(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 627 741**

(51) Int. CI.:

H04N 19/70 (2014.01)
H04N 19/44 (2014.01)
H04N 19/31 (2014.01)
H04N 19/423 (2014.01)
H04N 19/597 (2014.01)
H04N 19/503 (2014.01)
H04N 19/152 (2014.01)
H04N 19/587 (2014.01)
H04N 19/85 (2014.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **26.09.2013 E 16169447 (6)**

(97) Fecha y número de publicación de la concesión europea: **01.03.2017 EP 3076673**

(54) Título: **Decodificación y codificación de imágenes de una secuencia de video**

(30) Prioridad:

28.09.2012 US 201261706869 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
31.07.2017

(73) Titular/es:

TELEFONAKTIEBOLAGET LM ERICSSON (PUBL) (100.0%)
164 83 Stockholm, SE

(72) Inventor/es:

SJÖBERG, RICKARD y
SAMUELSSON, JONATAN

(74) Agente/Representante:

ELZABURU SLP, .

DESCRIPCIÓN

Decodificación y codificación de imágenes de una secuencia de video

Campo técnico

Las realizaciones presentes se refieren, en general, a la decodificación y codificación de imágenes de una secuencia de video y, en particular, a la generación como salida o transferencia rápida de imágenes desde una memoria intermedia de imágenes decodificadas en relación con la decodificación y codificación de imágenes.

Antecedentes

Compresión de video H.264

La norma H.264 (o Moving Picture Experts Group-4 Advanced Video Coding (MPEG-4 AVC)) es la norma de codificación de video del estado de la técnica. Consiste en un esquema de codificación de video híbrido basado en bloques, que se aprovecha de redundancias temporales y espaciales. La norma H.264/AVC se define en un texto de especificación que contiene muchos procesos de decodificación que se tienen que ejecutar en la secuencia especificada para que un decodificador cumpla con la norma. No se establecen requisitos sobre el codificador, pero es frecuente el caso en el que el codificador también ejecuta la mayor parte de los procesos al objeto de obtener una buena eficiencia de compresión.

La norma H.264/AVC define una memoria intermedia de imágenes decodificadas (DPB, decoded picture buffer, por sus siglas en inglés) que almacena las imágenes decodificadas después de que se hayan decodificado. Esto significa que se requiere que el decodificador utilice una cantidad de memoria definida al objeto de decodificar una secuencia. La DPB contiene imágenes que se utilizan como referencia durante la decodificación de imágenes futuras. “Utilizada como referencia” significa en la presente memoria que una imagen particular se utiliza para hacer una predicción cuando se decodifica otra imagen. Los valores de los píxeles de la imagen que se utiliza como referencia se pueden utilizar de esta forma para predecir los valores de los píxeles de la imagen que se está decodificando en ese momento. A esto también se hace referencia como predicción intermedia. La DPB contiene además imágenes que están esperando para ser generadas como salida. “Generada como salida” significa en la presente memoria la función por la que un decodificador da salida a una imagen a la parte exterior del decodificador. La especificación H.264 describe la forma según la cual una secuencia de bits se convierte en unas imágenes decodificadas que se generan como salida a continuación, véase la figura 1. Las imágenes generadas como salida se pueden, por ejemplo, visualizar o escribir a disco.

Una causa común por la que una imagen de la DPB espera para ser generada como salida es que hay una imagen, que no se ha decodificado todavía, que será generada como salida antes que la propia imagen.

La figura 2 muestra un ejemplo de tres imágenes: A, B y C. El orden de decodificación es el orden según el cual las imágenes en formato comprimido se suministran al decodificador. Éste normalmente es el mismo que el orden según el cual las imágenes se codifican por medio del codificador. La figura 2 muestra que el orden de decodificación en este ejemplo es A, B y C. El orden de salida es el orden según el cual las imágenes decodificadas se generan como salida. El orden de salida no tiene que ser el mismo que el orden de decodificación, tal y como se ilustra en el ejemplo de la figura 2, en el que el orden de salida es A, C, B. Las flechas de la figura muestran qué imágenes se utilizan como referencia para cada imagen: la imagen A se utiliza como referencia para ambas imágenes B y C.

En la figura 2, la imagen C se decodifica después de la B, pero se genera como salida antes que ella. Cuando la imagen B se ha decodificado, no se puede generar como salida de forma inmediata, ya que la imagen C no se ha decodificado todavía y tiene que ser generada como salida antes que la imagen B. Por tanto, la imagen B se tiene que almacenar en la DPB después de que se haya decodificado, incluso aunque no se utilice como referencia por parte de ninguna otra imagen. Cuando se decodifica la imagen C, la imagen A debe estar presente también en la DPB, debido a que la imagen C utiliza como referencia la imagen A.

El orden de salida se regula por medio de la indicación de un valor PictureOrderCount (POC). Hay elementos de sintaxis en la secuencia de bits para la transmisión del POC de todas las imágenes, y estos valores se utilizan al objeto de definir el orden de salida de las imágenes.

Para hacer un seguimiento de la DPB, la norma H.264/AVC contiene tres procesos que tienen lugar después de que se haya decodificado una imagen: el proceso de marcado de imagen, el proceso de salida de imagen y el proceso de liberación.

El proceso de marcado de imagen marca las imágenes como “utilizada como referencia” o como “no utilizada como referencia”. Una imagen marcada como “utilizada como referencia” está disponible como referencia, lo cual significa que una imagen siguiente en el orden de decodificación puede utilizar como referencia la imagen en sus procesos de decodificación. Una imagen marcada como “no utilizada como referencia” no se puede utilizar como referencia por parte de imágenes siguientes. Este proceso se controla por medio del codificador a través de la secuencia de

bits. Hay una sintaxis opcional en la secuencia de bits de la norma H.264/AVC que, cuando está presente, indica qué imágenes se han de marcar como “no utilizadas como referencia”. A esta operación se hace referencia a menudo como operación de control de la gestión de memoria (MMCO, memory management control operation, por sus siglas en inglés). Si no hay una sintaxis MMCO opcional, se define un mecanismo de primero en entrar/primero en salir, denominado proceso de “ventana móvil”. El proceso de ventana móvil significa que, cuando la última imagen que se decodifica fuera a dar lugar a demasiadas imágenes en la DPB, la imagen más antigua en el orden de decodificación se marca automáticamente como “no utilizada como referencia”.

El proceso de salida de imagen, que se realiza después del proceso de marcado de imagen, marca las imágenes como “necesaria para la salida” o como “no necesaria para la salida”. Una imagen marcada como “necesaria para la salida” no se ha generado como salida todavía, mientras que una imagen marcada como “no necesaria para la salida” se ha generado como salida y ha dejado de esperar a ser generada como salida. El proceso de salida de imagen además genera como salida las imágenes. Esto significa que el proceso selecciona las imágenes que están marcadas como “necesarias para la salida”, las genera como salida y a continuación las marca como “no necesarias para la salida”. El proceso de salida de imagen determina el orden según el cual se generan como salida las imágenes. Se ha de observar que el proceso de salida de imagen puede generar como salida y marcar cero, una o muchas imágenes, después de que se haya decodificado una imagen particular.

Después de que se han invocado estos dos procesos por medio del decodificador, se invoca el proceso de liberación. Las imágenes que están marcadas como “no utilizadas como referencia” y “no necesarias para la salida” se vacían y eliminan de la DPB. A veces se hace referencia a esto diciendo que se ha liberado uno de los espacios de memoria de imagen de la DPB.

El tamaño de la DPB en la norma H.264/AVC está limitado. Esto significa que está limitado el número de imágenes que se pueden almacenar debido a que están esperando a ser generadas como salida o a que están disponibles como referencia. La variable *max_dec_frame_buffering* indica el tamaño de la DPB, al cual se hace referencia a veces como el número de espacios de memoria de imagen que hay en la DPB. El codificador tiene que asegurar que nunca se excede el tamaño de la DPB.

Los tres procesos se describen en la norma. Esto significa que el decodificador está controlado por el codificador y que, por tanto, el decodificador no tiene ninguna libertad en lo que respecta al orden de salida. Está todo determinado por el proceso de salida de imagen y los elementos relacionados de la secuencia de bits enviada por el codificador. En la figura 3 se muestra un diagrama de flujo simplificado de las etapas de decodificación de la norma H.264/AVC.

El proceso de salida de imagen de la norma H.264 define el orden según el cual las imágenes se generarán como salida. Un decodificador que genera como salida las imágenes en el orden correcto es uno que cumple que con el orden de salida. Un decodificador puede ajustarse al proceso de salida de imagen descrito en la norma H.264, pero a veces es posible utilizar la variable *num_reorder_frames* para generar como salida las imágenes con anterioridad a lo determinado por el proceso de salida de imagen. *num_reorder_frames* indica el máximo número de imágenes que preceden a cualquier imagen en el orden de decodificación y que van detrás de ella en el orden de salida.

La figura 4 muestra un ejemplo en el que la imagen B se acaba de decodificar. Sin embargo, la imagen B no se puede generar como salida debido a que no se conoce si la imagen C se ha de generar como salida antes o después de la imagen B. Si el codificador ha decidido que el orden de salida es el mismo que el orden de decodificación, puede indicar un valor de 0 en *num_reorder_frames* al decodificador. El codificador ha asegurado de este modo que la imagen C del ejemplo se generará como salida después de la imagen B, y un decodificador puede generar como salida la imagen B de forma inmediata en el momento en que se haya decodificado. En este caso, cuando *num_reorder_frames* es 0, no hay ningún retardo de reordenación adicional en el decodificador. Si *num_reorder_frames* se fija a 1 en el ejemplo, es posible que la imagen C se genere como salida antes que la imagen B. Con *num_reorder_frames* igual a 1, hay un retardo de reordenación adicional de 1 imagen, con *num_reorder_frames* igual a 2, el retardo de reordenación es de 2 imágenes, y así sucesivamente.

Compresión de video HEVC

La norma de codificación de video de alta eficiencia (HEVC, High Efficiency Video Coding, por sus siglas en inglés), a la que también se hace referencia como norma H.265, es una norma de codificación de video desarrollada en el Joint Collaborative Team – Video Coding (JCT-VC). JCT-VC es un proyecto de colaboración entre la MPEG y la Sección de Normalización de las Telecomunicaciones de la Unión Internacional de Telecomunicaciones (ITU-T, International Telegraph Union Telecommunication, por sus siglas en inglés). La norma HEVC incluye una serie de nuevas herramientas y es considerablemente más eficiente que la norma H.264/AVC. La norma HEVC define además un *temporal_id* para cada imagen, que se corresponde con la capa temporal a la que pertenece la imagen. Las capas temporales están ordenadas y tienen la propiedad de que una capa temporal inferior nunca depende de una capa temporal más alta. Por tanto, las capas temporales más altas se pueden eliminar sin afectar a las capas temporales inferiores. La eliminación de las capas temporales se puede denominar como un escalado temporal. Una secuencia de bits HEVC contiene un elemento de sintaxis, *max_sub_layers_minus*1, que especifica el máximo número de capas temporales que pueden estar presentes en la secuencia de bits. Un decodificador puede

decodificar todas las capas temporales o únicamente decodificar un subconjunto de capas temporales. A la capa temporal más alta que realmente decodifica el decodificador se hace referencia como la sub-capa temporal más alta, y se puede fijar igual o menor que los máximos números de capas que se especifican por medio de *max_sub_layers_minus*1. Por lo tanto, el decodificador decodifica todas las capas que son iguales o menores que la sub-capa temporal más alta. La sub-capa temporal más alta se puede fijar a través de medios externos.

Se ha de observar que la descripción anterior no es específica de las capas temporales, sino que también es válida para otros tipos de capas, tales como capas espaciales y capas de calidad, etc. A la capa temporal que decodifica en consecuencia el decodificador se hace referencia como la capa decodificada más alta.

El flujo de decodificación de la norma HEVC es ligeramente diferente del de la norma H.264/AVC. La norma HEVC tiene una DPB, un proceso de marcado de imagen que marca las imágenes como “utilizada como referencia” o como “no utilizada como referencia”, un proceso de salida de imagen que marca las imágenes como “necesaria para la salida” o como “no necesaria para la salida” y un proceso de liberación. Al igual que la norma H.264/AVC, la norma HEVC utiliza también valores POC para definir el orden de salida de las imágenes. Un valor POC se representa en la norma HEVC por medio de la variable PicOrderCntVal, en la que las imágenes se generan como salida según un orden creciente de PicOrderCntVal.

Sin embargo, la norma HEVC no tiene MMCO ni proceso de ventana móvil. En su lugar, la norma HEVC especifica que en cada cabecera de segmento se envía de forma explícita una lista de las imágenes que están marcadas como “utilizada como referencia”. El marcado de imagen en la norma HEVC utiliza esta lista y asegura que todas las imágenes de la DPB que figuran en la lista se marcan como “utilizada como referencia” y que todas las imágenes de la DPB que no figuran en la lista se marcan como “no utilizada como referencia”. Esta lista se denomina el conjunto de imágenes de referencia (RPS, reference picture set, por sus siglas en inglés), y el envío de uno en cada cabecera de segmento significa que el estado del marcado de referencia en la DPB es explícito y se repite en cada segmento, lo cual no ocurre en la norma H.264/AVC.

Debido a que en la norma HEVC se utilizan RPSs, el proceso de marcado de imagen, el proceso de salida de imagen y el proceso de liberación se realizan todos después del análisis de la primera cabecera de segmento de una imagen, véase la figura 5.

La funcionalidad *num_reorder_frames*, tal y como se ha descrito para la norma H.264/AVC, también está presente en la norma HEVC. Una secuencia de bits HEVC contiene un elemento de sintaxis para cada capa temporal, denotado como *max_num_reorder_pics*[i], en donde i es la capa temporal. La función de *max_num_reorder_pics*[i] es la misma que la de *num_reorder_frames*, pero aquí cada clave indica el máximo número de imágenes permitido en la misma capa temporal, o en una inferior, que preceden a una imagen en el orden de decodificación y que van detrás de esa imagen en el orden de salida.

Considérese el ejemplo de la figura 6, en el que el orden de decodificación es A, B, C, D, E, y el orden de salida es A, D, C, E, B. Ésta es una estructura de imágenes que utiliza capas temporales, en donde las imágenes A y B pertenecen a la capa temporal más baja (capa 0), la imagen C pertenece a una capa temporal media (capa 1), y las imágenes D y E pertenecen a la capa temporal más alta (capa 2). Las flechas en la figura muestran qué imágenes se utilizan como referencia por parte de otras imágenes. Por ejemplo, la imagen A se utiliza como referencia por parte de la imagen B, ya que hay una flecha que va desde la imagen A hasta la imagen B. La mejor forma de utilizar *max_num_reorder_pics* en la norma HEVC es fijarlo tan bajo como sea posible al objeto de reducir el retardo de salida tanto como sea posible. En la figura 6 se muestran los valores más bajos posibles de *max_num_reorder_pics* para cada capa temporal. La razón por la que es 0 para la capa más baja es porque no hay ninguna imagen en la capa 0 que preceda a ninguna imagen en el orden de decodificación, ni que vaya detrás de ella en el orden de salida. Para la capa 1, tenemos la imagen B que precede a la imagen C en el orden de decodificación, pero que va detrás de ella en el orden de salida, y para la capa 2, tenemos que las imágenes B y C preceden, ambas, a la imagen D en el orden de decodificación, pero que van detrás de ella en el orden de salida.

Si un decodificador supiera que únicamente decodificará la capa temporal 0, podría potencialmente generar como salida la imagen B tan pronto como se haya decodificado, pero si el decodificador decodifica todas las capas no lo puede hacer. En ese caso, tendría que esperar hasta que hubiera dos imágenes decodificadas que vayan detrás de B en el orden de salida.

El documento JCTVC-K0030_v3, “Proposed editorial improvement for high efficiency video coding (HEVC) text specification draft 8”, de B. Bross et al., 11ª conferencia de JCT-VC de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, Shanghái, 10-19 de octubre de 2012, publicado el 12 de septiembre de 2012, analiza la utilización de *max_num_reorder_pics* en la sección 7.4.2.1 en la página 62, y en la sección 7.4.2.2 en la página 64.

no_output_of_prior_pics_flag

Tanto la secuencia de bits de la norma H.264 como la de la norma HEVC especifican un indicador denominado *no_output_of_prior_pics_flag*. Este indicador está presente en la cabecera de segmento de las imágenes de acceso aleatorio (RAP, random access picture, por sus siglas en inglés). Las imágenes de acceso aleatorio son imágenes a partir de las cuales es posible ajustar una emisión en continuo. Garantizan que la decodificación de las imágenes

futuras se puede realizar de forma correcta si un decodificador comienza a decodificar desde el punto de acceso aleatorio. No se tiene que suministrar al decodificador ningún dato que contenga las imágenes que preceden a la imagen de acceso aleatorio en el orden de decodificación para que el ajuste funcione.

