que, en este ejemplo, es constante para los dos sectores. Aunque, en la mayoría de los casos, el tamaño 213 de MTU es una propiedad de red y constante para cualquier paquete, con el propósito de optimizar la carga útil de los paquetes mediante

el uso de sectores, otras consideraciones pueden instar a la adopción de un tamaño objetivo variable para los sectores codificados. Por ejemplo, para la resistencia al error, puede ser útil que algunos paquetes lleven copias redundantes de conjuntos de parámetros además de su carga útil de sector. En tales casos, el tamaño 213 de MTU puede diferir de un paquete a otro y, por lo tanto, también de un sector codificado a un sector codificado. El término "tamaño de MTU" se usa en este sentido amplio a lo largo de la descripción.

Una ventaja de permitir que los sectores abarquen varias columnas a través de los límites entre ellos es permitir una buena coincidencia de tamaño de MTU. Con sectores capaces de abarcar varias columnas, ninguno de los sectores tendrá un tamaño artificialmente limitado debido a la presencia de un límite de columna. Otra ventaja puede ser que es posible representar la imagen completa como un solo sector (por ejemplo, donde el tamaño de la MTU es, en la mayoría de los casos, mayor que el tamaño de la imagen codificada), al tiempo que permite la paralelización. Por conveniencia, se puede suponer que un límite de columna rompe todas las formas de predicción o filtrado en bucle. Bajo este supuesto, una configuración ejemplar de un codificador paralelo puede ser la siguiente. Cada procesador/núcleo del decodificador paralelo puede comenzar a codificar el área espacial de una columna, comenzando con el CTB superior izquierdo dentro de la columna y trabajando secuencialmente en orden de escaneo hasta el macrobloque inferior derecho, es decir, de izquierda a derecha y de arriba hacia abajo. Suponiendo N columnas en una imagen y N procesadores/núcleos con el decodificador paralelo, este enfoque de codificación puede generar N flujos de bits secundarios generados, uno de cada procesador/núcleo. Los N flujos de bits secundarios pueden combinarse para formar un único flujo de bits que representa la imagen completa. Un decodificador que recibe dicho flujo de bits puede, a partir del conjunto de parámetros (por ejemplo, incluido en el flujo de bits), determinar que se le está asignando la tarea de decodificar en paralelo en N columnas. El decodificador puede procesar previamente el flujo de bits combinado decodificando primero solo los elementos de sintaxis incluidos en el flujo de bits (sin realizar también una reconstrucción computacionalmente intensiva de CTB) para identificar los límites entre las columnas en el dominio del flujo de bits. Como ejemplo, uno de estos límites estaría ubicado entre CTB 8 codificado y CTB 9 codificado de la imagen 201. Posteriormente, el decodificador puede extraer y/o copiar los flujos de bits secundarios, divididos dentro del flujo de bits combinado en los límites previamente identificados, a los procesadores/núcleos para su procesamiento. Los procesadores/núcleos dentro del decodificador pueden reconstruir independientemente sus columnas asignadas en el dominio de píxeles, que luego pueden copiarse juntas para reconstruir la imagen completa. Otras estrategias de implementación posibles se explican con más detalle a continuación.

Una desventaja potencial de tener sectores que se extienden a través de los límites de la columna puede surgir debido a dificultades en la implementación de la paralelización, por ejemplo, cuando solo un subconjunto (pequeño) de predicción y/o mecanismos de filtrado de bucle se rompen por un límite de columna. Por ejemplo, si la columna 202 se decodificara en un primer procesador y la columna 203 se decodificara en un segundo procesador, entonces el estado del motor de decodificación (incluida información como el estado CABAC, el estado de predicción de movimiento, etc.), o al menos aquellas partes del mismo que se relacionan con las herramientas de predicción o filtro de bucle que no se rompen por los límites de las columnas, en al menos algunos casos tendrían que pasar del primer procesador al segundo procesador después de la decodificación de CTB 8, que en el orden de escaneo local de columna es el último macrobloque en la columna 202. La implementación y la complejidad computacional de este traspaso pueden depender, por ejemplo, de las herramientas de predicción y filtro de bucle que se ven interrumpidas por la presencia de un límite de columna. Si, por ejemplo, un límite de columna rompe cualquier forma de predicción, pero la predicción en la codificación de entropía, entonces solo se debe entregar la información relativa al motor de codificación de entropía. La solicitud de patente de EE. UU., en trámite número 13/336,475, presentada el 23 de diciembre de 2011, titulada "METHOD AND SYSTEM FOR SELECTIVELY BREAKING PREDICTION IN VIDEO CODING" ("MÉTODO Y SISTEMA PARA ROMPER SELECTIVAMENTE PREDICCIÓN EN CODIFICACIÓN DE VIDEO"), expone, entre otras cosas, técnicas para controlar la cantidad de datos que han de traspasarse cuando un sector alcanza un límite de una columna.

La FIG. 3 ilustra la interacción entre sectores y columnas de una imagen de video en un caso 300 menos general, en particular, que está optimizado para el procesamiento en paralelo de procesadores igualmente potentes (u otras entidades de procesamiento en paralelo, tales como núcleos de procesador), a expensas de la mejor posible coincidencia de tamaño de MTU. (Dicha optimización también puede depender de las herramientas de predicción y filtrado de bucles que se rompen mediante el uso de columnas; la solicitud de patente de EE. UU., en trámite número de serie 13/336,475, expone, entre otras cosas, técnicas para controlar las herramientas afectadas por los límites de columna).

Como se ve en la FIG. 3, una imagen 301 se divide en dos columnas 302, 303 de igual ancho (en este ejemplo, tres CTB de ancho), lo que da como resultado un número similar de CTB que se ha de asignar a cada una de las columnas 302, 303. Bajo el supuesto de que una columna determinada es procesada por un procesador o núcleo dado, la división de la imagen 301 en columnas 302, 303 de anchos iguales también puede dar como resultado una carga aproximadamente igual para cada procesador o núcleo. El límite de la columna 304 entre las columnas 302 y 303 se muestra como una línea en negrita. Todos los CTB de la columna 302 están codificados en un solo sector 305. Mientras que en la columna 303, se definen dos sectores 306, 307 e incluyen colectivamente todos los CTB de la columna 303, con cada sector 306, 307 definido para mantenerse dentro de las restricciones de tamaño de MTU. Por ejemplo, sectores 305, 306 y 307 pueden satisfacer, cada uno, restricciones de tamaño de MTU, a pesar de contener un número diferente de macrobloques (doce, ocho y cuatro, respectivamente) donde se requiere un mayor número promedio de bits para los CTB de la columna 303 con relación a los CTB de la columna 302, lo que puede ocurrir si/cuando una parte de la imagen 201 es más "compleja" que la otra. El límite 308 de sector entre los sectores 306 y 307 dentro de la columna 303 se

indica con una línea en negrita, discontinua. Debe observarse que los sectores 305 y 306 son considerablemente más "compactos" que los sectores que contienen un número similar de CTB, pero dispuestos en orden de escaneo de imágenes. Por lo tanto, el uso de las columnas 302 y 303 puede no solo facilitar el uso del procesamiento paralelo (como se describirá con mayor detalle a continuación), sino que también puede ayudar a la eficacia de la codificación. El sector 5 307, en este ejemplo, aunque no es muy compacta en comparación con los sectores 305 y 306, aún puede ser más compacta de lo que hubiera sido sin el uso de las columnas 302 y 303. Una vez más, el orden de escaneo de los CTB se cambia como se describió anteriormente.

El flujo de bits codificado 309 puede contener un conjunto 310 de parámetros que comprende información 311 de límite de columna. Mientras que el conjunto 309 de parámetros puede mostrarse en la FIG. 3 como adyacente a los sectores 10 codificados de la imagen codificada, en diversas realizaciones, puede ser posible que el conjunto de parámetros esté ubicado en otra parte del flujo de bits 309 o incluso en algunos casos sea transportado fuera de banda.

La imagen codificada contiene tres sectores codificados (es decir, sector 0, sector 1 y sector 2), cada uno de los cuales comprende un encabezado 312, 314, 316 de sector correspondiente y los datos 313, 315, 317 de sector correspondiente, respectivamente. El sector 0 codificado incluye el encabezado 312 codificado y los datos 313 15 codificados del sector, que incluyen la información codificada del sector 305, y se representa con un tamaño (en bits codificados) en algún lugar por debajo del tamaño 318 de MTU. El sector 1 codificado incluye el encabezado 314 de sector codificado y los datos 315 de sector codificados (correspondientes al sector 306). En este ejemplo, cuando se codifica el sector 1, el codificador puede llenar el paquete casi hasta el tamaño 318 de MTU con el encabezado 314 de sector y los datos 315 de sector. Hay que tener en cuenta que no siempre es posible llenar el paquete hasta el último bit 20 del tamaño 318 de MTU porque no es posible fragmentar un CTB en más de un sector. En consecuencia, sujeto a esta restricción, puede haber restos de bits adicionales después de la codificación del último CTB dentro del sector. Finalmente, el sector 2 incluye el encabezado 316 del sector y los datos 317 del sector (correspondientes al sector 307), y es considerablemente más pequeño que el tamaño 318 de MTU porque hay menos CTB para codificar en este sector, por ejemplo, en comparación con el sector 0 y el sector 1.

Se puede especificar un ancho estático de las columnas dentro de una imagen en el nivel de secuencia (por ejemplo, en un conjunto de parámetros de secuencia), en el nivel de imagen (por ejemplo, en un conjunto de parámetros de imagen) o en otras estructuras de datos en el flujo de bits que cubren al menos una imagen codificada. Especificar los límites de las columnas de una vez para todos los sectores en una secuencia o imagen tiene ventajas para las aplicaciones cuyos sectores no varían en ancho de una imagen a otra, ya que la sobrecarga asociada con la especificación de una 30 estructura de sector diferente no se requiere para cada imagen. Además, tiene la ventaja de permitir eficientemente la distribución de bits codificados entre múltiples procesadores o equivalentes en un escenario de decodificación paralela. Los ajustes dinámicos de ancho de columna se describen más adelante.