El indicador *no_output_of_prior_pics_flag* especifica cómo se tratan en la memoria intermedia de imágenes decodificadas las imágenes previamente decodificadas, después de la decodificación de una imagen de acceso aleatorio. De forma resumida, si *no_output_of_prior_pics_flag* es igual a 1, no se ha de generar como salida ninguna imagen de la DPB que esté marcada como “necesaria para la salida”, pero si *no_output_of_prior_pics_flag* es igual a 0, se deben generar como salida.

Considérese la figura 7, que muestra un ejemplo en el que *max_num_reorder_pics* es 0 y la imagen C es una imagen de acceso aleatorio con *no_output_of_prior_pics_flag* igual a 1. En la norma H.264, sería posible generar como salida la imagen B inmediatamente después de que se hubiera decodificado. Esto no ocurre en la especificación actual de la norma HEVC, ya que el decodificador no sabe inmediatamente después de que se ha decodificado la imagen B, si la imagen C es una imagen RAP con *no_output_of_prior_pics_flag* igual a 1 o no. Si la imagen C no es una de esas imágenes, la imagen B se podría generar como salida inmediatamente después de que se hubiera decodificado. Pero si la imagen C es realmente una imagen RAP con *no_output_of_prior_pics_flag* igual a 1, la imagen B no se debería generar como salida, debido a que la imagen B está marcada como “necesaria para la salida” cuando se decodifica la cabecera de segmento de la imagen C.

Debido a que en la norma HEVC el proceso de salida de imagen se realiza cuando se analiza la cabecera de segmento y a que *no_output_of_prior_pics_flag* es una característica importante, hay un retardo de salida mayor en la norma HEVC actual que en la norma H.264/AVC.

La información sobre la utilización de *no_output_of_prior_pics_flag* está descrita en la sección 7.4.7.1 en la página 75, y en la sección C.5.2 en la página 26 del documento JCTVC-K0030_v3.

La ventaja de la utilización de RPSs en la norma HEVC es que se tiene mucha más tolerancia al error en comparación con el método de la norma H.264/AVC. Además, la escalabilidad temporal es más sencilla. Un problema de la solución de la norma HEVC es que introduce un retardo adicional en la generación como salida de imágenes en comparación con la norma H.264/AVC. En la norma H.264/AVC, las imágenes se pueden generar como salida después de que se haya decodificado una imagen. En la norma HEVC, el decodificador tiene que esperar a que se analice la cabecera de segmento de la imagen siguiente antes de que las imágenes se generen como salida. Esto da lugar a un retardo. Por lo tanto, existe la necesidad de resolver las deficiencias de la técnica anterior de codificación de video, y en particular, los problemas de retardo que pueden surgir en la codificación de video de la técnica anterior.

En el documento JCTVC-F493 (sección 3.3) se propone aplicar siempre el proceso de “transferencia rápida” antes de decodificar una imagen con el fin de reducir los retardos de salida.

Compendio

Es un objetivo general la provisión de una decodificación y una codificación mejoradas de las imágenes de una secuencia de video.

Es un objetivo particular la provisión de una decodificación y una codificación que hagan posible obtener un retardo de salida pequeño para las imágenes.

Estos y otros objetivos se consiguen por medio de las realizaciones descritas en la presente memoria. La invención se define en las reivindicaciones adjuntas.

Breve descripción de los dibujos

Las realizaciones, junto con objetos y ventajas adicionales de las mismas, se pueden comprender de la mejor manera haciendo referencia a la siguiente descripción, considerada en combinación con los dibujos que se acompañan, en los que:

La figura 1 es una vista general esquemática de un decodificador de acuerdo a la norma H.264/AVC.

La figura 2 ilustra el orden de salida y de decodificación de un ejemplo de una secuencia de video multi-capa.

La figura 3 es un diagrama de flujo de decodificación simplificado de la norma H.264/AVC.

La figura 4 ilustra el orden de salida y de decodificación de un ejemplo de una secuencia de video multi-capa.

La figura 5 es un diagrama de flujo de decodificación simplificado de la norma HEVC.

La figura 6 ilustra el orden de salida y de decodificación y las capas temporales de un ejemplo de una secuencia de video multi-capa.

La figura 7 ilustra el orden de salida y de decodificación de un ejemplo de una secuencia de video.

La figura 8 es un diagrama de flujo de un método ejecutado por medio de un decodificador según una realización.

La figura 9 es un diagrama de flujo de etapas adicionales opcionales del método de la figura 8.

La figura 10 es un diagrama de flujo de una etapa adicional opcional del método de la figura 8.

La figura 11 es un diagrama de flujo de un método ejecutado por medio de un decodificador según otra realización.

La figura 12 es un diagrama de flujo de etapas adicionales opcionales del método de las figuras 9 u 11.

La figura 13 es un diagrama de flujo de una etapa adicional opcional del método las figuras 8 u 11.

La figura 14 es un diagrama de flujo de decodificación simplificado de una realización.

La figura 15 es un diagrama de flujo de decodificación simplificado de otra realización.

La figura 16 es un diagrama de bloques esquemático de un decodificador según una realización.

La figura 17 es un diagrama de bloques esquemático de un decodificador según otra realización.

La figura 18 es un diagrama de bloques esquemático de un decodificador según una realización adicional.

La figura 19 es un diagrama de bloques esquemático de un decodificador según otra realización más.

La figura 20 es un diagrama de bloques esquemático de un decodificador según una realización adicional.

La figura 21 es un diagrama de flujo de un método ejecutado por medio de un codificador según una realización.

La figura 22 es un diagrama de flujo de un método ejecutado por medio de un codificador según otra realización.

La figura 23 es un diagrama de bloques esquemático de un codificador según una realización.

La figura 24 es un diagrama de bloques esquemático de un codificador según otra realización.

La figura 25 es un diagrama de bloques esquemático de un codificador según una realización adicional.

La figura 26 es un diagrama de bloques esquemático de un codificador según otra realización más.

La figura 27 es un diagrama de bloques esquemático de un codificador según una realización adicional.

La figura 28 es un diagrama de bloques esquemático de un terminal móvil según una realización.

La figura 29 es un diagrama de bloques esquemático de un nodo de red según una realización; y

La figura 30 ilustra el orden de salida y de decodificación de un ejemplo de una secuencia de video multi-capa.

Descripción detallada

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

Las realizaciones presentes se refieren, en general, a la decodificación y codificación de imágenes de una secuencia de video y, en particular, a la generación como salida o transferencia rápida de imágenes desde una memoria intermedia de imágenes decodificadas en relación con la decodificación y codificación de imágenes. Estas realizaciones permiten de esta forma que se reduzca el retardo de salida y hacen posible una generación de imágenes más rápida que en las soluciones del estado de la técnica.

En una realización general, cuando se codifica una imagen, el decodificador o el codificador (la decodificación de imágenes también se lleva a cabo durante la codificación al objeto de obtener imágenes de referencia reconstruidas a partir de imágenes previamente codificadas) determina el número de imágenes de una memoria intermedia de imágenes decodificadas (DPB) que se marcan como necesarias para la salida, y compara ese número con un valor que se obtiene de los elementos de sintaxis de la secuencia de bits. Si el número de imágenes de la DPB que están marcadas como necesarias para la salida es mayor que el valor que se obtiene de los elementos de sintaxis de la secuencia de bits, se lleva a cabo un proceso de salida modificado, o se genera como salida la imagen de las imágenes de la DPB marcadas como necesarias para la salida que es la primera imagen en el orden de salida, se señala el proceso de salida de imágenes como #2, y se marca como no necesaria para la salida.

A continuación, en la presente memoria se describirán en mayor detalle diferentes realizaciones y aspectos de implementaciones particulares, comenzando con la parte de decodificación y continuando seguidamente con la parte de codificación.

En la presente memoria, que una imagen se marque como necesaria para la salida implica que una imagen está esperando a ser generada como salida, por ejemplo, para visualización o para almacenamiento. De forma correspondiente, en la presente memoria, que una imagen se marque como no necesaria para la salida implica que una imagen no está esperando a ser generada como salida, y que, por ejemplo, se puede haber generado ya como salida. De forma correspondiente, que una imagen se marque como utilizada como referencia implica que una imagen se utiliza para hacer una predicción cuando se decodifica otra imagen. Que una imagen se marque como no utilizada como referencia implica que la imagen no se puede utilizar como referencia o predicción por parte de imágenes subsiguientes. El marcado, tal y como se ha mencionado con anterioridad, no se debe interpretar como un marcado literal de las imágenes con una etiqueta “necesaria para la salida”, “no necesaria para la salida”, “utilizada como referencia” o “no utilizada como referencia”. En su lugar, el marcado se debe interpretar en el sentido de que la imagen se identifica de alguna forma de manera que se debería / no se debería generar como salida, y de manera que se podría utilizar como imagen de referencia / no se podría utilizar como imagen de referencia. El marcado se puede implementar de diferentes formas. Por ejemplo, se podría almacenar la imagen en una parte específica de la DPB dedicada a las imágenes que se deben generar como salida o que no se deben generar como salida o a imágenes que se podrían utilizar como imagen de referencia o que no se podrían utilizar como imagen de referencia. De forma alternativa, la imagen se podría etiquetar o asociar de cualquier otra forma con un indicador o con otra clave que identifique la imagen como necesaria para la salida frente a no necesaria para la salida, y como utilizada como referencia frente a no utilizada como referencia. En el caso de que se implemente un indicador, el indicador podría tener un primer valor (0_{bin} o 1_{bin}) para indicar una de las alternativas, y podría tener por tanto un segundo valor (1_{bin} o 0_{bin}) para indicar la otra alternativa. Por ejemplo, se podría utilizar un primer indicador con respecto a la salida de imágenes, y se podría utilizar un segundo indicador con respecto a las referencias.

Por lo tanto, un aspecto general se refiere a un método llevado a cabo por medio de un decodificador. En el método, el decodificador determina el número de imágenes de la DPB que están marcadas como necesarias para la salida, y compara ese número con un valor que se obtiene de los elementos de sintaxis de la secuencia de bits. Si el número de imágenes de la DPB que están marcadas como necesarias para la salida es mayor que el valor que se obtiene de los elementos de sintaxis de la secuencia de bits, se lleva a cabo un proceso de salida modificado, o se genera como salida la imagen de entre las imágenes de la DPB marcadas como necesarias para la salida que es la primera imagen en el orden de salida, se señala el proceso de salida de imágenes como #2 en este caso, y se marca como no necesaria para la salida.

Este aspecto general se implementa preferiblemente en un decodificador que cumple con la norma HEVC, al que también se hace referencia en la presente memoria como decodificador que cumple con la norma H.265. En este caso, una realización de implementación se refiere a un método llevado a cabo por medio de un decodificador que cumple con la norma HEVC y que comprende las siguientes etapas. El decodificador que cumple con la norma HEVC determina un número de imágenes de una DPB que están marcadas como necesarias para la salida. El decodificador que cumple con la norma HEVC compara además el número con un valor que se obtiene de al menos un elemento de sintaxis presente en una secuencia de bits que representa las imágenes de una secuencia de video. El decodificador que cumple con la norma HEVC genera como salida a continuación una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor. Además, el decodificador que cumple con la norma HEVC marca la imagen generada como salida como no necesaria para la salida si el número es mayor que el valor.

La figura 8 es un diagrama de flujo de un método llevado a cabo por medio de un decodificador según una realización. El método comprende la determinación, en la etapa S1 y después de que se haya decodificado y almacenado en una DPB una imagen actual, de un número de imágenes de la DPB que están marcadas como necesarias para la salida. Una etapa siguiente S2 comprende la comparación de este número con un valor *sps_max_num_reorder_pics[HighestTid]*. Si el número es mayor que el valor, el método avanza a la etapa S3. Esta etapa S3 comprende la generación como salida de una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida. La imagen que se ha generado como salida en la etapa S3 se marca a continuación como no necesaria para la salida en la etapa S4. Por lo tanto, esta etapa S4 se ejecuta si el número es mayor que el valor.

En esta realización, se utiliza *sps_max_num_reorder_pics[HighestTid]* como representación preferida del valor que se obtiene de al menos un elemento de sintaxis presente en una secuencia de bits que representa las imágenes de una secuencia de video. Además, *HighestTid* especifica la capa más alta que se decodifica por medio del decodificador de la secuencia de video. Por tanto, *HighestTid* indica la sub-capa temporal más alta si la secuencia de video comprende una o más capas temporales, e indica la capa más alta decodificada si, por el contrario, la secuencia de video comprende otro tipo de capas, tales como una o más capas espaciales, capas de calidad, etc.

En una realización, *sps_max_num_reorder_pics*[i] indica el máximo número de imágenes permitido que puede preceder a una imagen en la secuencia de video codificada en el orden de decodificación y que van detrás de esa imagen en el orden de salida cuando *HighestTid* es igual a i. El prefijo *sps* de *max_num_reorder_pics*[i] indica que el elemento de sintaxis está presente preferiblemente en un conjunto de parámetros de secuencia (SPS, sequence parameter set, por sus siglas en inglés) de la secuencia de bits.

Por lo tanto, en esta realización se podría obtener un valor por capa en la secuencia de bits, y el valor que se habría de utilizar en la comparación que se ejecuta en la etapa S2 sería el valor que se obtuviera para la capa más alta que realmente se decodificara por medio del decodificador de la secuencia de video. Se ha de tener en cuenta que esta capa decodificada más alta puede ser igual o inferior al número máximo de capas de la secuencia de video que se especifica por medio de un elemento de sintaxis *max_sub_layers_minus*1, tal como *vps_max_sub_layers_minus*1 o *sps_max_sub_layers_minus*1, dependiendo de si el elemento de sintaxis se recupera de un conjunto de parámetros de video (VPS, video parameter set, por sus siglas en inglés) o de un SPS.

Por tanto, en una realización particular la secuencia de video es una secuencia de video multi-capa que comprende múltiples capas de imágenes, es decir, al menos dos. Cada capa de las múltiples capas tiene preferiblemente entonces un elemento de sintaxis asociado que define un valor respectivo. El valor utilizado en la comparación de la etapa S2 es por tanto, en esta realización, el valor que se obtiene del elemento de sintaxis asociado con la capa más alta que se decodifica por medio del decodificador de entre las múltiples capas.

En una realización particular, *sps_max_num_reorder_pics*[i] indica el máximo número de imágenes permitido en la misma capa, o en la inferior, como por ejemplo en la misma capa temporal o en la capa temporal inferior, en comparación con la capa i, que precede a una imagen en el orden de decodificación y que va detrás de esa imagen en el orden de salida.

En una realización, si el número no es mayor que el valor determinado en la comparación de la etapa S2, el método preferiblemente termina, y no se genera como salida ni se marca ninguna imagen. Por tanto, en tal caso, las etapas S3 y S4 se omiten y no se ejecutan, véase la línea de puntos a la derecha.

En una realización, las etapas S2–S4 sólo se pueden ejecutar una vez, después de que se haya decodificado y almacenado en la DPB una imagen actual. De forma alternativa, el bucle formado por las etapas S2–S4, véase la línea de puntos a la izquierda, se puede ejecutar hasta que el número de imágenes de la DPB que están marcadas como necesarias para la salida deje de ser mayor que el valor *sps_max_num_reorder_pics[HighestTid]*. Cada vez que se ejecuta el bucle de las etapas S2–S4, el número de imágenes de la DPB que están marcadas como necesarias para la salida se reduce en una unidad por medio del marcado como no necesaria para la salida, en la etapa S4, de la imagen generada como salida en la etapa S3. Esto significa que las etapas S3 o S4 se pueden ejecutar, después de que la imagen actual se haya decodificado y almacenado en la DPB, cero veces, si número≤*sps_max_num_reorder_pics[HighestTid]*; una vez, si número=*sps_max_num_reorder_pics[HighestTid]*+1; o más de una vez si número>*sps_max_num_reorder_pics[HighestTid]*+1; es decir, n veces si número=*sps_max_num_reorder_pics[HighestTid]*+n. Una vez que la comparación de la etapa S2 determina que el número ya no es mayor que el valor *sps_max_num_reorder_pics[HighestTid]*, el método termina.

La determinación del número de imágenes de la DPB marcadas como necesarias para la salida en la etapa S1 se lleva a cabo, preferiblemente, después de que la imagen actual se haya decodificado y almacenado en la DPB.

El método, tal y como se ha descrito con anterioridad y se ha ilustrado en la figura 8, se ejecuta preferiblemente una vez para cada una de las imágenes de la secuencia de video que se decodifica y almacena en la DPB. Por tanto, el método se ejecuta preferiblemente de forma instantánea, una vez que la imagen actual se considera decodificada, es decir, después de que se decodifica la última unidad de decodificación de la imagen, y de que la imagen actual decodificada se almacena en una memoria intermedia de almacenamiento de imágenes vacía, es decir, en espacio de memoria de imagen, de la DPB.