Los límites de las columnas se pueden especificar utilizando una o más estructuras de datos, parámetros, reglas, 35 indicadores u otra sintaxis adecuada. Usando el conjunto de parámetros de secuencia (SPS) como la estructura de sintaxis ejemplar de alto nivel para transmitir los límites de las columnas, el SPS puede incluir, por ejemplo:

(1) una variable  $N$  por la cual se codifica el número de columnas por imagen. En tales casos, si/cuando sea necesario y / o apropiado, el ancho de cada columna (en unidades de CTB) puede derivarse dividiendo el ancho de una imagen (en unidades de CTB) por  $N$ , si el ancho de la imagen es divisible por  $N$ . De lo contrario, el ancho de  $N-1$  columnas se puede 40 calcular dividiendo el ancho de la imagen por  $N-1$ , por lo que el ancho de la columna final (por ejemplo, la columna más a la derecha de la imagen) sería el resto de la operación de división.

(2) una variable  $W$  por la cual se codifica el ancho máximo (en unidades de CTB) de una columna. En tales casos, deje que  $N$  sea el ancho de la imagen dividido por  $W$ . Luego, el ancho de las primeras  $N$  columnas es igual a  $W$ , y el ancho de la columna final sería el resto de la operación de división.

(3) una variable  $N$  que contiene el número de columnas y un campo de enteros  $N-1$  de largo que contiene valores para 45 representar los anchos de cada columna. Esta opción permite un ancho de columna no uniforme como se muestra en la FIG. 2.

Aún otras representaciones de información de límite de columna que son adecuadas para diferentes aplicaciones pueden ser evidentes después de familiarizarse con la presente exposición y se pretende que queden cubiertas dentro del alcance de las realizaciones descritas.

Un estándar de codificación de video puede establecer ventajosamente límites para un ancho de columna mínimo o máximo que puede ocurrir en un flujo de bits. Los límites se pueden especificar, por ejemplo, en las especificaciones de perfil y nivel. Por ejemplo, un perfil diseñado para procesamiento paralelo de alto nivel en caso de decodificación de video de alta resolución puede requerir que las columnas se usen con un ancho no mayor que, por ejemplo, 1920 50 muestras (una resolución de 1080p, que comúnmente se puede decodificar en un solo núcleo, tiene una resolución de 1920 muestras horizontales). Por otro lado, el perfil puede requerir, por ejemplo, un ancho mínimo de columna de no menos de 640 muestras. El límite inferior puede ayudar a evitar los peores escenarios y los sectores "alargados" en la dimensión  $Y$ . El límite superior puede ser útil por varias razones diferentes. En primer lugar, como ya se describió, la eficiencia de codificación puede mejorarse debido a sectores cada vez menos alargados. En segundo lugar, el

5 decodificador puede basarse en decodificadores de imagen secundaria, cada uno capaz de decodificar una sola columna (en adelante, llamados "decodificadores de columna"), pero no necesariamente capaz de manejar una imagen completa. En tercer lugar, los requisitos de memoria en los decodificadores de columna se pueden reducir ya que las memorias temporales de línea no necesitan abarcar todo el ancho de la imagen, sino solo el ancho de la columna (más quizás el relleno dependiente de la implementación). Otras ventajas de dicho límite superior en el ancho de columna también pueden ser evidentes.

10 A continuación, se describe la operación de un decodificador en un flujo de bits que contiene sectores y columnas, y se observa cómo dicha operación puede diferir del proceso de decodificación de un flujo de bits que se ha generado sin el uso de columnas. Por conveniencia, en la siguiente descripción se supondrá que la codificación del ancho de columna se realizó de acuerdo con la opción (3) anterior. Sin embargo, la descripción no debe entenderse como limitada solo a este tipo de codificación y se aplicará también a las otras dos opciones descritas anteriormente, si/donde sea necesario, realizando modificaciones y/o alteraciones adecuadas en la descripción.

15 En los estándares modernos de codificación de video, los parámetros pertenecientes a más de un sector pueden almacenarse en estructuras de sintaxis de nivel superior conocidas como conjuntos de parámetros. En estándares más antiguos, estos parámetros se pueden almacenar en encabezados tales como la imagen, GOP, o encabezados de secuencia. Para facilitar la siguiente descripción, se suponen conjuntos de parámetros a partir de ahora, aunque las realizaciones de la invención pueden utilizar igualmente otras formas de estructuras de sintaxis de alto nivel.

20 La FIG. 4 muestra un método 400 que ilustra el funcionamiento de un decodificador ejemplar de acuerdo con realizaciones de la invención. En primer lugar, para poder decodificar datos de sector, se pueden decodificar uno o más conjuntos de parámetros (401) como parámetros almacenados en los conjuntos de parámetros y a los que se puede acceder a través de una referencia almacenada en el encabezado del sector. En algunas realizaciones, los conjuntos de parámetros pueden contener información perteneciente a, y que es suficiente para definir completamente, el mecanismo de columna utilizado dentro de las imágenes a decodificar. Por ejemplo, los conjuntos de parámetros pueden incluir parámetros utilizados para indicar el número de columnas (es decir, "num\_column\_minus\_1" en la FIG. 4) y el ancho de cada una de esas columnas excluyendo la última columna (es decir, "columnWidth[]" en la FIG. 4). El ancho de la última columna en la imagen codificada puede ser igual a la distancia entre el límite de la columna más a la derecha y el límite de la imagen a la derecha.

30 Los parámetros almacenados (por ejemplo, "num\_column\_minus\_1" y "columnWidth[]") pueden decodificarse (402) como una parte de la decodificación del conjunto de parámetros (401). La decodificación (402) puede ser una etapa secundaria de la decodificación (401) y puede realizarse en cualquier momento apropiado durante la decodificación del conjunto de parámetros (401). El orden exacto puede, por ejemplo, estar limitado por la sintaxis de los propios conjuntos de parámetros (que pueden ordenar la ubicación de la información decodificada en 402 en relación con otras entradas en el conjunto de parámetros).

35 Durante la decodificación (401) del conjunto de parámetros, la información relacionada con las herramientas de predicción y/o filtrado en bucle también se puede decodificar, como se describe más adelante en la solicitud de patente de EE. UU., en tramitación número 13/336,475. La decodificación de información relacionada con herramientas de predicción y/o de filtrado en bucle se indica generalmente en la FIG. 4 en 403, y también puede considerarse como una parte de la decodificación (401) del conjunto de parámetros.

40 Después de la decodificación (401) de la información del conjunto de parámetros en el método 400, puede comenzar la decodificación y reconstrucción de sector.

45 En primer lugar, se puede decodificar (404) un encabezado de sector. En el encabezado del sector, puede haber información disponible que permita que el decodificador asocie el sector dado con la columna en la que se incluye ese sector dado. Por ejemplo, el encabezado del sector puede contener una dirección CTB (es decir, como se indica mediante la "first\_ctb\_address" en la FIG. 4) que representa la dirección del primer CTB incluido dentro del sector. La dirección de CTB codificada puede decodificarse (405) y usarse como el valor inicial para un parámetro utilizado por el decodificador para realizar un seguimiento de la dirección CTB actual (es decir, "currentCTB" en la FIG. 4) durante el proceso de decodificación. Luego, la columna en la que reside el sector se puede identificar (406) en función de la dirección del primer CTB incluido dentro del sector asociando ese primer CTB con una columna particular definida en la imagen codificada.

50 Por ejemplo, haciendo referencia de nuevo a la FIG. 1, las direcciones CTB (de las cuales la dirección CTB decodificada en 405 es un ejemplo) pueden ser representaciones codificadas por entropía de las ubicaciones de cada CTB dado en la imagen, en orden de escaneo, como se indica mediante los números de la FIG. 1 centrados en los CTB. Para asociar sectores con columnas, el componente horizontal de la dirección CTB del primer CTB en el sector dado se puede comparar con los límites de la columna, definidos para la imagen codificada, que pueden derivarse a partir de la información de ancho de columna obtenida durante la decodificación de los conjuntos de parámetros (y posiblemente pueden modificarse durante la posterior decodificación de sector, como se describe a continuación). En algunas realizaciones de la invención, donde cada sector dado puede estar limitado a residir en su totalidad dentro de una columna dada, los sectores pueden asociarse con columnas identificando la columna en la que reside el primer CTB dentro de ese sector dado (porque, por definición, todos los CTB restantes en el sector se ubicarían dentro de la misma

columna). Esta identificación puede llevarse a cabo de acuerdo con el bloque 611 y 612 de la FIG. 6, lo que se trata con más detalle a continuación.

Refiriéndose nuevamente a la FIG. 4, se identifica (406) la primera columna (inicial) para el sector a decodificar. La decodificación de los CTB codificados en el sector puede entonces proseguir como se especifica en uno o más estándares de compresión de video para la decodificación de CTB a los que puede cumplirse la decodificación, que puede incluir etapas tales como decodificación por entropía, compensación de movimiento para formar un predictor, cuantificación inversa y transformación inversa de coeficientes para calcular un residual, agregando el residual al predictor compensado por movimiento, desbloqueando-filtrando en los CTB las muestras resultantes, y así sucesivamente, con la excepción del manejo de direcciones de CTB y el uso de datos fuera del sector en función de su disponibilidad para la predicción. A continuación, se describe un ejemplo que describe una posible estrategia de implementación, aunque otras estrategias también pueden ser evidentes después de familiarizarse con la presente exposición.

En primer lugar, la textura, el movimiento y otra información de los CTB pueden decodificarse y el CTB actual que es manejado por el decodificador puede reconstruirse (407), lo que puede implicar cualquiera o todas las diferentes etapas mencionadas anteriormente en el contexto de la decodificación de CTB, mientras también observar la disponibilidad de información de predicción fuera del límite de sector o columna.