La figura 9 es un diagrama de flujo de etapas adicionales opcionales del método de la figura 8. El método comienza en la etapa S10, la cual comprende el análisis de una cabecera de segmento de la imagen actual de la secuencia de video que se ha de decodificar. Una etapa S11 siguiente determina un conjunto de imágenes de referencia (RPS) para la imagen actual a partir del análisis de la cabecera de segmento. Todas las imágenes de la DPB que no están presentes en el RPS se marcan como no utilizadas como referencia en la etapa S12. Una etapa S13 siguiente comprende la generación como salida de cero, una o muchas, es decir, más de una, imágenes, las cuales están marcadas como necesarias para la salida, de entre las imágenes de la DPB. Esta etapa S13 comprende también el marcado de las cero, una o muchas imágenes generadas como salida como no necesarias para la salida. La etapa S14 comprende el vaciado o eliminación, de la DPB, de toda imagen marcada como no utilizada como referencia y como no necesaria para la salida de entre las imágenes de la DPB. La imagen actual se decodifica a continuación en la etapa S15. El método avanza a continuación a la etapa S1 de la figura 8. Por lo tanto, en esta realización, la determinación del número de imágenes en la etapa S1, la comparación del número en la etapa S2, la generación como salida de la imagen en la etapa S3 y el marcado de la imagen en la etapa S4 se llevan a cabo después de la decodificación de la imagen actual en la etapa S15.

Además de la decodificación de la imagen actual en la etapa S15, esta etapa comprende preferiblemente además el marcado de la imagen actual como utilizada como referencia, o en una realización opcional, el marcado de la imagen actual como utilizada como referencia a corto plazo. La imagen actual se marca preferiblemente además como necesaria para la salida o no necesaria para la salida, de acuerdo al valor de la variable PicOutputFlag, obtenido preferiblemente en la etapa S10.

Por lo general, una secuencia de video codificada, es decir, una secuencia de bits, comprende unidades de capa de abstracción de red (NAL, network abstraction layer, por sus siglas en inglés). Básicamente, una unidad NAL comprende un segmento con una correspondiente cabecera de segmento que incluye información de control para ese segmento y la carga útil de datos de video, o bien la unidad NAL comprende un conjunto de parámetros, tales como un VPS, un SPS y un conjunto de parámetros de imagen (PPS, picture parameter set, por sus siglas en inglés). El conjunto de parámetros comprende información de control. Una imagen de la secuencia de video puede constar de un único segmento o de múltiples segmentos. La etapa S10 de la figura 9 comprende de esta forma el análisis de la parte de cabecera de segmento de la unidad NAL que comprende un segmento de la imagen. Si la imagen comprende múltiples segmentos y está por lo tanto distribuida entre múltiples unidades NAL que comprenden una respectiva cabecera de segmento, entonces la etapa S10 se ejecuta preferiblemente para cada segmento de la imagen. Sin embargo, las etapas S11 a S14 se ejecutan preferiblemente sólo para uno de los segmentos de la imagen, normalmente para el primer segmento de la imagen.

La cabecera de segmento analizada en la etapa S10 comprende información que hace posible que el decodificador genere un RPS. El RPS es un conjunto de imágenes de referencia asociado con la imagen actual, y consta de todas las imágenes de referencia que son anteriores a la imagen actual en el orden de decodificación y que se pueden utilizar como referencia, es decir, para una predicción intermedia, de la imagen actual o de cualquier imagen de la secuencia de video que va detrás de la imagen actual en el orden de decodificación.

La información obtenida en la etapa S10 y utilizada para determinar el RPS en la etapa S11 puede comprender, por ejemplo, un identificador de una estructura de sintaxis de RPS incluida en un conjunto de parámetros, tal como un SPS, aplicable al segmento actual. Un ejemplo de tal identificador aplicable a la norma HEVC es *short_term_ref_pic_set_idx*. El conjunto de parámetros comprende por tanto elementos de sintaxis que definen uno o más RPS, tales como *num_short_term_ref_pic_sets*, que definen el número de elementos de sintaxis *short_term_ref_pic_set( )* incluidos en el SPS, en donde el elemento de sintaxis *short_term_ref_pic_set( )* define un RPS candidato para la imagen actual.

Alternativamente, la información obtenida en la etapa S10 se podría utilizar directamente para determinar el RPS en la etapa S11. Esta información puede comprender, por ejemplo, un elemento de sintaxis *short_term_ref_pic_set( )*, en donde el elemento de sintaxis *short_term_ref_pic_set( )* define el RPS de la imagen actual.

Se puede encontrar más información sobre la determinación del RPS en la solicitud internacional de patente WO2013/002700 y en las secciones 7.3.2.2, 7.3.5.1, 7.3.5.2, 7.4.2.2, 7.4.5.1, 7.4.5.2 y 8.3.2 del documento JCTVC-J1003_d7, “High efficiency video coding (HEVC) text specification draft 8”, de B. Bross et al., 10ª conferencia de JCT-VC de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, Estocolmo, 11-20 de julio de 2012.

El RPS especifica, como se ha mencionado con anterioridad, las imágenes de la secuencia de video que se han de mantener en la DPB, es decir, que han de estar disponibles para su utilización como imágenes de referencia cuando se decodifica la imagen actual y/o cuando se decodifican las imágenes que van detrás de la imagen actual en el orden de decodificación. Esto significa que toda imagen de referencia almacenada en la DPB pero que no figure en el RPS no se utilizará como imagen de referencia ninguna vez más. Por lo tanto, la etapa S12 comprende de esta forma el marcado de tales imágenes presentes en la DPB pero que no están identificadas en el RPS como no utilizadas como referencia.

La salida de imágenes en la etapa S13 puede comprender la generación como salida de una imagen marcada como necesaria para la salida, la generación como salida de más de una imagen marcada como necesaria para la salida o, efectivamente, no generar como salida ninguna imagen. Todas las imágenes generadas como salida en la etapa S13 se generan como salida de acuerdo a un orden de salida, preferiblemente se generan como salida según el orden del menor valor del número de orden de imagen (POC, picture order count, por sus siglas en inglés).

Toda imagen generada como salida en la etapa S13 se marca a continuación como no necesaria para la salida debido a que la imagen ya se ha generado como salida. Si se generan como salida cero imágenes, evidentemente no se lleva a cabo ningún marcado en esta etapa S13.

Las imágenes que se marcan como no utilizadas como referencia y como no necesarias para la salida ya no se necesitan. Ni como imágenes de referencia ni para la salida. Por tanto, la etapa S14 vacía o elimina todas las imágenes con tal marcado de la DPB, para de esta forma liberar una memoria intermedia de almacenamiento de imágenes, es decir, espacio de memoria de imagen, en la DPB.

La decodificación de la imagen actual se lleva a cabo según métodos bien conocidos de decodificación de imágenes, tales como los especificados en la norma HEVC/H.265. Por tanto, la decodificación de la etapa S35 supone la generación de valores de píxeles para los píxeles o muestras de la imagen actual, utilizando normalmente los valores de los píxeles de una o más de las imágenes previamente decodificadas almacenadas en la DPB como imágenes de referencia.

En la realización que se muestra en la figura 9, el método que se muestra en la figura 8 y que comprende las etapas S1–S4 se podría considerar como un proceso de salida de imagen adicional o ampliado, el cual ya se ha mencionado con anterioridad como proceso de salida de imágenes #2. Esta realización hace posible de esta forma

la generación como salida de imágenes en la etapa S3, incluso después de que la imagen actual se haya decodificado en la etapa S15, pero antes de comenzar el procesamiento de una imagen siguiente de la secuencia de bits, es decir, antes de la ejecución de la etapa S10 para una nueva imagen en el orden de decodificación.

La figura 14 sintetiza esta realización. Por lo tanto, en una realización se añade una etapa de proceso de salida de imagen adicional después de la decodificación de la imagen, al objeto de hacer posible una salida de imágenes más rápida, véase la figura 14. Un método de decodificador puede comprender y/o un decodificador puede estar configurado para la ejecución de las siguientes etapas en orden, según la realización:

1. Se analiza la cabecera de segmento, incluyendo el RPS, de la primera cabecera de segmento de una imagen P.

2. Se lleva a cabo un proceso de marcado de imagen, por ejemplo, las imágenes se pueden marcar como no utilizadas como referencia por medio del RPS de la cabecera de segmento.

3. Se lleva a cabo un proceso de salida de imagen en el que se pueden generar como salida las imágenes.

4. Se decodifica la imagen P.

5. Después de decodificar la imagen P, el decodificador determina el número de imágenes de la DPB que se marcan como necesarias para la salida y compara ese número con un valor que se obtiene de los elementos de sintaxis de la secuencia de bits, representado por *sps_max_num_reorder_pics[HighestTid]*.

6. Si el número de imágenes de la DPB que se marcan como necesarias para la salida es mayor que el valor que se obtiene de los elementos de sintaxis de la secuencia de bits, se genera como salida la imagen de las imágenes de la DPB marcadas como necesarias para la salida que es la primera imagen en el orden de salida, y se marca como no necesaria para la salida. Por tanto, según la realización, se introduce el proceso de salida de imágenes #2.

7. La siguiente imagen Q se decodifica por medio de la repetición de las etapas 1–6 anteriores para la imagen Q.

El método descrito en la figura 8 se podría implementar también como parte de un proceso de salida modificado, por ejemplo para la norma HEVC. Según esta aproximación, se modifica la etapa existente de proceso de salida de imágenes al objeto de hacer posible una salida de imágenes más rápida.

La modificación proporciona una solución al problema de *no_output_of_prior_pics_flag*, tal y como se ha analizado en la sección de antecedentes. La solución es similar a las realizaciones analizadas con anterioridad, pero con el cambio de que, en lugar de la adición de un proceso de salida adicional, el proceso de salida HEVC existente se modifica cuando *no_output_of_prior_pics_flag* es igual a 1.

La figura 10 es un diagrama de flujo de una etapa adicional opcional del método de la figura 8, cuando se implementa la realización que se acaba de mencionar con anterioridad. La etapa S20 comprende el análisis de una cabecera de segmento de una imagen de acceso aleatorio (RAP), a la que también se hace referencia en la técnica como imagen de punto de acceso aleatorio interior (IRAP, intra random access point, por sus siglas en inglés), de la secuencia de video, al objeto de obtener un valor de un indicador *no_output_of_prior_pics_flag*. Si el valor del indicador *no_output_of_prior_pics_flag* es 1, el método avanza a la etapa S1 de la figura 8. Por tanto, en esta realización, la determinación del número en la etapa S1, la comparación del número en la etapa S2, la generación como salida de la imagen en la etapa S3 y el marcado de la imagen en la etapa S4 se llevan a cabo si el valor de *no_output_of_prior_pics_flag* es uno.

Por lo tanto, si el valor de *no_output_of_prior_pics_flag* es cero (0), el método termina y no se ejecuta el proceso de salida modificado.

Cuando esta realización se aplica a la norma HEVC, el método comprende preferiblemente las etapas adicionales de determinación de un RPS para la imagen de acceso aleatorio a partir de la cabecera de segmento analizada en la etapa S20 (compárese con la etapa S11 de la figura 9). El método comprende además el marcado de todas las imágenes de la DPB que no figuran en el RPS como no utilizadas como referencia (compárese con la etapa S12 de la figura 9). Se decodifica la imagen de acceso aleatorio (compárese con la etapa S15 de la figura 9). En esta realización, la determinación del número en la etapa S1, la comparación del número en la etapa S2, la generación como salida de la imagen en la etapa S3 y el marcado de la imagen en la etapa S4 se llevan a cabo después del marcado de las imágenes que no figuran en el RPS, pero antes de la decodificación de la imagen de acceso aleatorio.

La figura 15 es un diagrama de flujo de decodificación simplificado de esta realización. Un método de decodificador puede comprender y/o un decodificador puede estar configurado para la ejecución de las siguientes etapas en orden, según la realización:

1. Se analiza la cabecera de segmento de la primera cabecera de segmento de una imagen P.

2. Se lleva a cabo un proceso de marcado de imagen.

3. Se lleva a cabo un proceso de salida de imagen en el que se pueden generar como salida las imágenes. El proceso de salida de imagen se modifica de manera que si *no_output_of_prior_pics_flag* es igual a 1, se aplica lo siguiente:

a. El decodificador determina el número de imágenes de la DPB que están marcadas como necesarias para la salida y compara ese número con un valor que se obtiene de los elementos de sintaxis de la secuencia de bits.

b. Si el número de imágenes de la DPB que están marcadas como necesarias para la salida es mayor que el valor que se obtiene de los elementos de sintaxis de la secuencia de bits, se genera como salida la imagen de entre las imágenes de la DPB marcadas como necesarias para la salida que es la primera imagen en el orden de salida, y se marca como no necesaria para la salida.

c. Todas las imágenes restantes de la DPB que están marcadas como necesarias para la salida se marcan como no necesarias para la salida. No se generan como salida.

4. Se decodifica la imagen P.

La figura 11 es un diagrama de flujo de un método ejecutado por medio de un decodificador según otra realización. El método comprende el análisis de una cabecera de segmento de una imagen actual que se ha de decodificar de una secuencia de video en la etapa S30. Se determina un RPS para la imagen actual a partir de la cabecera de segmento analizada en la etapa S31. La siguiente etapa S32 comprende el marcado de todas las imágenes de la DPB que no están presentes en el RPS como no utilizadas como referencia. En la etapa S33, se generan como salida cero, una o muchas imágenes marcadas como necesarias para la salida de entre las imágenes de la DPB, y se marcan como no necesarias para la salida. La etapa S34 siguiente comprende el vaciado o eliminación, de la DPB, de toda imagen que esté marcada como no utilizada como referencia y como no necesaria para la salida de entre las imágenes de la DPB. La imagen actual se decodifica a continuación en la etapa S35.

Estas etapas S30–S35 se corresponden básicamente con las etapas S10–S15 analizadas con anterioridad e ilustradas en la figura 9.

El método avanza a continuación a la etapa S36, la cual comprende la determinación de un número de imágenes de la DPB que se marcan como necesarias para la salida. Este número se compara en la etapa S37 con un valor que se obtiene de al menos un elemento de sintaxis presente en una secuencia de bits que representa las imágenes de la secuencia de video. Si el número es mayor que el valor, el método avanza a las etapas S38 y S39. La etapa S38 comprende la generación como salida de una imagen, que es la primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida. La etapa S39 comprende el marcado de la imagen, generada como salida en la etapa S38, como no necesaria para la salida.

En esta realización, la determinación del número de imágenes en la etapa S37, la comparación del número en la etapa S38, la generación como salida de la imagen en la etapa S38 y el marcado de la imagen en la etapa S39 se llevan a cabo después de la decodificación de la imagen actual en la etapa S35.

Estas etapas S36 a S39 se corresponden con las etapas S1 a S4 analizadas con anterioridad y mostradas en la figura 8. Pero con la diferencia de que el valor con el que se compara en la etapa S37 el número determinado no tiene que ser necesariamente *sps_max_num_reorder_pics[HighestTid]*, sino que, en su lugar, podría ser un valor que se obtuviera de otro(s) elemento(s) de sintaxis presente(s) en la secuencia bits, lo cual se analiza en mayor detalle más adelante.

En una realización, si el número no es mayor que el valor, tal y como se determine en la comparación de la etapa S37, el método preferiblemente termina y no se genera como salida ni se marca ninguna imagen. Por tanto, en tal caso, las etapas S38 y S39 se omiten y no se ejecutan, véase la línea de puntos a la derecha.

En una realización, las etapas S37–S39 sólo se pueden ejecutar después de que se haya decodificado y preferiblemente almacenado en la DPB una imagen actual. De forma alternativa, el bucle formado por las etapas S37–S39, véase la línea de puntos a la izquierda, se puede ejecutar hasta que el número de imágenes de la DPB que se marcan como necesarias para la salida deje de ser mayor que el valor. Cada vez que se ejecuta el bucle de las etapas S37–S39, el número de imágenes de la DPB que están marcadas como necesarias para la salida se reduce en una unidad por medio del marcado como no necesaria para la salida, en la etapa S39, de la imagen generada como salida en la etapa S38.