Entonces, la dirección CTB actual puede incrementarse (408). Si se determina (409) que la dirección de CTB incrementada todavía queda a la izquierda del límite derecho de la columna, no se requiere una actualización adicional de la dirección de CTB (indicando en la FIG. 4 mediante la flecha que puentea el bloque 410 directamente al bloque 411). Sin embargo, si se determina (409) que la dirección CTB incrementada está fuera de la columna (es decir, igual o a la derecha del límite derecho de la columna), entonces la dirección CTB se restablece (410) a la dirección del primer CTB en la siguiente línea de CTB dentro de la columna. En otras palabras, el componente horizontal de la dirección CTB se establece en el CTB más a la izquierda de la columna (como lo indica el parámetro "colBound[]" en la FIG. 6), y el componente vertical de la dirección CTB se incrementa en uno.

Si, después de la ejecución de los bloques 408 a 410, el componente vertical de la dirección CTB excede el tamaño vertical de la imagen, entonces el sector puede abarcar más de una columna. En este caso, la dirección CTB puede restablecerse, por ejemplo, a la dirección CTB superior izquierda en la columna a la izquierda de la columna anterior. Los cambios de decodificación y dirección CTB continúan en esta columna.

En este punto, se determina (411) si hay más datos (que indican uno o más CTB) disponibles en el sector a decodificar. Si se determina (411) que quedan más datos en el sector a decodificar, entonces el método 400 continúa (412) de regreso al bloque 407, y la decodificación continúa con el siguiente CTB dentro del sector. Sin embargo, si se determina (411) que no quedan datos en el sector a decodificar, entonces el método 400 continúa (413) de regreso al bloque 404, y la decodificación continúa con el siguiente sector en la imagen.

Según las realizaciones de la invención, un decodificador puede leer información de límite de columna (inicial) a partir de estructuras de alto nivel tales como conjuntos de parámetros de imagen o secuencia. Por ejemplo, esto se puede implementar en un mecanismo que analiza primero un elemento de sintaxis (es decir, "num\_column\_minus\_1") del flujo de bits. Este elemento de sintaxis puede ser ventajosamente parte de una estructura de sintaxis de imagen (o superior), tal como un conjunto de parámetros de imagen, conjunto de parámetros de secuencia, encabezado de imagen o estructura comparable. Si el valor de "num\_column\_minus\_1" es mayor que 0, entonces se puede inferir que el número de límites de columna presentes en un flujo de bits conformes será igual al valor almacenado en este parámetro, y allí se pueden analizar los anchos de columna num\_column\_minus\_1. Los anchos de las columnas analizadas correspondientemente se pueden indicar en términos de CTB. La sintaxis descrita anteriormente se especifica en el estilo de la especificación de sintaxis H.264, partes de las cuales se resumen a continuación por conveniencia.

Específicamente, en la descripción de sintaxis que se proporciona a continuación, una variable en negrita (es decir, las líneas 1 y 4 a continuación), en comparación con una línea con formato simple, puede representar bits que se obtienen a partir del flujo de bits como un efecto secundario o en conjunto con la determinación del valor de la variable. Una vez leído, el valor se puede usar en el proceso de decodificación, sin embargo, la variable ya no se pone en negrita, dado que el uso de la variable desde ese punto en adelante ya no está asociado con una búsqueda de flujo de bits.

**num\_column\_minus\_1**

```

si (num_column_minus:1 > 0) {
    para (i = 0; i < num_column_minus_1; i++) {
        columnWidth[i]
    }
}

```

Si los límites de columna se pueden especificar en más de un nivel, según el estándar de compresión dado, entonces la sintaxis de los límites de columna puede ser la misma en varios niveles, por ejemplo, secuencia y/o nivel de imagen. En este caso, ventajosamente, los límites de columna indicados en un nivel relativo más bajo (es decir, el nivel de imagen) pueden tener prioridad sobre los límites de columna indicados en un nivel relativo más alto (es decir, el nivel de secuencia).

Los valores de `columnWidth[]` analizados pueden usarse para derivar los límites de la columna como se muestra a continuación:

```

    si (num_column_minus_1 > 0) {
        columnBoundary[1] = columnWidth[0];
10     para (i=1; i < num_column_minus_1; i++) {
            columnBoundary[i + 1] = columnBoundary[i] + columnWidth[i];
        }
    }

```

A continuación, se describen realizaciones de la presente invención en las que los límites de columna se pueden ajustar durante la codificación de una imagen, es decir, a nivel de sector.

Con referencia a la FIG. 5, se muestra un diagrama que ilustra la modificación 500 de ancho de columna de acuerdo con realizaciones de la invención. En la FIG. 5, la ubicación de los límites de la columna se puede modificar a nivel de sector (por ejemplo, según la información de un encabezado de sector). La modificación 500 puede pertenecer solo a la imagen actual y no a otras imágenes en el flujo de bits codificado. Cuando está implicada la paralelización del procesamiento de codificación y las columnas pueden ser procesadas por procesadores independientes o unidades similares, la modificación 500 puede proporcionar una ventaja ya que el número de CTB por columna se puede ajustar a la carga del procesador posiblemente desde factores de carga externos (otros procesos no relacionados con la codificación del video), la complejidad del contenido (que puede variar de una columna a otra) o aún otros factores.

En la siguiente descripción, por conveniencia, el encabezado del sector se usa para transmitir información de modificación porque, en ciertos contextos, el encabezado del sector puede ser la estructura de sintaxis más apropiada en H.264 para tal propósito. El uso del encabezado de sector para información perteneciente a más de un sector no está necesariamente en línea con el modelo de sintaxis de H.264 (por ejemplo, en H.264, toda la información que pertenece a más de un sector debe incluirse solo en conjuntos de parámetros), pero está en línea con algunos estándares de compresión de video más antiguos (por ejemplo, en H.261, la información de encabezado GOB se puede predecir a partir de encabezados GOB anteriores, y los GOB son al menos en algunos aspectos similares al equivalente a los sectores de H.264). Al mismo tiempo, si se incluyera un mecanismo en los estándares de compresión de video más nuevos, lo que permitiría la codificación de propiedades en el nivel de sector que pertenece a algo más que el sector actual, entonces dicha estructura de sintaxis podría ser en algunos casos una ubicación más adecuada para la información presentada a continuación que en el propio encabezado del sector. Por ejemplo, en WD4 se incluye un Conjunto de Parámetros de Adaptación (APS), que está diseñado para transportar información que probablemente cambie de una imagen a otra (en contraste con el conjunto de parámetros de la imagen, que transporta información que es improbable que cambie de una imagen a otra imagen). En JCT-VC también se están considerando propuestas para permitir "actualizar" un APS en los límites de sector, lo que, si se acepta, también haría que el APS sea una estructura de sintaxis adecuada para la información de actualización de ancho de columna, como se describe a continuación.

En algunas realizaciones de la invención, los límites de la columna que se han cambiado a través del mecanismo de anulación descrito a continuación vuelven a las ubicaciones especificadas en el nivel de imagen o secuencia cuando se completa la codificación/decodificación de la imagen actual. Esta característica puede, al menos en parte, compensar la falta de limpieza arquitectónica del uso del encabezado del sector para codificar la información perteneciente a más de un sector. Por ejemplo, una posible razón para no permitir el uso de encabezados de imagen y estructuras de datos similares en H.264 es evitar la propagación de errores debido a estructuras de encabezado corruptas. Volver a los datos transmitidos en conjuntos de parámetros (que han sido diseñados para ser muy difíciles de corromper) después de cada imagen puede garantizar que la corrupción del encabezado se limite a una imagen dada, en lugar de a todas las imágenes que siguen a la imagen en la que ha ocurrido la corrupción del encabezado.

Se muestran en la FIG. 5 dos límites 501 y 502 de columna originales, que se representan como líneas verticales discontinuas, en negrita. En el ejemplo, se supone que el codificador desea por alguna razón reducir el ancho de columna de la columna 503 más a la izquierda. La motivación del codificador para hacer este ajuste generalmente no está limitada y puede incluir, por ejemplo, aspectos de equilibrio de carga del procesador, o relativamente (con respecto a otras columnas) una alta complejidad de codificación del contenido en la columna más a la izquierda, que conduciría a más bits en los CTB de esta columna y, en consecuencia, da como resultado sectores alargados a menos que se reduzca el ancho de la columna.

Según la modificación 500, el límite 502 de columna abarca toda la altura vertical de la imagen, mientras que el límite 501 de columna abarca solo una parte de la altura vertical completa de la imagen, es decir, las dos filas más superiores de CTB. El resto del límite de la columna está desplazado con respecto al límite 501 de la columna y, en consecuencia, la columna 503 más a la izquierda tiene un ancho variable. Por ejemplo, la columna 503 puede dividirse en dos sectores diferentes de ancho diferente. Como se muestra en la FIG. 5, el sector 504 dentro de la columna 503 puede tener un primer ancho, que se establece por el límite original de la columna 501, mientras que el sector 505 dentro de la columna 503 puede tener un segundo ancho diferente del primer ancho, y que se establece por el límite de la columna de desplazado según la modificación 500.

Los límites de columna se pueden modificar en el nivel de sector incluyendo valores de desplazamiento de límite para desplazar un límite de columna, por ejemplo, a la izquierda, si el valor de desplazamiento es negativo, o alternativamente, por ejemplo, a la derecha, si el valor de desplazamiento es positivo. El encabezado de sector del sector 505 en la columna 503 más a la izquierda puede, por ejemplo, incluir información en la medida en que el límite 501 de columna original entre las columnas más a la izquierda y central, como se define con respecto al sector 504, sea desplazado por un CTB a la izquierda con respecto al sector 505. Como resultado, CTB 22 pertenece ahora a la columna central (en oposición a la columna 503 más a la izquierda), y la codificación continúa en la columna 503 más a la izquierda con el sector 505, que abarca los CTB 20, 21, 30, 31, 40, 41, 50, y 51 en este ejemplo particular.

Hay varios casos de esquina que podrían hacer potencialmente que la implementación de la modificación 500 descrita anteriormente parezca innecesariamente compleja y/o ineficiente desde el punto de vista del almacenamiento en caché. Sin embargo, tales casos de esquina se pueden manejar de manera efectiva mediante una serie de restricciones relacionadas con el movimiento de los límites de la columna que se pueden imponer en diversas realizaciones, y que se describen más adelante.