La figura 12 es un diagrama de flujo de etapas adicionales opcionales del método de la figura 11, pero que también es aplicable a las realizaciones mostradas en las figuras 8 y 9. El método continúa desde la etapa S35 de la figura 11 o desde la etapa S15 de la figura 9, en las que se decodifica la imagen actual. Una etapa S40 siguiente comprende el almacenamiento de la imagen actual decodificada en la DPB en una memoria intermedia de almacenamiento de imágenes vacía, es decir, en un espacio de memoria de imagen. La imagen actual decodificada se marca en la etapa S41 como necesaria para la salida o como no necesaria para la salida. La imagen se marca también, opcionalmente, como utilizada como referencia a corto plazo, es decir, como utilizada como imagen de referencia. Esta etapa S41 se ejecuta preferiblemente en función de un PicOutputFlag que se asigna a la imagen

actual. Por lo tanto, si el indicador tiene un valor de 1, entonces la imagen actual decodificada se marca como necesaria para la salida y, en caso contrario, es decir, si el indicador tiene un valor de 0, la imagen actual decodificada se marca como no necesaria para la salida. El PicOutputFlag se puede obtener a partir de un elemento de sintaxis de la secuencia de bits aplicable a la imagen actual presente, tal como a partir del elemento de sintaxis *pic_output_flag*, que puede estar presente en la cabecera de segmento de la imagen actual.

El método avanza a continuación a la etapa S36 de la figura 11 o a la etapa S1 de la figura 8, en donde se determina el número de imágenes de la DPB marcadas como necesarias para la salida.

En la presente memoria, se describirán diferentes realizaciones de la selección del valor utilizado en la comparación de la etapa S37 de la figura 11 y en la de la etapa S2 de la figura 8.

En una realización, la secuencia de video es una secuencia de video multi-capa que comprende múltiples capas de imágenes. Cada capa de las múltiples capas tiene entonces un elemento de sintaxis asociado que define un respectivo valor utilizado en el proceso de salida. El método comprende además una etapa adicional opcional, tal y como se muestra en el diagrama de flujo de la figura 13. El método continúa desde la etapa S35 de la figura 11 o desde la etapa S1 de la figura 8. Una etapa S50 siguiente comprende la selección de un valor que se obtiene de un elemento de sintaxis asociado con la capa más alta que se decodifica por medio del decodificador de entre las múltiples capas. El método avanza a continuación a la etapa S37 de la figura 11 o a la etapa S2 de la figura 8, en donde se utiliza este valor seleccionado.

La etapa S37 y la etapa S2 comprenden preferiblemente, en esta realización, la comparación del número determinado en la etapa S36 o S1 con un valor de *sps_max_num_reorder_pics[HighestTid]*. Si el número es mayor que el valor *sps_max_num_reorder_pics[HighestTid]*, el método avanza a la etapa S38 o S3, las cuales comprenden, en esta realización, la generación como salida de una imagen de la DPB que tiene el menor valor de *PicOrderCntVal* de entre todas las imágenes de la DPB que están marcadas como necesarias para la salida. *PicOrderCntVal* representa un valor del número de orden de imagen de la imagen, el cual, a su vez, define el orden de salida de las imágenes almacenadas en la DPB.

En un aspecto particular de la implementación de esta realización, la secuencia de video es una secuencia de video multi-capa que comprende múltiples capas de imágenes. En este caso, cada capa tiene un valor respectivo, es decir, *sps_max_num_reorder_pics*[i] para la capa número i. El valor que se ha de utilizar es, por lo tanto, el valor asociado a la capa decodificada más alta, es decir, la capa más alta que se decodifica por medio del decodificador, como por ejemplo la sub-capa más alta si las capas son capas temporales diferentes.

En otra realización, la secuencia de video es una secuencia de video multi-capa que comprende múltiples capas de imágenes. Cada capa de las múltiples capas tiene entonces un elemento de sintaxis asociado que define un respectivo valor. El método comprende además una etapa adicional opcional, tal y como se muestra en el diagrama de flujo de la figura 13. El método continúa desde la etapa S36 de la figura 11. Una etapa S50 siguiente comprende la selección de un valor que se obtiene de un elemento de sintaxis asociado con la capa más alta de las múltiples capas. El método avanza a continuación a la etapa S37, en donde se utiliza este valor seleccionado.

La etapa S37 comprende preferiblemente, en esta realización, la comparación del número determinado en la etapa S36 con un valor de *sps_max_num_reorder_pics[sps_max_sub_layers_minus1]*. Si el número es mayor que el valor *sps_max_num_reorder_pics[sps_max_sub_layers_minus1]*, el método avanza a la etapa S38, la cual comprende, en esta realización, la generación como salida de una imagen de la DPB que tiene el menor valor de *PicOrderCntVal* de entre todas las imágenes de la DPB que están marcadas como necesarias para la salida. El elemento de sintaxis *sps_max_sub_layers_minus1* especifica un número máximo de capas de la secuencia de video.

En aspecto particular de la implementación de esta realización, la secuencia de video es una secuencia de video multi-capa que comprende múltiples capas de imágenes. En este caso, cada capa tiene un valor respectivo, es decir, *sps_max_num_reorder_pics*[i] para la capa número i. El valor que se ha de utilizar es, por lo tanto, el valor asociado a la capa más alta de la secuencia de bits.

En una realización adicional, la secuencia de video es una secuencia de video multi-capa que comprende múltiples capas de imágenes. Cada capa de las múltiples capas tiene entonces un elemento de sintaxis asociado que define un respectivo valor. El método comprende además una etapa adicional opcional, tal y como se muestra en el diagrama de flujo de la figura 13. El método continúa desde la etapa S36 de la figura 11. Una etapa S50 siguiente comprende la selección de un valor que se obtiene de un elemento de sintaxis asociado con una capa de las múltiples capas a la cual pertenece la imagen actual decodificada. El método avanza a continuación a la etapa S37, en donde se utiliza este valor seleccionado.

En otra realización más, la secuencia de video es una secuencia de video multi-capa que comprende múltiples capas de imágenes. Cada capa de las múltiples capas tiene entonces un elemento de sintaxis asociado que define un respectivo valor. El método comprende además una etapa adicional opcional, tal y como se muestra en el diagrama de flujo de la figura 13. El método continúa desde la etapa S36 de la figura 11. Una etapa S50 siguiente comprende la selección de un valor máximo de entre los respectivos valores.

El método mostrado en la figura 11 se puede aplicar a una imagen actual que es una imagen de acceso aleatorio de la secuencia de video. En este caso, la etapa S30 comprende preferiblemente el análisis de la cabecera de segmento de la imagen de acceso aleatorio al objeto de obtener un valor de un indicador *no_output_prior_flag*. En tal caso, la determinación del número en la etapa S36, la comparación del número en la etapa S37, la generación como salida de la imagen en la etapa S38 y el marcado de la imagen en la etapa S39 podrían depender del valor de este indicador. Por lo tanto, en una realización opcional, estas etapas S36–S39 se ejecutan si el valor del indicador *no_output_prior_flag* es 1.

En una realización, el marcado de la imagen, tal y como se lleva a cabo en la etapa S12 de la figura 9 y en la S32 de la figura 11, comprende preferiblemente el marcado de todas las imágenes de la DPB que no están presentes en el RPS como no utilizadas como referencia. Estas etapas comprenden también, opcionalmente, el marcado de todas las imágenes de la DPB que están presentes en el RPS como utilizadas como referencia. Sin embargo, en general se marca una imagen como utilizada como referencia una vez que se ha decodificado y almacenado en la DPB. Por lo tanto, las etapas S12 y S32 implican normalmente el remarcado de las imágenes que ya no se necesitan como referencia.

En una realización, la generación como salida de la imagen, tal y como se lleva a cabo en la etapa S3 de la figura 8 y en la etapa S38 de la figura 11, comprende preferiblemente la generación como salida de una imagen que tiene el valor del número de orden de imagen más pequeño de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor determinado en la etapa S2 o S37. El valor del número de orden de imagen se representa preferiblemente por medio del parámetro *PicOrderCntVal*.

En una realización, la generación como salida de la imagen, tal y como se lleva a cabo en la etapa S3 de la figura 8 y en la etapa S38 de la figura 11, se lleva a cabo preferiblemente antes de analizar una cabecera de segmento de una imagen siguiente de la secuencia de video codificada que se ha de decodificar.

En una realización, el método, tal y como se lleva a cabo en la figura 8 o en la figura 11, comprende la etapa adicional de recuperación de al menos un elemento de sintaxis de un conjunto de parámetros asociado con la secuencia de bits, y seleccionado de un grupo que consta de un PPS, un SPS y un VPS. Se identifica un PPS a partir de un identificador PPS presente en la cabecera de segmento de la imagen actual. Un SPS que se aplica a la imagen actual se identifica por medio de un identificador SPS presente en el PPS identificado por medio del identificador PPS presente en la cabecera de segmento de la imagen actual. De forma correspondiente, un VPS que se aplica a la imagen actual se identifica por medio de un identificador VPS presente en el SPS que se aplica a la imagen actual. En una realización particular, el al menos un elemento de sintaxis se recupera de un SPS.

Una realización particular, que es aplicable a una implementación que utiliza *no_output_prior_pics_flag*, comprende una etapa adicional preferiblemente de marcado de todas las imágenes restantes en la DPB marcadas como necesarias para la salida a como no necesarias para la salida si el indicador *no_output_prior_pics_flag* es uno.

En una realización relacionada, el método comprende la etapa adicional de vaciado de todas las imágenes de la DPB, sin generación como salida de ninguna imagen, si el indicador *no_output_prior_pics_flag* es uno. La completitud de la DPB se fija igual a cero a continuación, para indicar que la DPB está vacía.

A continuación, en la presente memoria se analizarán en mayor detalle diferentes realizaciones a modo de ejemplo.

Realización a modo de ejemplo 1

En una realización, se añade una etapa adicional al proceso de salida de imagen, después de la decodificación de la imagen, al objeto de hacer posible que la generación como salida de imágenes sea más rápida, véase la figura 14.

Un método de decodificador puede comprender y/o un decodificador puede estar configurado para la ejecución de las siguientes etapas en orden, según la realización:

1. Se analiza la cabecera de segmento de la primera cabecera de segmento de la imagen P.

2. Se lleva a cabo un proceso de marcado de imagen.

3. Se lleva a cabo un proceso de salida de imagen en el que se pueden generar como salida las imágenes.

4. Se decodifica la imagen P.

5. Después de decodificar la imagen P, el decodificador determina el número de imágenes de la DPB que están marcadas como “necesarias para la salida” y compara ese número con un valor que se obtiene de los elementos de sintaxis (que se muestran a modo de ejemplo en la realización 9) de la secuencia de bits.

6. Si el número de imágenes de la DPB que están marcadas como “necesarias para la salida” es mayor que el valor que se obtiene de los elementos de sintaxis de la secuencia de bits, se genera como salida la imagen de las imágenes de la DPB marcadas como “necesarias para la salida” que es la primera imagen en el orden de salida, y

se marca como “no necesaria para la salida”. Por tanto, según la realización, se introduce el proceso de salida de imágenes #2.

7. Se decodifica una imagen Q siguiente.

Realización a modo de ejemplo 2

La misma que la realización a modo de ejemplo 1, en la que el valor se representa por medio de una única clave del elemento de sintaxis de la secuencia de bits.

Realización a modo de ejemplo 3

La misma que la realización a modo de ejemplo 2, en la que el valor se representa por medio de una clave en un conjunto de parámetros de secuencia, por ejemplo, un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de video (VPS).

Realización a modo de ejemplo 4

La misma que la realización 1–3, en la que hay un valor dependiente de capa que se obtiene para cada capa en la secuencia de bits, y el valor que se ha de utilizar en la comparación es el valor que se obtiene para la misma capa que la imagen P.

Realización a modo de ejemplo 5

La misma que la realización 1–3, en la que hay un valor que se obtiene para cada capa en la secuencia de bits, y el valor que se ha de utilizar en la comparación es el valor que se obtiene para la capa más alta de la secuencia de bits.

Realización a modo de ejemplo 6

La misma que la realización 1–3, en la que hay un valor que se obtiene para cada capa en la secuencia de bits, y el valor que se ha de utilizar en la comparación es el valor máximo de todas las capas.

Realización a modo de ejemplo 7

La misma que la realización 5, en la que la capa más alta se define como la sub-capa más alta.

Realización a modo de ejemplo 8

La misma que la realización 1–7, en la que la primera imagen en el orden de salida se define como la imagen de la DPB que tiene el menor valor de PicOrderCntVal. PicOrderCntVal define el POC y se ha descrito con anterioridad.

Realización a modo de ejemplo 9

La misma que la realización 4–8, en la que las capas son capas temporales.

Realización a modo de ejemplo 10

La combinación de las realizaciones 1, 2, 3, 5, 8 y 9 es una realización preferida. La expresión de esta realización en la especificación actual de la norma HEVC (JCTVC-K0030_v3) podría dar lugar a los siguientes cambios en la especificación de la norma HEVC en la sección C.5.3, en donde el texto en negrita significa texto añadido:

C.5.3 Decodificación, marcado y almacenamiento de imágenes y posible generación como salida de una imagen

Lo que sigue ocurre de forma instantánea cuando la última unidad de decodificación de la unidad de acceso n que contiene la imagen actual se elimina de la CPB.

La imagen actual se considera decodificada después de que se decodifica la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en una memoria intermedia de almacenamiento de imágenes vacía de la DPB, y se aplica lo siguiente.

1.- Si la imagen decodificada actual tiene PicOutputFlag igual a 1, se marca como “necesaria para la salida”.

2.- En caso contrario (la imagen decodificada actual tiene PicOutputFlag igual a 0), se marca como “no necesaria para la salida”.

Si la imagen decodificada actual es una imagen de referencia, se marca como “utilizada como referencia”, en caso contrario (la imagen decodificada actual no es una imagen de referencia), se marca como “no utilizada como referencia”.

Cuando el número de imágenes de la DPB que están marcadas como "necesaria para la salida" es mayor que *sps_max_num_reorder_pics[sps_max_sub_layers_minus1]* después de que la imagen decodificada actual se ha almacenado en la DPB, la imagen de la DPB que tiene el menor valor de PicOrderCntVal de entre todas las imágenes de la DPB se recorta, se genera como salida y se marca como "no necesaria para la salida".

Realización a modo de ejemplo 11

La combinación de las realizaciones 1, 2, 3, 7, 8 y 9 es otra realización preferida. La expresión de esta realización en la especificación actual de la norma HEVC (JCTVC-K0030_v3) podría dar lugar a los siguientes cambios en la especificación de la norma HEVC en la sección C.5.3, en donde el texto en negrita significa texto añadido:

C.5.3 Decodificación, marcado y almacenamiento de imágenes y posible generación como salida de una imagen

Lo que sigue ocurre de forma instantánea cuando la última unidad de decodificación de la unidad de acceso n que contiene la imagen actual se elimina de la CPB.

La imagen actual se considera decodificada después de que se decodifica la última unidad de decodificación de la imagen. La imagen decodificada actual se almacena en una memoria intermedia de almacenamiento de imágenes vacía de la DPB, y se aplica lo siguiente.

1.- Si la imagen decodificada actual tiene PicOutputFlag igual a 1, se marca como "necesaria para la salida".

2.- En caso contrario (la imagen decodificada actual tiene PicOutputFlag igual a 0), se marca como "no necesaria para la salida".

Si la imagen decodificada actual es una imagen de referencia, se marca como "utilizada como referencia", en caso contrario (la imagen decodificada actual no es una imagen de referencia), se marca como "no utilizada como referencia".

Cuando el número de imágenes de la DPB que están marcadas como "necesaria para la salida" es mayor que *sps_max_num_reorder_pics[HighestTid]* después de que la imagen decodificada actual se ha almacenado en la DPB, la imagen de la DPB que tiene el menor valor de PicOrderCntVal de entre todas las imágenes de la DPB se recorta, se genera como salida y se marca como "no necesaria para la salida".

Realización a modo de ejemplo 12

En una realización, se modifica la etapa existente de proceso de salida de imagen, al objeto de hacer posible que la generación como salida de imágenes sea más rápida, véase la figura 15.

La modificación proporciona una solución al problema de *no_output_of_prior_pics_flag*, tal y como se ha descrito con anterioridad. La solución es similar a las realizaciones a modo de ejemplo 1–11, pero con el cambio de que, en lugar de la adición de un proceso de salida adicional, el proceso de salida HEVC existente se modifica cuando *no_output_of_prior_pics_flag* es igual a 1, tal y como se muestra a continuación.

Un método de decodificador puede comprender y/o un decodificador puede estar configurado para la ejecución de las siguientes etapas en orden, según la realización:

1. Se analiza la cabecera de segmento de la primera cabecera de segmento de una imagen P.

2. Se lleva a cabo un proceso de marcado de imagen.

3. Se lleva a cabo un proceso de salida de imagen en el que se pueden generar como salida las imágenes. El proceso de salida de imagen se modifica de manera que si *no_output_of_prior_pics_flag* es igual a 1, se aplica lo siguiente:

    a. El decodificador determina el número de imágenes de la DPB que están marcadas como "necesarias para la salida" y compara ese número con un valor que se obtiene de los elementos de sintaxis de la secuencia de bits.

    b. Si el número de imágenes de la DPB que están marcadas como "necesarias para la salida" es mayor que el valor que se obtiene de los elementos de sintaxis de la secuencia de bits, se genera como salida la imagen de las imágenes de la DPB marcadas como "necesarias para la salida" que es la primera imagen en el orden de salida, y se marca como "no necesaria para la salida".

    c. Todas las imágenes restantes de la DPB que están marcadas como "necesarias para la salida" se marcan como "no necesarias para la salida". No se generan como salida.