Durante la decodificación de la imagen, el desplazamiento del límite de la columna puede entrar en vigor comenzando en la línea CTB en la que se encuentra la dirección del primer CTB del sector para contener la información de desplazamiento del límite en su encabezado de sector. A partir de entonces, el desplazamiento del límite puede permanecer en vigor para todas las líneas CTB restantes en la imagen a menos que dicho desplazamiento del límite sea anulado por otro desplazamiento del límite en un encabezado de sector posterior.

Los valores de desplazamiento de límite pueden indicarse en unidades de anchos CTB.

El flujo de bits puede contener un "boundary\_offset\_flag" (es decir, como se ve en la FIG. 6), que puede analizarse desde el flujo de bits. Si el boundary\_offset\_flag tiene un valor de uno, entonces uno o ambos de un left\_boundary\_offset y right\_boundary\_offset pueden seguir en el flujo de bits a través del cual se pueden especificar los desplazamientos de límite de columna a la derecha o izquierda. De lo contrario, si el boundary\_offset\_flag tiene un valor de cero, entonces no puede haber presentes valores de desplazamiento en el flujo de bits después del boundary\_offset\_flag. En el estilo de la especificación de sintaxis H.264, a continuación, se muestra un ejemplo de sintaxis para especificar los valores de desplazamiento:

```

boundary_offset_flag
    si (boundary_offset_flag) {
        left_boundary_offset
        right_boundary_offset
    }

```

Como en el caso anterior, las variables en negrita (es decir, las líneas 1, 3 y 4 anteriores), en comparación con una línea de formato simple (es decir, la línea 2), pueden representar bits que se obtienen del flujo de bits como un efecto secundario o de otro modo en combinación con la determinación del valor de la variable.

La FIG. 6 es un diagrama de flujo que ilustra un método 600 que puede usarse para determinar los límites de columna en base a los valores de desplazamiento de límite analizados como antes, de acuerdo con las realizaciones de la invención.

El método 600 se puede inicializar (601) mediante las siguientes configuraciones, que por conveniencia se expresan en pseudocódigo de estilo C. Dicha inicialización, entre otras cosas, copia los límites analizados a partir de la variable columnBoundary[], que se ha derivado a partir de los elementos de sintaxis de alto nivel descritos anteriormente, de la siguiente manera:

```

pictureWidth = ancho de la imagen // en unidades de CTB
colBound[0] = 0;
colBound[num_column_minus_1 + 1] = pictureWidth;
para (j = 1; j < num_column_minus_1 + 1; j++)

```

```
colBound[j] = columnBoundary[j];
```

Después de ello, se determina (602) si se ha establecido el `boundary_offset_flag` (introducido anteriormente). Si se determina (602) que el indicador no se ha establecido, entonces no se permiten actualizaciones de los límites de las columnas, y el método 600 se deriva (603) para que el sector se decodifique (604) como se describió y especificó previamente en el estándar de codificación de video con el que cumple la codificación, por ejemplo, que puede implicar, para cada CTB del sector, procesos tales como decodificación por entropía, compensación de movimiento para formar un predictor, cuantificación inversa y transformación inversa de coeficientes para calcular un residual, adición del residual al predictor de movimiento compensado, desbloqueo-filtrado en CTB de las muestras resultantes, y así sucesivamente. Luego, si se determina (605), se ha descodificado el último sector de la imagen (una situación que puede probarse mediante muchos mecanismos diferentes, tales como detectar un cambio en el valor de la ID de la imagen, Referencia Temporal, indicación de fecha y hora de presentación, o elemento de sintaxis similar, así como mecanismos de tiempo de espera basados en indicaciones de fecha y hora RTP), luego el método 600 continúa (606) inicializando (601) como se describió anteriormente para procesar la siguiente imagen en el flujo de bits. Sin embargo, si se determina (605) que el sector no fue el último de la imagen, entonces el método 600 continúa (607) determinando (602) si se ha establecido `boundary_offset_flag`.

Si se determina (602) que se ha establecido `boundary_offset_flag`, el encabezado del sector contiene desplazamientos de límite que se pueden analizar y utilizar para modificar `colBound[]`, y el método se deriva (608) para dicha modificación de los límites de columna. La modificación puede comenzar mediante una identificación de la columna a la que pertenece el sector. Dicha identificación puede comenzar a su vez con una inicialización (609), por ejemplo, que puede contener instrucciones como las siguientes:

```
firstCTBMod = // primer CTB de pictureWidth de módulo de sector actual
              // donde el primer CTB es la dirección CTB en orden de escaneo
leftOffset = left_boundary_offset; // como analizado
rightOffset = right_boundary_offset; // según lo analizado
i = 1;
```

En este punto, el método 600 puede buscar a través del estado actual de los límites de la columna (por ejemplo, en `colBound[]`) para identificar la columna particular a la que pertenece el sector que se está decodificando. Por ejemplo, el método 600 puede realizar dicha identificación verificando (611) el componente de dirección horizontal de la dirección CTB del CTB actual (que es el primer CTB en el sector en cuestión), según lo determinado en la inicialización y almacenado en `firstCTBMod`, contra los valores de límite de columna almacenados (`colBound[]`). Al buscar a través de los límites de la columna incrementando (610) un índice de búsqueda adecuado, una vez que se identifica un límite de columna como ubicado a la "derecha" (es decir, que tiene una dirección horizontal mayor en términos de orden de escaneo) del primer CTB de sector actual (como se expresa en `firstCTBMod`), la búsqueda finaliza y el método 600 sale (612) del bucle de búsqueda con el estado superviviente del índice `i` de búsqueda.

Los límites de la columna a la que pertenece el sector que se está decodificando se pueden actualizar (613), de acuerdo con el estado superviviente del índice `i` de búsqueda, añadiendo la información de diferencia `leftOffset` y `rightOffset`, como se inicializó (609), a los límites de columna izquierda y derecha. Dicha actualización puede lograrse, por ejemplo, mediante lo siguiente:

```
colBound[i-1] = colBound[i-1] + leftOffset;
colBound[i] = colBound[i] + rightOffset;
```

En este punto, el método 600 prosigue a la decodificación (604) de sector como se describió anteriormente.

En algunos casos, puede ser ventajoso y/o conveniente imponer restricciones en la indicación de límite de columna, incluida la siguiente restricción:

```
colBound[i] < colBound[i + 1]
```

Dicha restricción requiere que las columnas se numeren de izquierda a derecha en la imagen y también implica que los valores de desplazamiento no se pueden usar para cambiar el orden de los límites de la columna.

Los valores `leftOffset`, `rightOffset` y subsecuentemente derivados de `colBound[]` también pueden estar restringidos de manera que cada columna resultante tenga un ancho que sea mayor o igual que un ancho mínimo de columna. Por ejemplo, el ancho mínimo de columna puede ser específico de nivel y/o especificado en el estándar de compresión de video.

En algunos casos, `leftOffset` y `rightOffset` no pueden usarse para modificar un `colBound[]` usado como límite de sector para un sector previamente codificado.

En algunos casos, el valor de colBound[0] no se puede cambiar desde su valor inicial de '0' para reflejar que la primera columna numerada comienza en el borde más a la izquierda de la imagen. En consecuencia, el valor de leftOffset sería '0' en tales casos cuando  $i = 1$ .

5 En algunos casos, el valor de colBound[num\_columns\_minus1+1] no se puede cambiar desde su valor inicial de pictureWidth. En consecuencia, el valor de rightOffset sería '0' en tales casos cuando  $i = \text{num\_column\_minus\_1} + 1$ .

10 Con referencia a la FIG. 7, se muestra un diagrama de bloques de un decodificador 700 que se basa en una arquitectura de multiprocesador o multinúcleo de acuerdo con las realizaciones descritas. Por conveniencia, se supone que el decodificador 700 funciona en un flujo de bits configurado o adecuado para la decodificación paralela, por ejemplo, al requerir el uso de columnas con ciertas limitaciones en el ancho de la columna, como ya se describió (por ejemplo, límites superior e inferior en el ancho de la columna).

15 En algunas realizaciones, el decodificador 700 puede recibir un flujo de bits 701 codificados en un intérprete 702 de sintaxis de alto nivel. El flujo de bits 701 se muestra en la FIG. 7 como una línea discontinua para distinguir el flujo de bits 701 de los metadatos (que se muestra como una fina línea continua) y de los datos de la muestra (que se muestran en línea continua en negrita). El intérprete 702 de sintaxis de alto nivel puede configurarse para decodificar conjuntos de parámetros y almacenamiento posterior en una base de datos 703 de conjunto de parámetros a la que el intérprete 702 de sintaxis de alto nivel tiene acceso de lectura y escritura. Los conjuntos de parámetros decodificados pueden almacenarse y proporcionarse 704 a uno o más componentes en el decodificador 700, por ejemplo, a un decodificador 705 de columna/sector. En algunas realizaciones, el intérprete 702 de sintaxis de alto nivel puede ser capaz de manejar más partes de, o la totalidad de, la decodificación por entropía de elementos de sintaxis de alto nivel y/o de sintaxis de bajo nivel (por debajo del nivel CU) incluidos en el flujo de bits 701, cuya capacidad puede depender de la asignación de decodificación por entropía a los decodificadores 707, 708 de columna. Los datos 706 de salida del intérprete 702 de sintaxis de alto nivel pueden ser, por ejemplo, componentes de metadatos derivados del flujo de bits 701 que pueden incluir información a partir del flujo de bits, tal como símbolos (creados mediante decodificación por entropía) o memorias temporales que contienen símbolos codificados por entropía, así como información de control.

25 El decodificador 705 de columna/sector puede estar acoplado al intérprete 702 de sintaxis de alto nivel y a la base de datos 703 del conjunto de parámetros, y puede recibir los datos 706 de salida del intérprete 702 de sintaxis de alto nivel. En algunas realizaciones, el decodificador 705 de columna/sector puede decodificar encabezados de sector e identificar la columna a la que pertenece el primer CTB del sector, por ejemplo, teniendo en cuenta la información de límite de columna que puede ubicarse en conjuntos de parámetros, almacenados en la base de datos 703 del conjunto de parámetros, y recuperados 704 basándose en la referencia del conjunto de parámetros en el encabezado del sector. Si el decodificador 705 de sector/columna determina que un sector reside en una sola columna, como se describió anteriormente, entonces el decodificador 705 de sector/columna puede identificar la columna en la que reside el sector identificando la columna a la que pertenece el primer CTB del sector.

35 En base a la información de la columna, el decodificador 705 de columna/sector puede seleccionar un decodificador de columna para la decodificación del sector. Dos decodificadores 707, 708 de columna se muestran en la FIG. 7 como un ejemplo de implementación del decodificador 700 y no están destinados a limitar las posibles implementaciones. Por ejemplo, en realizaciones alternativas, el codificador 700 puede incluir un número mayor o menor de decodificadores de columna, siempre que el codificador 700 incluya al menos un decodificador de columna. Si hay más columnas en el flujo de bits 701 de las que los decodificadores 707, 708 de columna han incluido en el decodificador 700, entonces al menos uno de los decodificadores 707, 708 de columna puede seleccionarse para la decodificación de más de una columna. Alternativamente, si hay más decodificadores 707, 708 de columna disponibles en el decodificador 700 que columnas hay en el flujo de bits 701, entonces uno o más de los decodificadores 707, 708 de columna pueden estar inactivos ya que el decodificador 705 de columna/sector no les asignará ninguna columna. Sin embargo, el decodificador 705 de columna/sector puede asignar columnas en el flujo de bits 701 a los decodificadores 707, 708 de columna que pueden ser manejados por el decodificador 705 de columna/sector como un problema de equilibrio de carga para el que pueden ser evidentes muchas soluciones posibles después de familiarizarse con la presente exposición.

40 Suponiendo que el sector que se va a decodificar actualmente se asigna al decodificador 707 de columna, los datos 709 necesarios para decodificar la columna se pueden enviar al decodificador 707 de columna. Tales datos 709 pueden tener características similares a los datos 706 de salida del 702 intérprete de sintaxis de alto nivel. En algunas implementaciones, el decodificador 705 de columna/sector puede recortar, o marcar como irrelevantes, aquellas partes de los datos 706 de salida que no son necesarias para la decodificación de la columna, con el efecto de que esas partes innecesarias no se envían al decodificador 707 de columna como parte de los datos 709.

55 En algunas realizaciones, el flujo de bits 701 puede configurarse o puede permitir una decodificación completamente o casi completamente independiente de su contenido en decodificadores de múltiples columnas (por ejemplo, 707 y 708 en la FIG. 7). Sin embargo, en otros casos, algunos metadatos o información de datos de muestra pueden necesitar compartirse entre decodificadores de múltiples columnas. Se pueden encontrar más detalles relevantes sobre la naturaleza y la cantidad de datos que pueden necesitar potencialmente compartirse entre decodificadores de múltiples columnas en la solicitud de patente de EE. UU., en tramitación con la presente de número de serie 13/336,475.

En algunas realizaciones, la compartición de información puede manejarse mediante una unidad 710 de compartición configurada adecuadamente. La unidad 701 de compartición puede implementarse como memoria compartida, un mecanismo de cola que permite el intercambio de mensajes u otros mecanismos para la comunicación entre procesadores o entre núcleos. Cuando la compartición de información se habilita mediante la unidad 710 de compartición, puede potencialmente ser necesario compartir tanto la metainformación 711 (p. ej., estado de decodificación por entropía, predictores de vectores de movimiento) como/o los datos 712 de muestra (p. ej., valores de píxeles vecinos de la columna utilizados para predicción intra o compensación de movimiento) entre los decodificadores 707, 708 de columna. La unidad 710 de compartición también puede configurarse adecuadamente para facilitar el "traspaso" de información desde un decodificador 707, 708 de columna a otro en los casos en que un sector se extiende a través de un límite de columna entre columnas adyacentes, con el efecto de que múltiples decodificadores 707, 708 de columna serían asignados a su vez por el decodificador 705 de sector/columna para decodificar dicho sector.

En algunas realizaciones, por ejemplo, en implementaciones del decodificador 700 que utilizan un multiprocesador o un único procesador multinúcleo en un ordenador personal o arquitectura de hardware similar, la implementación de los decodificadores 707, 708 de columna como procesos que se ejecutan en núcleos o procesadores puede ser facilitado en gran medida manteniendo toda la información de estado del decodificador 700, incluidas, por ejemplo, imágenes de referencia completas, en la memoria que se comparte entre los decodificadores 707, 708 de columna. En este caso, la unidad 710 de compartición puede incluir una memoria compartida ayudada en algunos casos posiblemente por herramientas de comunicación entre procesadores tales como semáforos o paso de mensajes implementados, por ejemplo, en un núcleo del sistema operativo.

La salida del decodificador 708 de columna pueden ser muestras 713 parcialmente reconstruidas y metadatos 714 asociados proporcionados a un combinador 715. Del mismo modo, el decodificador 707 de columna puede producir muestras parcialmente reconstruidas y metadatos asociados, similares a las muestras 713 parcialmente reconstruidas y metadatos 714 asociados producidos por el decodificador 708 de columna. Las muestras 713 pueden describirse como reconstruidas "parcialmente" porque para la reconstrucción completa, como se especifica en el estándar de compresión de video, puede requerirse un procesamiento adicional en el generador 717 de muestras basado en imágenes, como se describe más adelante. Las muestras 713 y la salida de metadatos 714 asociados proporcionada por el decodificador 708 de columna pueden combinarse en el combinador 715 con las salidas correspondientes del decodificador 707 de columna para generar una imagen 716 procesada en columna y metadatos 719 asociados.

El generador 717 de muestras basado en imágenes puede configurarse para generar la imagen 718 reconstruida final basada en la imagen 716 procesada en columna y los metadatos 719 asociados, que se reciben desde el combinador 715. La imagen 718 reconstruida final puede usarse posiblemente para la salida por el decodificador 700 y/o como una imagen de referencia que es realimentada a otros componentes dentro del decodificador 700 (indicado más abajo). El generador 717 de muestras basado en imágenes puede ventajosamente implementar tareas que no son necesariamente muy intensivas computacionalmente, pero que requieren acceso a información relacionada con todas las columnas. El filtrado en bucle, el filtrado posterior y la ocultación de errores son ejemplos de esas tareas. Sin embargo, dependiendo de las características del flujo 701 de bits codificado, también puede ser posible que algunas o todas estas tareas puedan implementarse en su lugar en los decodificadores 707, 708 de columna (más detalles de los cuales se pueden encontrar en la solicitud de patente de EE. UU., en tramitación con la presente de número de serie 13/336,475). Por ello, en algunas realizaciones, puede omitirse el generador 717 de muestras basado en imágenes del decodificador 700.

No se muestra específicamente en la FIG. 7 la "realimentación" de las muestras de imagen de referencia y/o metadatos a una ubicación donde los decodificadores 707, 708 de columna tienen acceso a la información contenida dentro de las muestras de imagen de referencia y/o metadatos que son relevantes. En algunas realizaciones, el o los decodificadores 707, 708 de columna pueden estar provistos de memoria de imagen de referencia asociada, en la que puede no ser necesaria la realimentación de las muestras de imagen de referencia y/o metadatos. En otras realizaciones, el combinador 715 puede estar provisto de un trayecto de datos inversa que permite que los datos de imagen de referencia del generador 717 de muestras basado en imágenes sean realimentados al decodificador o decodificadores 707, 708 de columna. En la FIG. 7, dicho trayecto de realimentación se indica mediante las puntas de doble extremo de las flechas 713, 714, 716 y 719. En aún otras realizaciones, las imágenes de referencia pueden almacenarse en la unidad 710 de compartición (en cuyo caso podría proporcionarse un trayecto de datos correspondiente entre el generador 717 de muestras basado en imágenes y la unidad 710 de compartición, pero no mostrarse en la FIG. 7 para mejorar la claridad).

La FIG. 8 es un diagrama de bloques que ilustra una implementación basada en un sistema 800 de procesamiento de datos (por ejemplo, un ordenador personal ("PC")) de acuerdo con una realización de la invención. Hasta este punto, por conveniencia, la descripción no se ha relacionado explícitamente con la posible implementación física del codificador y/o decodificador en detalle. Son posibles muchas implementaciones físicas diferentes basadas en combinaciones de software y/o componentes. Por ejemplo, en algunas realizaciones, el o los codificadores y/o decodificadores de video pueden implementarse usando circuitos integrados personalizados o de matriz de puerta, en muchos casos, por razones relacionadas con la eficiencia de costos y/o la eficiencia del consumo de energía.

Adicionalmente, las implementaciones basadas en software son posibles utilizando arquitecturas de procesamiento de propósito general, un ejemplo de las cuales es el sistema 800 de procesamiento de datos. Por ejemplo, usando un ordenador personal o dispositivo similar (por ejemplo, decodificador, ordenador portátil, dispositivo móvil), tal estrategia de implementación puede ser posible como se describe a continuación. Como se muestra en la FIG. 8, según las

realizaciones descritas, el codificador y/o el decodificador para un PC o dispositivo similar 800 se puede prever en forma de un medio 801 legible por ordenador (por ejemplo, CD-ROM, ROM de semiconductores, tarjeta de memoria) que contiene instrucciones configuradas para habilitar un procesador 802, solo o en combinación con el hardware 803 del acelerador (por ejemplo, procesador de gráficos), en combinación con la memoria 804 acoplada al procesador 802 y/o al hardware acelerador 803 para realizar la codificación o decodificación. El procesador 802, la memoria 804 y el hardware acelerador 803 pueden estar acoplados a un bus 805 que puede usarse para entregar el flujo de bits y el video sin comprimir a/desde los dispositivos antes mencionados. Dependiendo de la aplicación, los periféricos para la entrada/salida del flujo de bits o el video sin comprimir pueden ser acoplados al bus 805. Por ejemplo, se puede conectar una cámara 806 a través de una interfaz adecuada, como un capturador de fotogramas 807 o un enlace USB 808, al bus 805 para la entrada en tiempo real de video sin comprimir. Se puede usar una interfaz similar para dispositivos de almacenamiento de video sin comprimir, tales como VTR. El video sin comprimir puede ser emitido a través de un dispositivo de visualización, tal como un monitor de ordenador o una pantalla 809 de TV. Se puede usar una unidad de DVD RW o equivalente (por ejemplo, CD ROM, CD-RW Blue Ray, tarjeta de memoria) 810 para introducir y/o emitir el flujo de bits. Finalmente, para la transmisión en tiempo real a través de una red 812, se puede usar una interfaz 811 de red para transmitir el flujo de bits y/o video sin comprimir, dependiendo de la capacidad del enlace de acceso a la red 812, y de la propia red 812.