4. Se decodifica la imagen P.

Las realizaciones a modo de ejemplo 2–12 se aplican también a esta realización a modo de ejemplo 12.

Realización a modo de ejemplo 13

La combinación de la realización a modo de ejemplo 12 y de las realizaciones a modo de ejemplo 2, 3, 7, 8 y 9 es una realización preferida. La expresión de esta realización en la especificación actual de la norma HEVC (JCTVC-K0030_v3) podría dar lugar a los siguientes cambios en la especificación de la norma HEVC en la sección C.3.1, en donde el texto en negrita significa texto añadido:

C.3.1 Eliminación de imágenes de la DPB

La eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después del análisis de la cabecera de segmento del primer segmento de la imagen actual) ocurre de forma instantánea en el instante de eliminación de la CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene la imagen actual), y se desarrolla como sigue.

Se invoca el proceso de decodificación para el conjunto de imágenes de referencia, tal y como se especifica en el punto 8.3.2.

Si la imagen actual es una imagen IDR o una imagen BLA, se aplica lo siguiente:

1. Cuando la imagen IDR o BLA no es la primera imagen decodificada y el valor de *pic_width_in_luma_samples* o *pic_height_in_luma_samples* o *sps_max_dec_pic_buffering*[i] para cualquier posible valor de i que se obtiene del conjunto de parámetros de secuencia activo es diferente del valor de *pic_width_in_luma_samples* o *pic_height_in_luma_samples* o *sps_max_dec_pic_buffering*[i] que se obtiene del conjunto de parámetros de secuencia que estaba activo para la imagen precedente, respectivamente, se infiere que *no_output_of_prior_pics_flag* es igual a 1 por parte del HRD, con independencia del valor real de *no_output_of_prior_pics_flag.*

Nota – Las implementaciones del decodificador deben intentar gestionar la imagen o los cambios de tamaño de la DPB de forma más adecuada que lo que hace el HRD con respecto a los cambios en *pic_width_in_luma_samples* o *pic_height_in_luma_samples* o *sps_max_dec_pic_buffering*[i].

2. Cuando *no_output_of_prior_pics_flag* es igual a 1, o se infiere que es igual a 1, **se aplican las siguientes etapas en orden:**

1. Cuando el número de imágenes de la DPB que están marcadas como "necesaria para la salida" es mayor que *sps_max_num_reorder_pics[HighestTid]*, la imagen de la DPB que tiene el menor valor de PicOrderCntVal de entre todas las imágenes de la DPB se recorta, se genera como salida y se marca como "no necesaria para la salida".

2. Todas las memorias intermedias de almacenamiento de imágenes de la DPB se vacían, sin generar como salida las imágenes que contienen, y la completitud de la DPB se fija a cero.

Todas las imágenes k de la DPB, para las cuales son verdaderas todas las condiciones siguientes, se eliminan de la DPB:

- la imagen k está marcada como "no utilizada como referencia".

- la imagen k tiene PicOrderCntVal igual a 0, o su instante de salida de la DPB es menor o igual al instante de eliminación de la CPB de la primera unidad de decodificación (a la que se hace referencia como unidad de decodificación m) de la imagen actual n; es decir, $t_{o,dpb}$ (k) <= t_r (m).

Cuando se elimina una imagen de la DPB, la completitud de la DPB se reduce en una unidad.

A continuación, haciendo referencia a la figura 30, se describirá una ventaja de las realizaciones descritas en la presente memoria, con respecto a la técnica anterior, tal y como se representa por medio de la sección C.5.1, en las páginas 216–217 del documento JCTVC-K0030_v3. La solución de la técnica anterior en el documento JCTVC-K0030_v3 utiliza el parámetro *sps_max_num_reorder_pics*[TemporalID] para determinar si se ha de generar como salida alguna imagen. En particular, el documento JCTVC-K0030_v3 estipula que cuando el número de imágenes de la DPB que están marcadas como necesarias para la salida es mayor que *sps_max_num_reorder_pics*[TemporalID], entonces el proceso de transferencia rápida especificado en el punto C.5.2.1 se invoca repetidamente hasta que hay una memoria intermedia de almacenamiento de imágenes vacía para el almacenamiento de la imagen decodificada actual.

Los números que se presentan en la figura 30 indican el orden de decodificación de las imágenes de la secuencia de video. Las imágenes se deben generar como salida de izquierda a derecha, es decir, I0, b5, B3, b6, B2, b7, B4, B8 y P1. La figura indica además el número de capa, es decir, TemporalID, de las diferentes capas y el parámetro *sps_max_num_reorder_pics* para cada capa. Si se transfirieran o se generaran como salida las imágenes, junto con

el análisis de las cabeceras de segmento por medio de la utilización de *sps_max_num_reorder_pics*[TemporalID], tal y como se sugiere en el documento JCTVC-K0030_v3, entonces el proceso de salida sería de acuerdo a lo siguiente:

Imagen	No generada como salida aún	*sps_max_num_reorder_pics*	Acción
I0	-	0	Sin acción
B1	I0	0	Salida I0
B2	B1	1	Sin acción
B3	B1 B2	2	Sin acción
B4	B1 B2 B3	2	Salida B3

Sin embargo, generar como salida la imagen B3 cuando se procesa la imagen B4 es incorrecto, debido a que la imagen b5 se debería generar como salida antes que la imagen B3.

Si se utilizara una realización como la descrita en la presente memoria para la transferencia, es decir, para la generación como salida, después de la decodificación de la imagen actual y utilizando HighesTId=3, entonces se obtendría el siguiente resultado.

Imagen	No generada como salida aún	*sps_max_num_reorder_pics*	Acción
I0	I0	4	Sin acción
B1	I0 B2	4	Sin acción
B2	I0 B1 B2	4	Sin acción
B3	I0 B1 B2 B3	4	Sin acción
B4	I0 B1 B2 B3 B4	4	Salida I0
B5	B1 B2 B3 B4 b5	4	Salida b5

Si se utiliza una escala temporal y sólo se decodifican las capas 0–2, entonces HighesTid es 2 y se obtiene el siguiente resultado según una realización:

Imagen	No generada como salida aún	*sps_max_num_reorder_pics*	Acción
I0	I0	2	Sin acción
B1	I0 B1	2	Sin acción
B2	I0 B1 B2	2	Salida I0
B3	B1 B2 B3	2	Salida B3
B4	B1 B2 B4	2	Salida B2

Por lo tanto, las realizaciones descritas en la presente memoria son capaces de generar como salida las imágenes en el orden correcto en el ejemplo ilustrado en la figura 30, mientras que la solución de la técnica anterior en el documento JCTVC-K0030_v3 no proporciona el orden de salida correcto cuando se invoca el proceso de transferencia después del análisis de las cabeceras de segmento.

Las etapas, funciones, procedimientos, módulos y/o bloques descritos con anterioridad en relación a las figuras 8–15 se pueden implementar en hardware por medio de la utilización de tecnología convencional, tal como tecnología de circuitos discretos o tecnología de circuitos integrados, incluyendo ambos, sistemas de circuitos de propósito general y sistemas de circuitos específicos de aplicación.

Los ejemplos particulares incluyen uno o más procesadores de señales digitales configurados de forma adecuada y otros circuitos electrónicos conocidos, por ejemplo, puertas lógicas discretas interconectadas al objeto de llevar a cabo una función, o circuitos integrados para aplicaciones específicas (ASICs, application specific integrated circuits, por sus siglas en inglés).

Alternativamente, al menos algunas de las etapas, funciones, procedimientos, módulos y/o bloques descritos con anterioridad en relación a las figuras 8–15 se pueden implementar en software, tal como en un programa de ordenador que se ejecuta por medio de circuitos de procesamiento apropiados que incluyen uno o más procesadores.

El procesador es capaz de ejecutar instrucciones software contenidas en un programa de ordenador almacenado en un producto de programa de ordenador, por ejemplo, en forma de memorias. El respectivo producto de programa de ordenador puede ser una memoria que sea una combinación de una memoria de acceso aleatorio (RAM, random access memory, por sus siglas en inglés) y una memoria de solo lectura (ROM, read-only-memory, por sus siglas en inglés). La memoria respectiva comprende un almacenamiento persistente que, por ejemplo, puede ser una cualquiera de entre memoria magnética, memoria óptica, memoria de estado sólido o incluso memoria montada de forma remota, o una combinación de las mismas.

El diagrama o diagramas de flujo presentados con anterioridad y mostrados en las figuras 8–15 se pueden considerar, por tanto, como un diagrama o diagramas de flujo de ordenador, cuando se ejecutan por medio de uno o más procesadores. Un aparato correspondiente se puede definir como un grupo de módulos de funciones, en donde cada etapa ejecutada por medio del procesador se corresponde con un módulo de funciones. En este caso, los módulos de funciones se implementan como un programa de ordenador que se ejecuta en el procesador.

Los ejemplos de circuitos de procesamiento incluyen, aunque no se limitan a los mismos, uno o más microprocesadores, uno o más procesadores de señales digitales (DSPs, digital signal processors, por sus siglas en inglés), una o más unidades centrales de proceso (CPUs, central processing units, por sus siglas en inglés), hardware de aceleración de video, y/o cualesquiera circuitos lógicos programables adecuados, tal como uno o más conjuntos de puertas programables por el usuario (FPGAs, field programable gate arrays, por sus siglas en inglés), o uno o más controladores de lógica programable (PLCs, programable logic controllers, por sus siglas en inglés).

Se ha de entender además que puede ser posible reutilizar las capacidades de procesamiento generales de cualquier dispositivo o unidad convencional en el que se implemente la tecnología propuesta. Puede ser posible también reutilizar software existente, por ejemplo, por medio de la reprogramación del software existente o por medio de la adición de nuevos componentes software.

Según un aspecto, se proporciona un decodificador configurado para llevar a cabo un método según cualquiera de las realizaciones descritas con anterioridad. El decodificador está configurado para la determinación, después de que se haya decodificado y almacenado en una DPB una imagen actual de una secuencia de bits que representa las imágenes de una secuencia de video, de un número de imágenes de la DPB que están marcadas como necesarias para la salida. El decodificador está configurado además para la comparación del número con un valor *sps_max_num_reorder_pics[HighestTid]*. El decodificador está configurado también para la generación como salida de una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor. El decodificador está configurado adicionalmente para marcar la imagen (que se ha generado como salida) como no necesaria para la salida si el número es mayor que el valor.

El decodificador 100 comprende, en una realización, un procesador 110 configurado para ejecutar las etapas del método descrito previamente en la presente memoria, véase la figura 8 y opcionalmente las figuras 9, 10 y 12–14. El decodificador 100 puede comprender además una memoria 120 conectada al procesador 110, véase la figura 16.

La figura 16 es un diagrama de bloques esquemático de un decodificador 100 según una realización. El decodificador 100 está configurado para recibir una secuencia de bits 10 que representa las imágenes de una secuencia de video y decodificar la secuencia de bits 10. El decodificador 100 comprende un procesador 110 y una memoria 120 que comprende una DPB 125. El procesador 110 está configurado para la determinación, después de que se haya decodificado y almacenado en la DPB 125 la imagen actual, del número de imágenes de la DPB 125 que están marcadas como necesarias para la salida. El procesador 110 está configurado además para la comparación del número con el valor *sps_max_num_reorder_pics[HighestTid]* 14. El procesador 110 está configurado también para la generación como salida de la imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB 125 que están marcadas como necesarias para la salida, si el número es mayor que el valor. El procesador 110 está configurado adicionalmente para marcar la imagen como no necesaria para la salida si el número es mayor que el valor.

En la figura 16, se ha ilustrado que el decodificador 100 comprende un procesador 110. Este procesador 110 se podría implementar como un único procesador o como múltiples procesadores, por ejemplo, como un circuito de procesamiento.

La figura 16 ilustra de esta forma una implementación en ordenador del decodificador 100. En este ejemplo particular, al menos algunas de las etapas, funciones, procedimientos, módulos y/o bloques descritos con anterioridad se implementan en un programa de ordenador, el cual se carga en la memoria 120 para su ejecución por parte del procesador 110. El procesador 110 y la memoria 120 están interconectados entre sí al objeto de hacer posible la ejecución normal del software. También se puede interconectar un dispositivo de entrada/salida opcional (no mostrado) al procesador 110 y/o a la memoria 120 para hacer posible la entrada de una secuencia de bits 10 de imágenes codificadas y la salida de las imágenes decodificadas.

El término “ordenador” se debe interpretar en sentido general, como cualquier sistema, dispositivo o aparato capaz de ejecutar código de programa o instrucciones de programa de ordenador para llevar a cabo una tarea concreta de procesamiento, determinación o cálculo.

En una realización, el procesador 110 está configurado preferiblemente para el análisis de una cabecera de segmento 12 de la imagen actual que se ha de decodificar de la secuencia de video. El procesador 110 está configurado también para determinar un RPS para la imagen actual a partir de la cabecera de segmento 12 analizada. El procesador 110 está configurado además para el marcado de todas las imágenes de la DPB 125 que no están presentes en el RPS como no utilizadas como referencia. El procesador 110 está configurado adicionalmente para generar como salida cero, una o muchas imágenes marcadas como necesarias para la salida de entre las imágenes de la DPB 125, y para marcar las cero, una o muchas imágenes como no necesarias para la salida. El procesador 110 está configurado preferiblemente además para el vaciado o eliminación, de la DPB 125, de toda imagen que esté marcada como no utilizada como referencia y como no necesaria para la salida de entre las imágenes de la DPB 125. El procesador 110 está configurado además para decodificar la imagen actual, por medio de la utilización preferiblemente de la carga útil de datos de video 16 proporcionados en forma codificada. En esta realización, el procesador 110 está configurado para determinar el número de imágenes, comparar el número, generar como salida la imagen y marcar la imagen después de decodificar la imagen actual.

En una realización, el procesador 110 está configurado opcionalmente para el análisis de una cabecera de segmento 12 de una imagen de acceso aleatorio de la secuencia de video, al objeto de obtener un valor del indicador *no_output_prior_pics_flag*. En tal caso, el procesador 110 está configurado opcionalmente para determinar el número, comparar el número, generar como salida la imagen y marcar la imagen si (y sólo si) el valor del indicador *no_output_prior_pics_flag* es uno.

Otro aspecto de las realizaciones se refiere a un decodificador configurado para analizar una cabecera de segmento de una imagen actual de una secuencia de video que se ha de decodificar. El decodificador está configurado también para determinar un RPS para la imagen actual a partir del análisis de la cabecera de segmento. El decodificador está configurado además para marcar todas las imágenes de la DPB que no están presentes en el RPS como no utilizadas como referencia. El decodificador está configurado adicionalmente para generar como salida cero, una o muchas imágenes, marcadas como necesarias para la salida, de entre las imágenes de la DPB, y para marcar las cero, una o muchas imágenes como no necesarias para la salida. El decodificador está configurado preferiblemente además para vaciar o eliminar, de la DPB, toda imagen marcada como no utilizada como referencia y como no necesaria para la salida de entre las imágenes de la DPB. El decodificador está configurado además para decodificar la imagen actual. En esta realización, el decodificador está configurado para determinar un número de imágenes de la DPB que están marcadas como necesarias para la salida. El decodificador está configurado también para comparar el número con un valor que se obtiene de al menos un elemento de sintaxis presente en la secuencia de bits. El decodificador está configurado además para generar como salida una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor. El decodificador está configurado adicionalmente para marcar la imagen como no necesaria para la salida si el número es mayor que el valor. Esto significa que en esta realización, el decodificador está configurado para determinar el número de imágenes, comparar el número, generar como salida la imagen y marcar la imagen después de decodificar la imagen actual.

El codificador 100 comprende, en una realización, un procesador 110 configurado para ejecutar las etapas del método descrito previamente en la presente memoria, véase la figura 11 y opcionalmente las figuras 12, 13 y 15. El codificador 100 puede comprender además una memoria 120 conectada al procesador 110, véase la figura 16.

En esta realización, el decodificador 100 comprende, por tanto, un procesador 110 y una memoria 120 que comprende una DPB 125. El procesador 110 está configurado para analizar la cabecera de segmento 12 de la imagen actual de la secuencia de video que se ha de decodificar. El procesador 110 está configurado también para determinar el RPS para la imagen actual a partir del análisis de la cabecera de segmento 12. El procesador 110 está configurado además para marcar todas las imágenes de la DPB 125 que no están presentes en el RPS como no utilizadas como referencia. El procesador 110 está configurado adicionalmente para generar como salida cero, una o muchas imágenes, marcadas como necesarias para la salida, de entre las imágenes de la DPB 125, y para marcar las cero, una o muchas imágenes como no necesarias para la salida. El procesador 110 está configurado preferiblemente además para vaciar o eliminar, de la DPB 125, toda imagen marcada como no utilizada como referencia y como no necesaria para la salida de entre las imágenes de la DPB 125. El procesador 110 está configurado además para decodificar la imagen actual. En esta realización, el procesador 110 está configurado para determinar el número de imágenes de la DPB 125 que están marcadas como necesarias para la salida. El procesador 110 está configurado también para comparar el número con un valor que se obtiene de al menos un elemento de sintaxis 14 presente en la secuencia de bits 10. El procesador 110 está configurado además para generar como salida la imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB 125 que están marcadas como necesarias para la salida, si el número es mayor que el valor. El procesador 110 está configurado adicionalmente para marcar la imagen como no necesaria para la salida si el número es mayor que el valor. Esto significa que en esta realización, el procesador 110 está configurado para determinar el número de imágenes, comparar el número, generar como salida la imagen y marcar la imagen después de decodificar la imagen actual.

En una realización, el procesador 110 está configurado preferiblemente para almacenar la imagen actual decodificada en la DPB 125 y para marcar la imagen actual decodificada como necesaria para la salida o como no necesaria para la salida, tal y como se ha descrito con anterioridad en la presente memoria.

En una realización de implementación, el procesador 110 está configurado preferiblemente para comparar el número con el valor *sps_max_num_reorder_pics[HighestTid]*. En esta realización de implementación, el procesador 110 está configurado para generar como salida una imagen de la DPB 125 que tiene un menor valor de *PicOrderCntVal* de entre todas las imágenes de la DPB 125 que están marcadas como necesarias para la salida, si el número es mayor que el valor *sps_max_num_reorder_pics[HighestTid]*.

En otra realización de implementación, la imagen actual es una imagen de acceso aleatorio de la secuencia de video. En este caso, el procesador 110 está configurado opcionalmente para analizar la cabecera de segmento 12 de la imagen de acceso aleatorio al objeto de obtener un valor de un indicador *no_output_prior_pics_flag*. El procesador 110 está configurado en ese caso de forma opcional para determinar el número, comparar el número, generar como salida la imagen y marcar la imagen si el valor del indicador *no_output_prior_pics_flag* es uno.

Si la secuencia de video es una secuencia de video multi-capa que comprende múltiples capas de imágenes, en donde cada capa de las múltiples capas tiene un elemento de sintaxis asociado que define un valor respectivo, entonces el procesador 110 está configurado preferiblemente para seleccionar un valor que se obtiene de un elemento de sintaxis, preferiblemente *sps_max_num_reorder_pics*, asociado con la capa más alta que se decodifica por medio del decodificador 100 de entre las múltiples capas.

En una realización, el procesador 110 está configurado preferiblemente para generar como salida una imagen que tiene el menor valor del número de orden de imagen de entre las imágenes de la DPB 125 que están marcadas como necesarias para la salida, si el número es mayor que el valor.

En una realización, el procesador 110 está configurado preferiblemente para generar como salida la imagen antes de analizar una cabecera de segmento de una imagen siguiente que se ha de decodificar de la secuencia de video.

La figura 17 es un diagrama de bloques esquemático de otro ejemplo de implementación del decodificador 200. Este ejemplo es especialmente apropiado para una implementación hardware del decodificador 200. El decodificador 200 comprende una unidad de entrada 210 configurada para recibir la secuencia de bits que representa las imágenes de la secuencia de video y para almacenar la secuencia de bits en una memoria 220 conectada que comprende la DPB 225. El decodificador comprende también una unidad de determinación de número 230 conectada a la memoria 220. Esta unidad de determinación de número 230 está configurada para determinar, después de que la imagen actual se haya decodificado y almacenado en la DPB 225, el número de imágenes de la DPB 225 que están marcadas como necesarias para la salida. Un comparador 240 está conectado a la unidad de determinación de número 230 y está configurado para comparar el número con el valor *sps_max_num_reorder_pics[HighestTid]*. El decodificador 200 comprende además una unidad de salida 250 conectada al comparador 240 y preferiblemente a la memoria 220. La unidad de salida 250 está configurada para generar como salida la imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB 225 que están marcadas como necesarias para la salida, si el número es mayor que el valor. Una unidad de marcado 260 está conectada a la memoria 220 y está configurada para marcar la imagen como no necesaria para la salida si el número es mayor que el valor.

El comparador 240 está conectado preferiblemente a la unidad de determinación de número 230 al objeto de recibir el número de imágenes determinado por medio de la unidad de determinación de número 230. De forma correspondiente, la unidad de salida 250 está conectada preferiblemente al comparador 240 al objeto de recibir información del comparador 240 sobre si el número determinado es mayor que el valor *sps_max_num_reorder_pics[HighestTid]*. La unidad de marcado 260 está conectada preferiblemente a la unidad de salida 250 al objeto de recibir información sobre qué imagen se ha generado como salida por medio de la unidad de salida 250.

La figura 18 es un diagrama de bloques esquemático de otro ejemplo más de implementación del decodificador 300. Este ejemplo es especialmente apropiado para una implementación hardware del decodificador 300. El decodificador 300 comprende una unidad de entrada 310 configurada para recibir la secuencia de bits que representa las imágenes de la secuencia de video y para almacenar la secuencia de bits en una memoria 320 conectada que comprende la DPB 325. Una unidad de análisis 370, conectada a la memoria 320, está configurada para analizar la cabecera de segmento de la imagen actual que se ha de decodificar de la secuencia de video. El decodificador 300 comprende una unidad de determinación del conjunto de imágenes de referencia 380 conectada a la unidad de análisis 370 y preferiblemente a la memoria 320. La unidad de determinación del conjunto de imágenes de referencia 380 está configurada para determinar el RPS para la imagen actual a partir del análisis de la cabecera de segmento. Una unidad de marcado 360 está conectada a la memoria 320 y está configurada para marcar todas las imágenes de la DPB 325 que no están presentes en el RPS como no utilizadas como referencia. El decodificador 300 comprende además una unidad de salida 350 conectada a la memoria 320 y preferiblemente a un comparador 340. La unidad de salida 350 está configurada para generar como salida cero, una o muchas imágenes, marcadas como necesarias para la salida, de entre las imágenes de la DPB 325, en el que la unidad de marcado 360 está configurada para marcar las cero, una o muchas imágenes como no necesarias para la salida. Una unidad de

vaciado de imágenes 390 está conectada a la memoria 320 y está configurada para vaciar o eliminar, de la DPB 325, toda imagen marcada como no utilizada como referencia y no necesaria para la salida de entre las imágenes de la DPB 325. Una unidad de decodificación 305 está conectada a la memoria 320 y está configurada para decodificar la imagen actual.

El decodificador 300 de la figura 18 comprende además una unidad de determinación de número 330 conectada a la memoria 320 y preferiblemente también a la unidad de análisis 370 y al comparador 340. La unidad de determinación de número 330 está configurada para determinar el número de imágenes de la DPB 325 que se marcan como necesarias para la salida. El comparador 340 mencionado con anterioridad está configurado para comparar el número con el valor que se obtiene de al menos un elemento de sintaxis presente en la secuencia de bits. En esta realización, la unidad de salida 350 está configurada además para generar como salida la imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB 325 que están marcadas como necesarias para la salida, si el número es mayor que el valor. La unidad de marcado 360 está configurada además para marcar la imagen como no necesaria para la salida si el número es mayor que el valor.

En una realización preferida, la unidad de determinación de número 330 está configurada para determinar el número de imágenes, el comparador 340 está configurado para comparar el número, la unidad de salida 350 está configurada para generar como salida la imagen y la unidad de marcado 360 está configurada para marcar la imagen después de que la unidad de decodificación 305 haya decodificado la imagen actual.

La unidad de determinación del conjunto de imágenes de referencia 380 está conectada preferiblemente a la unidad de análisis 370 al objeto de recibir la información presente en la cabecera de segmento analizada y que se ha utilizado para determinar el RPS. La unidad de determinación del conjunto de imágenes de referencia 380 está conectada preferiblemente además a la unidad de marcado 360 al objeto de proporcionar el RPS o la información de las imágenes que figuran en el RPS a la unidad de marcado 360. El comparador 340 está conectado preferiblemente a la unidad de determinación de número 330 al objeto de recibir el número de imágenes determinado por medio de la unidad de determinación de número 330. De forma correspondiente, la unidad de salida 350 está conectada preferiblemente al comparador 340 al objeto de recibir información sobre si el número determinado es mayor que el valor del comparador 340. La unidad de marcado 360 está conectada preferiblemente a la unidad de salida 350 al objeto de recibir información sobre qué imagen se ha generado como salida por medio de la unidad de salida 350.

Como se ha indicado con anterioridad, el decodificador se puede definir alternativamente como un grupo de módulos de funciones, en el que los módulos de funciones se implementan como un programa de ordenador que se ejecuta en un procesador.

La figura 16 es un diagrama de bloques esquemático que ilustra un ejemplo de un decodificador 100 que comprende un procesador 110 y una memoria 120 asociada.

El programa de ordenador que reside en la memoria 120 puede estar organizado, por lo tanto, como unos módulos de funciones apropiados configurados para llevar a cabo, cuando se ejecutan por medio del procesador 110, al menos parte de las etapas y/o tareas descritas con anterioridad. En la figura 19 se ilustra un ejemplo de tales módulos de funciones. La figura 19 es, por lo tanto, un diagrama de bloques esquemático que ilustra un ejemplo de un decodificador 400 que comprende un grupo de módulos de funciones 410–440. Estos módulos comprenden un módulo de determinación de número 410 para determinar, después de que la imagen actual de una secuencia de bits que representa las imágenes de una secuencia de video se haya decodificado y almacenado en una DPB, un número de imágenes de la DPB que están marcadas como necesarias para la salida. El decodificador 400 comprende además un módulo de comparación 420 para comparar el número, preferiblemente tal y como se recibe del módulo de determinación de número 410, con un valor *sps_max_num_reorder_pics[HighesTid]*. Se utiliza un módulo de salida 430 del decodificador 400 para generar como salida una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor, de forma opcional, pero preferiblemente, según se determina por medio del módulo de comparación 420. El decodificador 400 comprende además un módulo de marcado 440 para marcar la imagen como no necesaria para la salida si el número es mayor que el valor, de forma opcional, pero preferiblemente, según se determina por medio del módulo de comparación 420.

La figura 20 es un diagrama de bloques esquemático que ilustra otro ejemplo de un decodificador 500 con un grupo de módulos de funciones 510–580. El decodificador 500 comprende un módulo de análisis 510 para analizar una cabecera de segmento de una imagen actual que se ha de decodificar de una secuencia de bits que representa las imágenes de una secuencia de video. Se utiliza un módulo de determinación del conjunto de imágenes de referencia 520 del decodificador 500 para determinar un RPS para la imagen actual a partir del análisis de la cabecera de segmento, preferiblemente tal y como se analiza por medio del módulo de análisis 510. El decodificador 500 comprende además un módulo de marcado 530 para marcar todas las imágenes de una DPB que no están presentes en el RPS, de forma opcional, pero preferiblemente, según se determina por medio del módulo de determinación del conjunto de imágenes de referencia 520, como no utilizadas como referencia, y un módulo de salida 540 para generar como salida cero, una o muchas imágenes, marcadas como necesarias para la salida, de entre las imágenes de la DPB. En una realización, el módulo de marcado 530 es además para marcar las cero, una o muchas imágenes como no necesarias para la salida. El decodificador 500 comprende además un módulo de

vaciado 550 para vaciar o eliminar, de la DPB, toda imagen marcada como no utilizada como referencia y no necesaria para la salida de entre las imágenes de la DPB, y un módulo de decodificación 560 para decodificar la imagen actual. El decodificador 500 comprende adicionalmente un módulo de determinación de número 560 para determinar un número de imágenes de la DPB que están marcadas como necesarias para la salida y un módulo de comparación 570 para comparar el número, de forma opcional, pero preferiblemente, según se determina por medio del módulo de determinación de número 560, con un valor que se obtiene de al menos un elemento de sintaxis presente en la secuencia de bits.

En una realización, el módulo de salida 540 es además para generar como salida una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor, de forma opcional, pero preferiblemente, según se determina por medio del módulo de comparación 570. El módulo de marcado 530 es además para marcar la imagen como no necesaria para la salida, si el número es mayor que el valor, de forma opcional, pero preferiblemente, según se determina por medio del módulo de comparación 570.

En una realización, el módulo de determinación de número 570 determina el número de imágenes, el módulo de comparación 580 compara el número, el módulo de salida 540 genera como salida la imagen y el módulo de marcado 530 marca dicha imagen después de que el módulo de decodificación 560 decodifica la imagen actual.

Las realizaciones del decodificador 400, 500, según se muestran en las figuras 19 y 20, se pueden utilizar opcionalmente también para llevar a cabo las diferentes realizaciones de implementación, tal y como se han descrito con anterioridad en la presente memoria, por ejemplo, haciendo referencia a las figuras 9, 10, 12–15.

En una realización, el programa de ordenador comprende un código de programa que, cuando se ejecuta por medio de un procesador 110, véase la figura 16, o por medio de un ordenador, hace que el procesador 110 u ordenador ejecuten las etapas, funciones, procedimientos y/o bloques descritos con anterioridad, y mostrados en las figuras 8–15.

El software o programa de ordenador se puede generar como un producto de programa de ordenador, el cual se dispone o almacena normalmente en un medio legible por ordenador. El medio legible por ordenador puede incluir uno o más dispositivos de memoria extraíbles o no extraíbles, incluyendo, aunque sin limitarse a los mismos, una ROM, una RAM, un disco compacto (CD, compact disc, por sus siglas en inglés), un disco versátil digital (DVD, digital versatile disc, por sus siglas en inglés), una memoria de bus serie universal (USB, universal serial bus, por sus siglas en inglés), un dispositivo de almacenamiento de disco duro (HDD, hard disk drive, por sus siglas en inglés), una memoria flash, o cualquier otro dispositivo de memoria convencional. El programa de ordenador se puede cargar, por lo tanto, en la memoria de funcionamiento de un ordenador o de un dispositivo de procesamiento equivalente para su ejecución por parte de los circuitos de procesamiento del mismo.

El decodificador 100, 200, 300, 400, 500, según se muestra en las figuras 16-20, es un decodificador que cumple preferiblemente con la norma HEVC. Sin embargo, se debe apreciar que las realizaciones no se limitan a la norma HEVC.

Según un aspecto, se proporciona un método que se lleva a cabo por medio de un codificador. En el método, el codificador determina el número de imágenes de la DPB que están marcadas como necesarias para la salida, y compara ese número con un valor que está representado por los elementos de sintaxis de la secuencia de bits. Si el número de imágenes de la DPB que están marcadas como necesarias para la salida es mayor que el valor que se obtiene de los elementos de sintaxis de la secuencia de bits, se lleva a cabo un proceso de marcado de salida modificado, o se marca como no necesaria para la salida la imagen de las imágenes de la DPB marcadas como necesarias para la salida que es la primera imagen en el orden de salida.

La figura 21 es un diagrama de flujo de un método ejecutado por medio de un codificador según una realización. El método comprende la determinación, en la etapa S60, y después de que se haya decodificado y almacenado en una DPB una imagen actual, de un número de imágenes de la DPB que están marcadas como necesarias para la salida. Una etapa siguiente S61 comprende la comparación de este número con un valor *sps_max_num_reorder_pics[HighestTid]*, en donde *HighestTid* especifica una capa más alta que se decodifica por medio del codificador de una secuencia de video. Si el número es mayor que el valor, el método avanza a la etapa S62, que comprende el marcado como no necesaria para la salida de una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida.

El método, tal y como se lleva a cabo en el codificador en la figura 21, es fundamentalmente similar al método correspondiente que se lleva a cabo en un decodificador correspondiente, véase la figura 8. Aunque una diferencia es que el codificador, por lo general, no genera como salida ninguna imagen, lo cual se lleva a cabo en el decodificador. Por lo tanto, la realización del método mostrado en la figura 21 carece preferiblemente de la etapa de salida S3 del método correspondiente, la cual se ejecuta por medio de un decodificador en la figura 8.

En una realización, cuando la imagen actual se ha decodificado, tienen lugar preferiblemente dos marcados, uno para salida y uno para referencia. Por tanto, en una realización, estos marcados se llevan a cabo preferiblemente

con anterioridad a la etapa S60, ya que esto afecta al número de imágenes que se marcan como necesarias para la salida. Si la imagen actual tiene un PicOutputFlag=1, la imagen actual preferiblemente se cuenta también.