De acuerdo con diversas realizaciones, el o los métodos descritos anteriormente pueden implementarse mediante un módulo de software respectivo. Según otras realizaciones, el o los métodos descritos anteriormente pueden implementarse mediante un módulo de hardware respectivo. Según aún otras realizaciones, el o los métodos descritos anteriormente pueden implementarse mediante una combinación de módulos de software y hardware.

Mientras que las realizaciones se han descrito, por conveniencia, principalmente con referencia a un método ejemplar, el aparato tratado anteriormente con referencia a un sistema 800 de procesamiento de datos puede, de acuerdo con las realizaciones descritas, ser programado para permitir la práctica del método o métodos descritos. Además, un artículo de fabricación para su uso con un sistema 800 de procesamiento de datos, tal como un dispositivo de almacenamiento pregrabado u otro medio o producto similar legible por ordenador, incluidas las instrucciones del programa grabadas en el mismo, puede dirigir el sistema 800 de procesamiento de datos para facilitar la práctica del método o métodos descritos. Se entiende que tales aparatos y artículos de fabricación, además del método o métodos descritos, caen todos dentro del alcance de las realizaciones descritas.

En particular, las secuencias de instrucciones que cuando se ejecutan hacen que el método descrito en la presente memoria sea realizado por el sistema 800 de procesamiento de datos, puede estar contenidas en un producto portador de datos de acuerdo con una realización de la invención. Este producto de soporte de datos puede ser cargado y ejecutado mediante el sistema 800 de procesamiento de datos. Además, las secuencias de instrucciones que, cuando se ejecutan, hacen que el método descrito en la presente memoria sea ejecutado por el sistema 800 de procesamiento de datos puede estar contenido en un programa informático o producto de software según una realización de la invención. Este programa informático o producto de software puede ser cargado y ejecutado mediante el sistema 800 de procesamiento de datos. Además, las secuencias de instrucciones que, cuando se ejecutan, hacen que el método descrito en la presente memoria sea realizado por el sistema 800 de procesamiento de datos pueden estar contenidas en un producto de circuito integrado (por ejemplo, un módulo o módulos de hardware) que puede incluir un coprocesador o memoria según una realización de la invención. Este producto de circuito integrado se puede instalar en el sistema 800 de procesamiento de datos.

**REIVINDICACIONES**

1. Un método para codificar una imagen (200, 300) de video que comprende una pluralidad de bloques (1... 24) de árbol codificados dispuestos en al menos dos filas y una pluralidad de columnas (202, 203, 302, 303) de bloques de árbol codificados, comprendiendo el método:
- 5     codificar uno o más elementos de sintaxis que definen un ancho de columna correspondiente de cada columna, estando el uno o más elementos de sintaxis colocados en una o más estructuras de sintaxis de alto nivel, siendo cada ancho de columna mayor que un bloque de árbol codificado y siendo una suma de todas las columnas igual a la dimensión horizontal de la imagen (200, 300) de video, y
- 10    codificar al menos un sector de la imagen de video, en donde el al menos un sector comprende al menos dos bloques de árbol codificados ubicados en al menos dos filas de bloques de árbol codificados y abarca bloques de árbol codificados que están ubicados en más de una de las columnas.
- 15    caracterizado por que el método comprende además codificar la pluralidad de bloques de árbol codificados siguiendo un orden de escaneo por columna de bloques de árbol codificados, según el cual la pluralidad de bloques de árbol codificados son codificados secuencialmente por columna de izquierda a derecha a través de una pluralidad de columnas, y secuencialmente por bloque de árbol codificado de izquierda a derecha y de arriba hacia abajo dentro de cada una de la pluralidad de columnas de tal modo que los bloques de árbol codificados de una primera columna (202) en el orden de escaneo por columna son codificados en primer lugar en un orden de escaneo local por columna de izquierda a derecha, de arriba hacia abajo a través de la primera columna (202) antes de que los bloques de árbol codificados de una segunda columna (202) en el orden de escaneo por columna sean codificados.
- 20    2. El método de la reivindicación 1, que comprende además romper al menos una forma de predicción o de filtrado en bucle en los límites de columna entre columnas adyacentes de bloques de árbol codificados.
3. El método de la reivindicación 2, en donde romper al menos una forma de predicción o de filtrado en bucle en los límites de columna comprende romper un subconjunto de la predicción o del filtrado en bucle en los límites de columna.
- 25    4. El método de la reivindicación 1, que comprende además codificar al menos una de varias columnas y un ancho de una columna en un conjunto (207, 310) de parámetros.
5. El método de la reivindicación 1, en donde el ancho de columna correspondiente de las columnas es variable entre al menos dos filas de bloques de árbol codificados en la imagen de video.
6. El método de la reivindicación 5, en donde la variabilidad se controla codificando el ancho de columna correspondiente en un encabezado (209, 211, 312, 314, 316) de sector.
- 30    7. El método de la reivindicación 1, en donde el ancho de cualquier columna es un múltiplo entero de un tamaño de un bloque de árbol codificado o un múltiplo entero del tamaño del bloque de árbol codificado más pequeño utilizado dentro de la imagen.
- 35    8. Un método para codificar una imagen (200, 300) de video que comprende una pluralidad de bloques (1 ... 24) de árbol codificados dispuestos en al menos dos filas y una pluralidad de columnas (202, 203, 302, 303) de bloques de árbol codificados, comprendiendo el método:
- obtener, a partir de uno o más elementos de sintaxis colocados en una o más estructuras de sintaxis de alto nivel en la imagen de video codificada, al menos uno de entre
- un ancho de columna correspondiente de cada columna y
- varias columnas;
- 40    si no se obtiene a partir del uno o más elementos de sintaxis colocados en una o más estructuras de sintaxis de alto nivel en la imagen de video codificada, derivar el ancho de columna correspondiente de cada columna, en donde las columnas tienen un ancho mayor que un bloque de árbol codificado; y
- 45    decodificar al menos un sector de la imagen de video codificada, en donde el al menos un sector comprende al menos dos bloques de árbol codificados ubicados en al menos dos filas de bloques de árbol codificados y abarca bloques de árbol codificados que están ubicados en más de una de las columnas,
- 50    comprendiendo el método además decodificar la pluralidad de bloques de árbol codificados siguiendo un orden de escaneo de bloques de árbol codificados, según el cual la pluralidad de bloques de árbol codificados son decodificados secuencialmente por columna de izquierda a derecha a través de una pluralidad de columnas, y secuencialmente por bloque de árbol codificado de izquierda a derecha y de arriba hacia abajo dentro de cada una de la pluralidad de columnas de tal modo que los bloques de árbol codificados de una primera columna (202) en el orden de escaneo por columna son decodificados en primer lugar en un orden de escaneo local por columna de izquierda a derecha, de arriba hacia abajo a través de toda la primera columna (202) antes de que los bloques de árbol codificados de una segunda

columna (202) en el orden de escaneo por columna sean decodificados.

9. El método de la reivindicación 8, que comprende además romper al menos una forma de predicción o de filtrado en bucle en los límites de columna entre columnas adyacentes de bloques de árbol codificados.

5 10. El método de la reivindicación 9, en donde romper la al menos una forma de predicción o de filtrado en bucle en los límites de columna comprende romper un subconjunto de la predicción o del filtrado en bucle en los límites de columna.

11. El método de la reivindicación 8, en donde al menos uno del ancho de columna correspondiente de la columna y el número de columnas se obtienen a partir de un conjunto (207, 310) de parámetros.

12. El método de la reivindicación 8, en donde el ancho de columna correspondiente de cada columna es variable entre al menos dos filas de bloques de árbol codificados en la imagen de video codificada.

10 13. El método de la reivindicación 12, en donde la variabilidad se controla codificando un ancho de columna en un encabezado (209, 211, 312, 314, 316) de sector.

14. El método de la reivindicación 8, en donde el ancho de cualquier columna es un múltiplo entero de un tamaño de un bloque de árbol codificado o un múltiplo entero del tamaño del bloque de árbol codificado más pequeño utilizado dentro de la imagen.