En una realización, si el número no es mayor que el valor, según se determina en la comparación de la etapa S61, el método preferiblemente termina y no se marca ninguna imagen. Por tanto, en tal caso, la etapa S62 se omite y no se ejecuta, véase la línea de puntos a la derecha.

En una realización, las etapas S61–S62 de la figura 21 sólo se pueden ejecutar una vez, después de que se haya decodificado y almacenado en la DPB una imagen actual. De forma alternativa, el bucle formado por las etapas S61–S62, véase la línea de puntos a la izquierda, se puede ejecutar hasta que el número de imágenes de la DPB que están marcadas como necesarias para la salida deje de ser mayor que el valor *sps_max_num_reorder_pics[HighestTid]*. Cada vez que se ejecuta el bucle de las etapas S61–S62, el número de imágenes de la DPB que están marcadas como necesarias para la salida se reduce en una unidad por medio del marcado de la etapa S62.

La figura 22 es un diagrama de flujo de un método ejecutado por medio de un codificador según otra realización. El método comprende el marcado como no utilizadas como referencia, en la etapa S70, de todas las imágenes de la DPB que no están presentes en un RPS para una imagen actual de un conjunto de secuencia de video. El método comprende además el marcado como no necesarias para la salida, en la etapa S71, de cero, una o muchas imágenes, que están marcadas como necesarias para la salida, de entre las imágenes de la DPB. Toda imagen marcada como no utilizada como referencia y no necesaria para la salida de entre las imágenes de la DPB se vacía o elimina de la DPB en la etapa S72. La etapa S73 siguiente comprende la decodificación de la imagen actual. El método comprende además la determinación, en la etapa S74, de un número de imágenes de la DPB que están marcadas como necesarias para la salida, y la comparación, en la etapa S75, del número con un valor que se obtiene de al menos un elemento de sintaxis definido. Si este número es mayor que el valor, el método avanza a la etapa S76, que comprende el marcado como no necesaria para la salida de una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida. En una realización preferida, la determinación del número de imágenes en la etapa S74, la comparación del número en la etapa S75 y el marcado de la imagen en la etapa S76 se llevan a cabo después de la decodificación de la imagen actual en la etapa S73.

El método, tal y como se lleva a cabo en el codificador en la figura 22, es fundamentalmente similar al método correspondiente que se lleva a cabo en un decodificador correspondiente, véase la figura 11. Aunque una diferencia es que el codificador, por lo general, no genera como salida ninguna imagen, lo cual se lleva a cabo en el decodificador. Por lo tanto, la realización del método mostrado en la figura 22 carece preferiblemente de las etapas de salida S33 y S38 del método correspondiente, las cuales se ejecutan por medio de un decodificador en la figura 11. Además, en el método de la figura 22, el codificador no recibe ninguna secuencia de bits y, por lo tanto, no necesita analizar una cabecera de segmento al objeto de obtener información para determinar el RPS para la imagen actual. En claro contraste, el codificador genera por sí mismo y determina el RPS para la imagen actual. Por lo tanto, las etapas S30 y S31 del método correspondiente que se lleva a cabo por medio de un decodificador en la figura 11 no se ejecutan normalmente por un codificador.

En una realización, si el número no es mayor que el valor, según se determina en la comparación de la etapa S75, el método preferiblemente termina y no se marca ninguna imagen. Por tanto, en tal caso, la etapa S76 se omite y no se ejecuta, véase la línea de puntos a la derecha.

En una realización, las etapas S75–S76 sólo se pueden ejecutar una vez, después de que se haya decodificado y almacenado en la DPB una imagen actual. De forma alternativa, el bucle formado por las etapas S75–S76, véase la línea de puntos a la izquierda, se puede ejecutar hasta que el número de imágenes de la DPB que se marcan como necesarias para la salida deje de ser mayor que el valor. Cada vez que se ejecuta el bucle de las etapas S75–S76, el número de imágenes de la DPB que se marcan como necesarias para la salida se reduce en una unidad por medio del marcado de la etapa S76.

Las realizaciones analizadas en lo anterior en relación con los métodos llevados a cabo por medio de un decodificador también se pueden llevar a cabo por medio de un codificador.

Por ejemplo, un método de codificador puede comprender y/o un codificador puede estar configurado para la ejecución de las siguientes etapas en orden, según una realización:

1. Se codifica una imagen P.

2. Después de que se ha codificado la imagen P, el codificador determina el número de imágenes de la DPB que están marcadas como “necesarias para la salida” y compara ese número con un valor que se puede obtener de los elementos de sintaxis de la secuencia de bits. Se debe observar que esto se refiere a la DPB del codificador. El estatus de la DPB del codificador y del decodificador es el mismo. La comparación en el decodificador será exactamente la misma. La especificación HEVC especifica que el decodificador lo hará, pero que el codificador tendrá que tener un control de ello, es decir, el codificador hace lo que el decodificador hará.

3. Si el número de imágenes de la DPB que están marcadas como “necesarias para la salida” es mayor que el valor que se puede obtener de los elementos de sintaxis de la secuencia de bits, se marca como “no necesaria para la salida” la imagen de las imágenes de la DPB marcadas como “necesarias para la salida” que es la primera en el orden de salida. La imagen, opcionalmente, se puede generar como salida desde el codificador.

4. Se codifica una imagen Q siguiente.

Las etapas, funciones, procedimientos, módulos y/o bloques descritos con anterioridad en relación a las figuras 21 y 22 se pueden implementar en hardware por medio de la utilización de cualquier tecnología convencional, tal como tecnología de circuitos discretos o tecnología de circuitos integrados, incluyendo ambos, sistemas de circuitos electrónicos de propósito general y sistemas de circuitos específicos de aplicación.

Los ejemplos particulares incluyen uno o más procesadores de señales digitales configurados de forma adecuada y otros circuitos electrónicos conocidos, por ejemplo, puertas lógicas discretas interconectadas al objeto de llevar a cabo una función especializada, o ASICs.

Alternativamente, al menos algunas de las etapas, funciones, procedimientos, módulos y/o bloques descritos con anterioridad en relación a las figuras 21 y 22 se pueden implementar en software, tal como en un programa de ordenador que se ejecuta por medio de circuitos de procesamiento apropiados que incluyen uno o más procesadores.

El procesador es capaz de ejecutar instrucciones software contenidas en un programa de ordenador almacenado en un producto de programa de ordenador, por ejemplo, en forma de memorias. El respectivo producto de programa de ordenador puede ser una memoria que sea una combinación cualquiera de RAM y ROM. La memoria respectiva comprende un almacenamiento persistente que, por ejemplo, puede ser una cualquiera de entre memoria magnética, memoria óptica, memoria de estado sólido o incluso memoria montada de forma remota, o una combinación de las mismas.

El diagrama o diagramas de flujo presentados con anterioridad y mostrados en las figuras 21 y 22 se pueden considerar, por tanto, como un diagrama o diagramas de flujo de ordenador, cuando se ejecutan por medio de uno o más procesadores. Un aparato correspondiente se puede definir como un grupo de módulos de funciones, en donde cada etapa ejecutada por medio del procesador se corresponde con un módulo de funciones. En este caso, los módulos de funciones se implementan como un programa de ordenador que se ejecuta en el procesador.

Los ejemplos de circuitos de procesamiento incluyen, aunque no se limitan a los mismos, uno o más microprocesadores, uno o más procesadores DSPs, una o más CPUs, hardware de aceleración de video, y/o cualesquiera circuitos lógicos programables adecuados, tal como uno o más FPGAs, o uno o más PLCs.

Se ha de entender además que puede ser posible reutilizar las capacidades de procesamiento generales de cualquier dispositivo o unidad convencional en el que se implemente la tecnología propuesta. Puede ser posible también reutilizar software existente, por ejemplo, por medio de la reprogramación del software existente o por medio de la adición de nuevos componentes software.

Según un aspecto, se proporciona un codificador configurado para llevar a cabo el método. El codificador está configurado para la determinación, después de que se haya decodificado y almacenado en una DPB una imagen actual, de un número de imágenes de la DPB que están marcadas como necesarias para la salida. El codificador está configurado además para la comparación del número con un valor *sps_max_num_reorder_pics[HighestTid]*. El codificador está configurado también para marcar como no necesaria para la salida una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor.

El codificador 600 comprende, en una realización, un procesador 610 configurado para ejecutar las etapas del método descrito previamente en la presente memoria, véase la figura 21. El codificador 600 puede comprender además una memoria 620 conectada al procesador 610, véase la figura 23.

La figura 23 es un diagrama de bloques esquemático de un codificador 600 según una realización. El codificador 600 está configurado para codificar imágenes de una secuencia de video en una secuencia de bits 10. El codificador 600 comprende un procesador 610 y una memoria 620. La memoria 620 comprende una DPB 625. En una realización, el procesador 610 está configurado para la determinación, después de que se haya decodificado y almacenado en la DPB 625 la imagen actual, el número de imágenes de la DPB 625 que están marcadas como necesarias para la salida. El procesador 610 está configurado además para la comparación del número con el valor *sps_max_num_reorder_pics[HighestTid]*. El procesador 610 está configurado además para marcar como no necesaria para la salida la imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB 625 que están marcadas como necesarias para la salida, si el número es mayor que el valor.

En la figura 23, se ha ilustrado que el codificador 600 comprende un procesador 610. Este procesador 610 se podría implementar como un único procesador o como múltiples procesadores, por ejemplo, como un circuito de procesamiento.

La figura 23 ilustra de esta forma una implementación en ordenador del codificador 600. En este ejemplo particular, al menos algunas de las etapas, funciones, procedimientos, módulos y/o bloques descritos con anterioridad se implementan en un programa de ordenador, el cual se carga en la memoria 620 para su ejecución por parte del procesador 610. El procesador 610 y la memoria 620 están interconectados entre sí al objeto de hacer posible la ejecución normal del software. También se puede interconectar un dispositivo de entrada/salida opcional (no mostrado) al procesador 610 y/o a la memoria 620 para hacer posible la entrada de imágenes que se han de codificar y la salida de la secuencia de bits 10.

Otro aspecto de las realizaciones se refiere a un codificador configurado para marcar todas las imágenes de una DPB que no están presentes en un RPS para una imagen actual como no utilizadas como referencia. El codificador está configurado además para marcar como no necesarias para la salida cero, una o muchas imágenes, marcadas como necesarias para la salida, de entre las imágenes de la DPB, y para vaciar o eliminar, de la DPB, toda imagen marcada como no utilizada como referencia y como no necesaria para la salida de entre las imágenes de la DPB. El codificador está configurado además para decodificar la imagen actual y para determinar un número de imágenes de la DPB que están marcadas como necesarias para la salida. El codificador está configurado adicionalmente para comparar el número con un valor que se obtiene de al menos un elemento de sintaxis definido y para marcar como no necesaria para la salida una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor. En una realización, el codificador está configurado para determinar el número de imágenes, comparar el número y marcar la imagen después de decodificar la imagen actual.

El codificador 600 comprende, en una realización, un procesador 610 configurado para ejecutar las etapas del método descrito previamente en la presente memoria, véase la figura 22. El decodificador 600 puede comprender además una memoria 620 conectada al procesador 610, véase la figura 23.

La figura 23 es un diagrama de bloques esquemático de un codificador 600 según una realización. El codificador 600 está configurado para codificar imágenes de una secuencia de video en una secuencia de bits 10. El codificador 600 comprende un procesador 610 y una memoria 620. La memoria 620 comprende una DPB 625. En una realización, el procesador 610 está configurado para marcar todas las imágenes de la DPB 625 que no están presentes en el RPS para la imagen actual de la secuencia de video como no utilizadas como referencia. El procesador 610 está configurado además para marcar cero, una o muchas imágenes, marcadas como necesarias para la salida, como no necesarias para la salida, de entre las imágenes de la DPB 625. El procesador 610 está configurado además para vaciar o eliminar, de la DPB 625, toda imagen marcada como no utilizada como referencia y como no necesaria para la salida de entre las imágenes de la DPB 625. En esta realización, el procesador 610 está configurado además para decodificar la imagen actual, y para determinar un número de imágenes de la DPB 625 que están marcadas como necesarias para la salida. El procesador 610 está configurado también para comparar el número con un valor que se obtiene de al menos un elemento de sintaxis definido. El procesador 610 está configurado adicionalmente para marcar como no necesaria para la salida una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB 625 que están marcadas como necesarias para la salida, si el número es mayor que el valor. En esta realización, el procesador 610 está configurado para determinar el número de imágenes, comparar el número y marcar la imagen después de decodificar la imagen actual.

La figura 24 es un diagrama de bloques esquemático de otro ejemplo de implementación del codificador 700. Este ejemplo es especialmente apropiado para una implementación hardware del codificador 700. El codificador 700 comprende una unidad de determinación de número 730 conectada a una memoria 720 que comprende la DPB 725. La unidad de determinación de número 730 está configurada para determinar, después de que la imagen actual de la secuencia de video se haya decodificado y almacenado en la DPB 725, el número de imágenes de la DPB 725 que están marcadas como necesarias para la salida. El codificador 700 comprende además un comparador 740 conectado a la unidad de determinación de número 730 y preferiblemente a la memoria 720. El comparador 740 está configurado para comparar el número con el valor *sps_max_num_reorder_pics[HighestTid]*. Una unidad de marcado 760 del codificador está conectada a la memoria 720 y está configurada para marcar como no necesaria para la salida la imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB 725 que están marcadas como necesarias para la salida, si el número es mayor que el valor.

El comparador 740 está conectado preferiblemente a la unidad de determinación de número 730 al objeto de recibir el número de imágenes determinado por medio de la unidad de determinación de número 730.

La figura 25 es un diagrama de bloques esquemático de otro ejemplo más de implementación del codificador 800, especialmente apropiado para una implementación hardware. El codificador 800 comprende una unidad de marcado 860 conectada a la memoria 820 que comprende la DPB 825. La unidad de marcado 860 está configurada para i) marcar todas las imágenes de la DPB 825 que no están presentes en el RPS para la imagen actual de la secuencia de video como no utilizadas como referencia y ii) para marcar como no necesarias para la salida cero, una o muchas imágenes, marcadas como necesarias para la salida, de entre las imágenes de la DPB 825. Una unidad de vaciado de imágenes 890 está conectada a la memoria 820 y está configurada para vaciar o eliminar, de la DPB 825, toda imagen marcada como no utilizada como referencia y no necesaria para la salida de entre las imágenes de la DPB 825. El codificador 800 comprende además una unidad de decodificación 805 conectada a la memoria 820 y configurada para decodificar la imagen actual. Una unidad de determinación de número 830 del codificador 800 está

conectada a la memoria 830, y está configurada para determinar el número de imágenes de la DPB 825 que están marcadas como necesarias para la salida. El codificador 800 comprende además un comparador 840 conectado a la unidad de determinación de número 830 y preferiblemente a la memoria 820. El comparador 840 está configurado para comparar el número con el valor que se obtiene de al menos un elemento de sintaxis definido. En esta realización, la unidad de marcado 860 está configurada además para marcar como no necesaria para la salida la imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB 825 que están marcadas como necesarias para la salida, si el número es mayor que el valor. En una realización preferida, la unidad de determinación de número 830 está configurada para determinar el número de imágenes, el comparador 840 está configurado para comparar el número y la unidad de marcado 840 está configurada para marcar la imagen después de que la unidad de decodificación 805 haya decodificado la imagen actual.

El comparador 840 está conectado preferiblemente a la unidad de determinación de número 830 al objeto de recibir el número de imágenes determinado por medio de la unidad de determinación de número 830.

Como se ha indicado con anterioridad, el codificador se puede definir alternativamente como un grupo de módulos de funciones, en el que los módulos de funciones se implementan como un programa de ordenador que se ejecuta en un procesador.

La figura 23 es un diagrama de bloques esquemático que ilustra un ejemplo de un codificador 600 que comprende un procesador 610 y una memoria 620 asociada.

El programa de ordenador que reside en la memoria 620 puede estar organizado, por lo tanto, como unos módulos de funciones apropiados configurados para llevar a cabo, cuando se ejecutan por medio del procesador 610, al menos parte de las etapas y/o tareas descritas con anterioridad. En la figura 26 se ilustra un ejemplo de tales módulos de funciones. La figura 26 es, por lo tanto, un diagrama de bloques esquemático que ilustra un ejemplo de un codificador 900 que comprende un grupo de módulos de funciones 910–930. Estos módulos comprenden un módulo de determinación de número 910 para determinar, después de que una imagen actual de una secuencia de bits que representa las imágenes de una secuencia de video se haya decodificado y almacenado en una DPB, un número de imágenes de la DPB que están marcadas como necesarias para la salida. El codificador 900 comprende además un módulo de comparación 920 para comparar el número, de forma opcional, pero preferiblemente, según se determina por medio del módulo de determinación de número 910, con un valor *sps_max_num_reorder_pics[HighesTid]*. El codificador 900 comprende además un módulo de marcado 930 para marcar como no necesaria para la salida una imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor, de forma opcional, pero preferiblemente, según se determina por medio del módulo de comparación 920.