15

FIG. 1 100

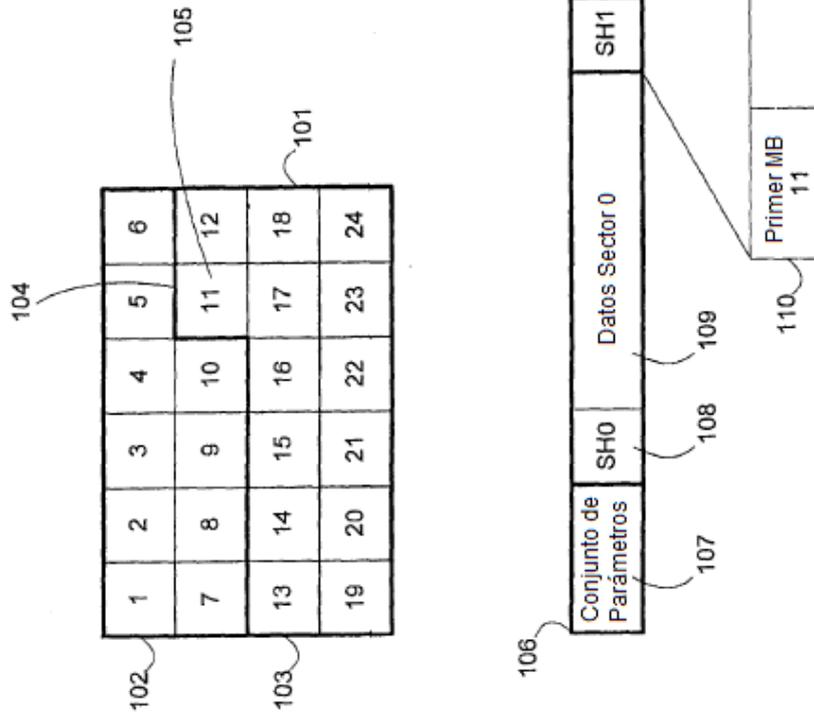


FIG. 2 200

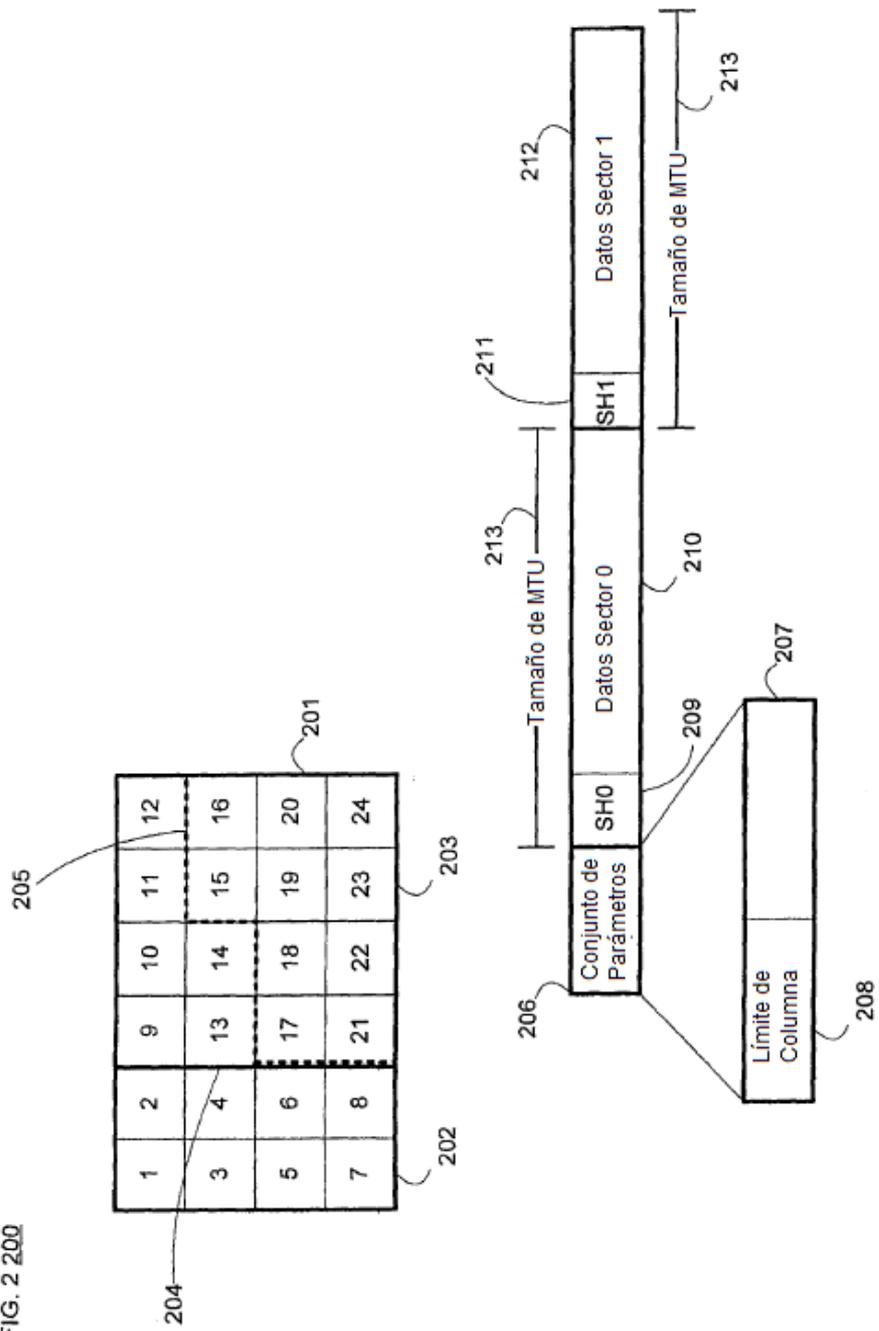


FIG. 3 300

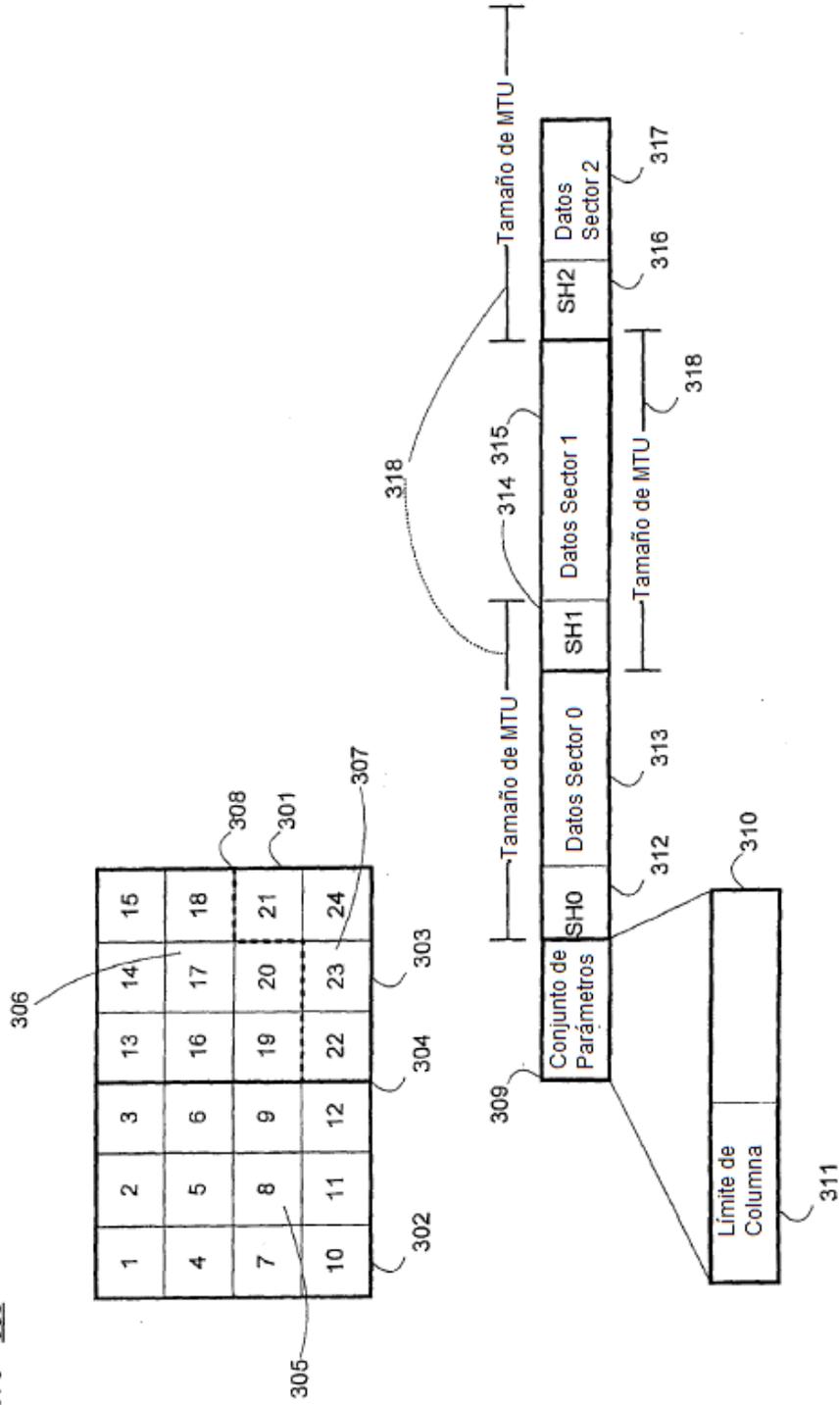


FIG. 4 400

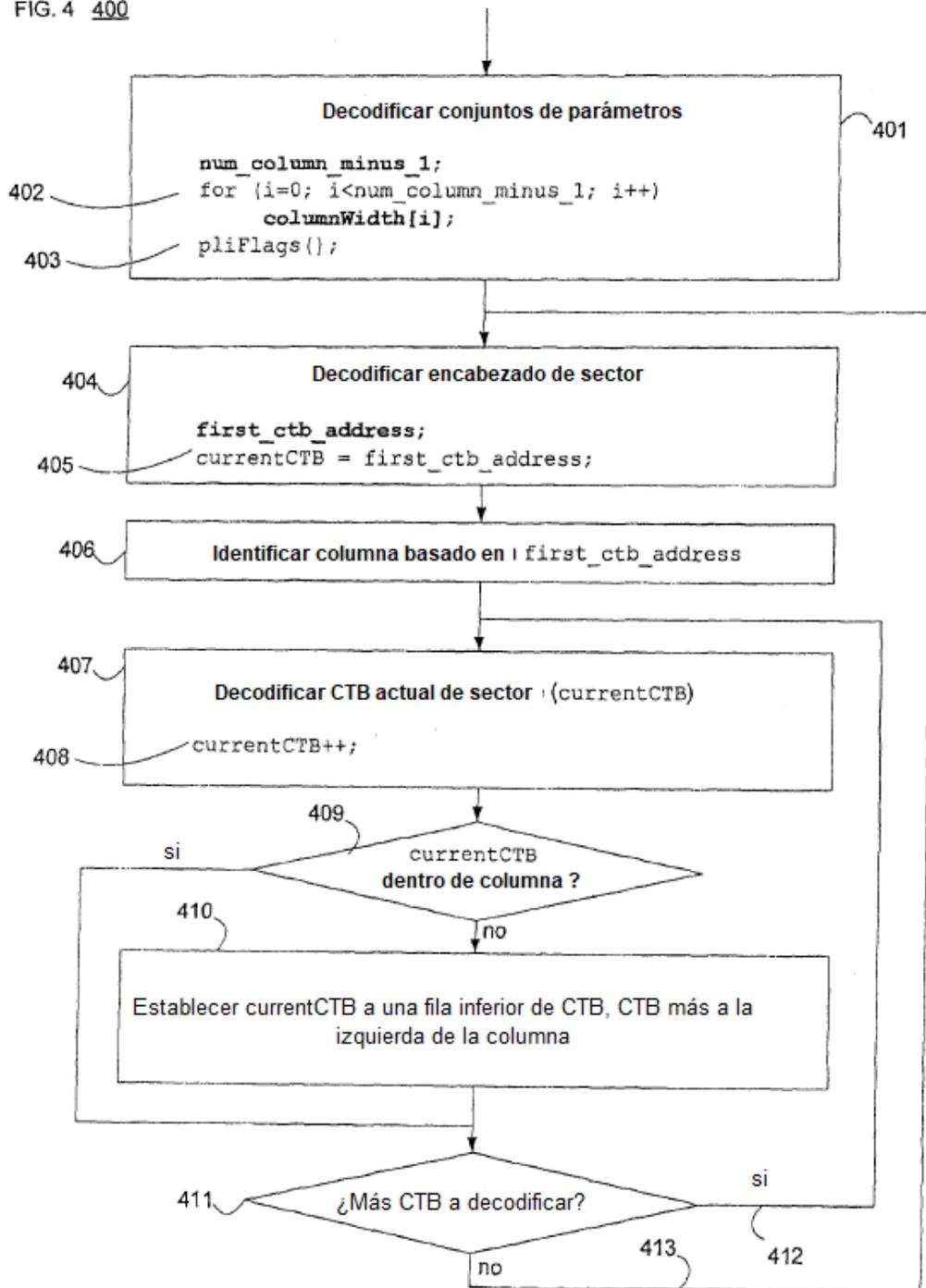


FIG. 5 500

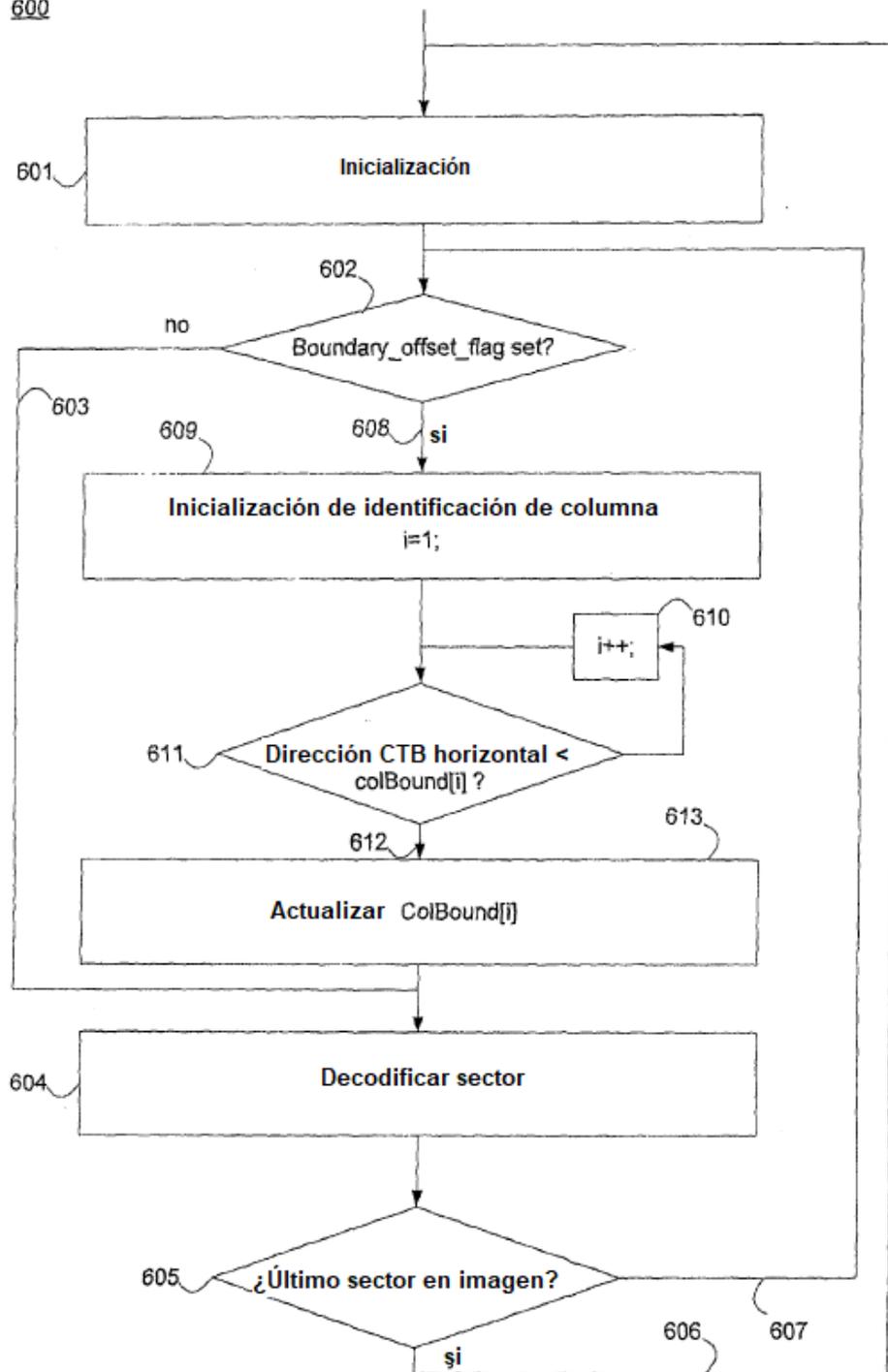
The figure shows a 5x10 grid of numbers. The numbers are arranged in rows and columns as follows:

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33						
40	41								
50	51								

Labels and dashed lines indicate specific regions:

- 501: A dashed line above the first three columns (0-2).
- 502: A dashed line above the last three columns (6-8).
- 503: A bracket above the first column (0).
- 504: A bracket to the left of the first row (0-9).
- 505: A bracket to the left of the last row (40-51).

FIG. 6 600



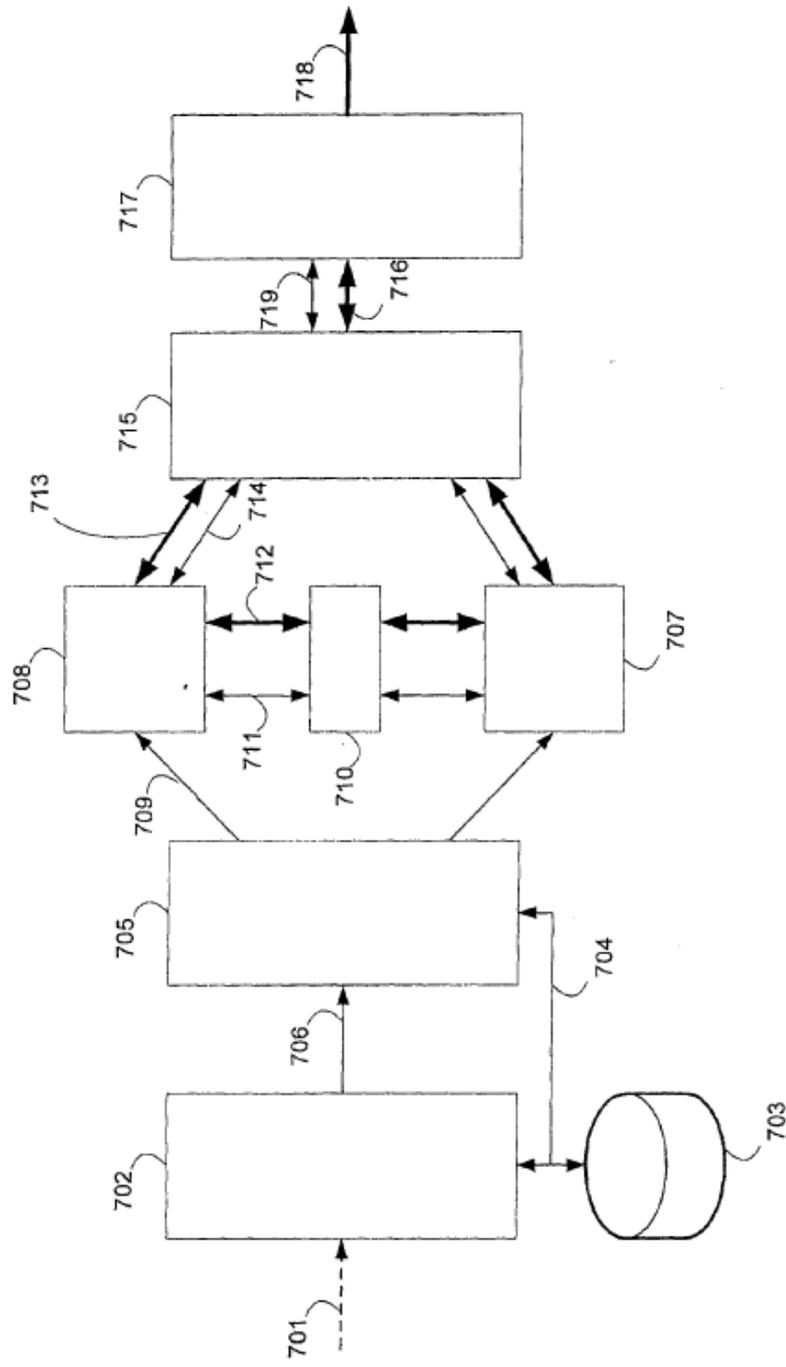


FIG. 7 700