La figura 27 es un diagrama de bloques esquemático que ilustra otro ejemplo de un codificador 1000 con un grupo de módulos de funciones 1000–1050. El codificador 1000 comprende un módulo de marcado 1010 para marcar todas las imágenes de una DPB que no están presentes en un conjunto RPS para una imagen actual de una secuencia de video como no utilizadas como referencia, y para marcar como no necesarias para la salida cero, una o muchas imágenes, marcadas como necesarias para la salida, de entre las imágenes de la DPB. El codificador 1000 comprende además un módulo de vaciado 1020 para vaciar o eliminar, de la DPB, toda imagen marcada como no utilizada como referencia y no necesaria para la salida de entre las imágenes de la DPB, y un módulo de decodificación 1030 para decodificar la imagen actual. El codificador 1000 comprende adicionalmente un módulo de determinación de número 1040 para determinar un número de imágenes de la DPB que están marcadas como necesarias para la salida y un módulo de comparación 1050 para comparar el número, de forma opcional, pero preferiblemente, según se determina por medio del módulo de determinación de número 1040, con un valor que se obtiene de al menos un elemento de sintaxis presente en la secuencia de bits.

En una realización, el módulo de marcado 1010 es además para marcar como no necesaria para la salida la imagen, que es una primera imagen en el orden de salida, de entre las imágenes de la DPB que están marcadas como necesarias para la salida, si el número es mayor que el valor, de forma opcional, pero preferiblemente, según se determina por medio del módulo de comparación 1050.

En una realización, el módulo de determinación de número 1040 determina el número de imágenes, el módulo de comparación 1050 compara el número y el módulo de marcado 1010 marca dicha imagen después de que el módulo de decodificación 1030 decodifica la imagen actual.

En una realización, el programa de ordenador comprende un código de programa que, cuando se ejecuta por medio de un procesador 610, véase la figura 23, o por medio de un ordenador, hace que el procesador 610 u ordenador ejecuten las etapas, funciones, procedimientos y/o bloques descritos con anterioridad, y mostrados en las figuras 21 o 22.

El software o programa de ordenador se puede generar como un producto de programa de ordenador, el cual se dispone o almacena normalmente en un medio legible por ordenador. El medio legible por ordenador puede incluir uno o más dispositivos de memoria extraíbles o no extraíbles, incluyendo, aunque sin limitarse a los mismos, una ROM, una RAM, un CD, un DVD, una memoria USB, un dispositivo de almacenamiento HDD, una memoria flash, o

cualquier otro dispositivo de memoria convencional. El programa de ordenador se puede cargar, por lo tanto, en la memoria de funcionamiento de un ordenador o de un dispositivo de procesamiento equivalente para su ejecución por parte de los circuitos de procesamiento del mismo.

El codificador 600, 700, 800, 900, 1000, según se muestra en las figuras 23–27, es un codificador que cumple preferiblemente con la norma HEVC. Sin embargo, se debe apreciar que las realizaciones no se limitan a la norma HEVC.

El decodificador 100, 200, 300, 400, 500 de cualquiera de las figuras 16–20 y el codificador 600, 700, 800, 900, 1000 de cualquiera de las figuras 23–27 se pueden implementar, por ejemplo, en un terminal móvil. Por ejemplo, el decodificador 100, 200, 300, 400, 500 puede estar ubicado, por ejemplo, en un receptor de una cámara de video o de cualquier otro dispositivo de visualización de una emisión de video en continuo. El codificador 600, 700, 800, 900, 1000 puede estar ubicado, por ejemplo, en un transmisor de una cámara de video, por ejemplo, en un dispositivo móvil.

La figura 28 es una diagrama de bloques esquemático de un terminal móvil 1100 según una realización. El terminal móvil 1100 comprende un decodificador 100, 200, 300, 400, 500 según las realizaciones y/o un codificador 600, 700, 800, 900, 1000 según las realizaciones, tales como el decodificador que se muestra en cualquiera de las figuras 16–20 y/o el codificador que se muestra en cualquiera de las figuras 23–27.

El terminal móvil 1100 comprende preferiblemente además una unidad de entrada y salida (I/O, input and output, por sus siglas en inglés) 1110 para hacer posible la comunicación, por lo general una comunicación inalámbrica, pero de forma alternativa, o adicionalmente, una comunicación por cable, con unidades externas. La unidad I/O 1110 se podría implementar como un transmisor y un receptor, o un transceptor, de comunicación inalámbrica. De forma alternativa, la unidad I/O 1110 podría ser una unidad o puerto I/O general 1110 capaz de llevar a cabo una comunicación por cable. Si el terminal móvil 1100 se implementa con un codificador 600, 700, 800, 900, 1000, la unidad I/O 1110 está configurada preferiblemente para transmitir o generar como salida una secuencia de bits que representa una secuencia de video codificada, tal y como se genera por medio del codificador 600, 700, 800, 900, 1000. De forma correspondiente, si el terminal móvil 1100 comprende un decodificador 100, 200, 300, 400, 500, la unidad I/O 1110 está configurada preferiblemente para recibir o tomar como entrada una secuencia de bits que representa una secuencia de video codificada.

El terminal móvil 1100 comprende una memoria 1120 configurada para almacenar imágenes codificadas de una secuencia de video codificada. Estas imágenes codificadas se pueden haber generado por medio del propio terminal móvil 1100. En tal caso, el terminal móvil 1100 comprende preferiblemente un motor o grabador multimedia (no mostrado) junto con un codificador 600, 700, 800, 900, 1000 conectado. De forma alternativa, la secuencia de video codificada se genera por medio de algún otro dispositivo y se transmite al terminal móvil 1100.

Las imágenes codificadas se llevan desde la memoria 1120 hasta un decodificador 100, 200, 300, 400, 500. El decodificador 100, 200, 300, 400, 500 decodifica a continuación las imágenes codificadas y las transforma en imágenes decodificadas. Las imágenes decodificadas se proporcionan a un reproductor multimedia 1130 que está configurado para traducir las imágenes decodificadas de la secuencia de video en datos de video que se pueden visualizar en un monitor o pantalla 1140 del terminal móvil 1100 o conectado al mismo.

En la figura 28, se ha ilustrado que el terminal móvil 1100 comprende tanto el decodificador 100, 200, 300, 400, 500 como el reproductor multimedia 1130, con el decodificador 100, 200, 300, 400, 500 implementado como una parte del reproductor multimedia 1130. Sin embargo, esto se ha de considerar meramente como un ejemplo ilustrativo y no limitativo de una realización de implementación del terminal móvil 1100. También son posibles implementaciones distribuidas, en las que el decodificador 100, 200, 300, 400, 500 y el reproductor multimedia 1130 se proporcionan en dos dispositivos separados físicamente dentro del alcance del terminal móvil 1100, tal y como se utiliza en la presente memoria. El monitor 1140 se podría proporcionar también como un dispositivo independiente conectado al terminal móvil 1100 en el que tiene lugar el procesamiento real de los datos.

El terminal móvil 1100 puede ser cualquier dispositivo que tenga funciones de decodificación multimedia que opere sobre una secuencia de video codificada de imágenes codificadas para de esta forma decodificar las imágenes y hacer que los datos de video estén disponibles. Ejemplos no limitativos de tales terminales móviles 1100 incluyen teléfonos móviles y otros reproductores multimedia portátiles, ordenadores, decodificadores, consolas de juegos, etc.

El codificador 600, 700, 800, 900, 1000 de las realizaciones, tal como el que se muestra en cualquiera de las figuras 23–27, y/o el decodificador 100, 200, 300, 400, 500 de las realizaciones, tal como el que se muestra en cualquiera de las figuras 16–20, se puede implementar en un nodo de red 2, tal y como se muestra en la figura 29.

Como se ilustra en la figura 29, el codificador 600, 700, 800, 900, 1000 y/o el decodificador 100, 200, 300, 400, 500 se puede implementar en un nodo de red 2 en una red de comunicación 1, entre una unidad emisora 3 y una unidad receptora 4. Tal nodo de red 2 puede ser un dispositivo para la conversión de video entre, por ejemplo, diferentes resoluciones de video, frecuencias de trama, calidades, velocidades de bits y normas de codificación. El nodo de red 2 puede adoptar la forma de una estación de base de radiocomunicaciones, un nodo-B o de cualquier otro nodo de red en una red de comunicación 1, tal como un red de radiocomunicaciones.

El codificador y/o el decodificador de las realizaciones se puede proporcionar además sobre cualquier elemento que opere sobre una secuencia de bits, tal como un elemento de red de gestión multimedia (MANE, media aware network element, por sus siglas en inglés).

Las realizaciones no están limitadas a la norma HEVC, sino que se pueden aplicar a cualquier extensión de la norma HEVC, tal como a una extensión escalable o a una extensión multivisión, o a un códec de video diferente.

Las realizaciones descritas con anterioridad se han de entender como unos cuantos ejemplos ilustrativos de la presente invención.

Sin embargo, el alcance de la presente invención está definido por medio de las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un método ejecutado por medio de un decodificador (100, 200, 400), dicho método comprende:

determinar (S1), después de que una imagen actual de una secuencia de bits se haya decodificado y almacenado en una memoria intermedia de imágenes decodificadas (125, 225), un número de imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que están marcadas como necesarias para la salida, perteneciendo dicha imagen actual a una capa distinta de la capa más alta que se decodifica por medio de dicho decodificador (100, 200, 400) de una secuencia de video multi-capa que comprende múltiples capas de imágenes;

obtener un valor sps_max_num_reorder_pics[HighestTid] de un elemento de sintaxis asociado con dicha capa más alta, en el que el *HighestTid* especifica dicha capa más alta, y el elemento de sintaxis se recupera de un conjunto de parámetros de secuencia en la secuencia de bits, y cada capa i de dichas múltiples capas tiene un elemento de sintaxis asociado que define un respectivo valor *sps_max_num_reorder_pics*[i], y dicho valor *sps_max_num_reorder_pics*[i] indica un máximo número de imágenes permitido que pueden preceder a cualquier imagen en dicha secuencia de video multi-capa en el orden de decodificación y que van detrás de esa imagen en el orden de salida cuando *HighestTid* es igual a i;

generar como salida (S3) una imagen, que es una primera imagen en el orden de salida de entre dichas imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que están marcadas como necesarias para la salida, si dicho número de imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que están marcadas como necesarias para la salida es mayor que dicho valor *sps_max_num_reorder_pics[HighestTid]*; y

marcar (S4) dicha imagen como no necesaria para la salida, si dicho número de imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que están marcadas como necesarias para la salida es mayor que dicho valor *sps_max_num_reorder_pics[HighestTid]*.

2. El método según la reivindicación 1, que comprende además:

analizar (S10) una cabecera de segmento de dicha imagen actual que se ha de decodificar de dicha secuencia de video multi-capa;

determinar (S11) un conjunto de imágenes de referencia para dicha imagen actual a partir del análisis de dicha cabecera de segmento;

marcar (S12) todas las imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que no están presentes en dicho conjunto de imágenes de referencia como no utilizadas como referencia;

generar como salida (S13) cero, una o muchas imágenes, las cuales están marcadas como necesarias para la salida, de entre dichas imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225), y marcar dichas cero, una o muchas imágenes como no necesarias para la salida;

vaciar (S14), de dicha memoria intermedia de imágenes decodificadas (125, 225), toda imagen marcada como no utilizada como referencia y como no necesaria para la salida de entre dichas imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225); y

decodificar (S15) dicha imagen actual; en el que la determinación (S1) de dicho número de imágenes, la generación como salida (S3) de dicha imagen y el marcado (S4) de dicha imagen se llevan a cabo después de la decodificación (S15) de dicha imagen actual.

3. Un decodificador (100, 200) configurado para:

determinar, después de que una imagen actual de una secuencia de bits (10) que representa las imágenes de una secuencia de video multi-capa que comprende múltiples capas de imágenes se haya decodificado y almacenado en una memoria intermedia de imágenes decodificadas (125, 225), un número de imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que están marcadas como necesarias para la salida, perteneciendo dicha imagen actual a una capa distinta de la capa más alta que se decodifica por medio de dicho decodificador (100, 200) de dicha secuencia de video multi-capa;

obtener un valor *sps_max_num_reorder_pics[HighestTid]* de un elemento de sintaxis asociado con dicha capa más alta, en el que el *HighestTid* especifica dicha capa más alta, y el elemento de sintaxis se recupera de un conjunto de parámetros de secuencia en la secuencia de bits, y cada capa i de dichas múltiples capas tiene un elemento de sintaxis asociado que define un respectivo valor *sps_max_num_reorder_pics*[i], y dicho valor *sps_max_num_reorder_pics*[i] indica un máximo número de imágenes permitido que pueden preceder a cualquier imagen en dicha secuencia de video multi-capa en el orden de decodificación y que van detrás de esa imagen en el orden de salida cuando *HighestTid* es igual a i;

generar como salida una imagen, que es una primera imagen en el orden de salida de entre dichas imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que están marcadas como necesarias para la

salida, si dicho número de imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que están marcadas como necesarias para la salida es mayor que dicho valor *sps_max_num_reorder_pics[HighestTid]*; y

marcar dicha imagen como no necesaria para la salida, si dicho número de imágenes de dicha memoria intermedia de imágenes decodificadas (125, 225) que están marcadas como necesarias para la salida es mayor que dicho valor *sps_max_num_reorder_pics[HighestTid]*.

4. El decodificador según la reivindicación 3, en el que dicho decodificador (100, 200, 300, 400, 500) es un decodificador (100, 200, 300, 400, 500) que cumple con la norma de codificación de video de alta eficiencia, HEVC.

5. Un método ejecutado por medio de un codificador (600, 700, 900), dicho método comprende:

determinar (S60), después de que una imagen actual se haya decodificado y almacenado en una memoria intermedia de imágenes decodificadas (625, 725), un número de imágenes de dicha memoria intermedia de imágenes decodificadas (625, 725) que están marcadas como necesarias para la salida, perteneciendo dicha imagen actual a una capa distinta de la capa más alta que se decodifica por medio de dicho codificador (600, 700, 900) de una secuencia de video multi-capa que comprende múltiples capas de imágenes;

seleccionar un valor *sps_max_num_reorder_pics[HighestTid]* para un elemento de sintaxis en un conjunto de parámetros de secuencia, en el que *HighestTid* especifica dicha capa más alta, cada capa i de dichas múltiples capas tiene un valor asociado *sps_max_num_reorder_pics*[i], y dicho valor *sps_max_num_reorder_pics*[i] indica un máximo número de imágenes permitido que pueden preceder a cualquier imagen en dicha secuencia de video multi-capa en el orden de decodificación y que van detrás de esa imagen en el orden de salida cuando *HighestTid* es igual a i;

marcar (S62) como no necesaria para la salida una imagen, que es una primera imagen en el orden de salida de entre dichas imágenes de dicha memoria intermedia de imágenes decodificadas (625, 725) que están marcadas como necesarias para la salida, si dicho número de imágenes de dicha memoria intermedia de imágenes decodificadas (625, 725) que están marcadas como necesarias para la salida es mayor que dicho valor *sps_max_num_reorder_pics[HighestTid]*.

6. Un codificador (600, 700) configurado para:

determinar, después de que una imagen actual se haya decodificado y almacenado en una memoria intermedia de imágenes decodificadas (625, 725), un número de imágenes de dicha memoria intermedia de imágenes decodificadas (625, 725) que están marcadas como necesarias para la salida, perteneciendo dicha imagen actual a una capa distinta de la capa más alta que se decodifica por medio de dicho decodificador (600, 700) de una secuencia de video multi-capa que comprende múltiples capas de imágenes;

seleccionar un valor *sps_max_num_reorder_pics[HighestTid]* para un elemento de sintaxis en un conjunto de parámetros de secuencia, en el que *HighestTid* especifica dicha capa más alta, cada capa i de dichas múltiples capas tiene un valor asociado *sps_max_num_reorder_pics*[i], y dicho valor *sps_max_num_reorder_pics*[i] indica un máximo número de imágenes permitido que pueden preceder a cualquier imagen en dicha secuencia de video multi-capa en el orden de decodificación y que van detrás de esa imagen en el orden de salida cuando *HighestTid* es igual a i;

marcar como no necesaria para la salida una imagen, que es una primera imagen en el orden de salida de entre dichas imágenes de dicha memoria intermedia de imágenes decodificadas (625, 725) que están marcadas como necesarias para la salida, si dicho número de imágenes de dicha memoria intermedia de imágenes decodificadas (625, 725) que están marcadas como necesarias para la salida es mayor que dicho valor *sps_max_num_reorder_pics[HighestTid]*.

7. El codificador según la reivindicación 6, en el que dicho codificador (600, 700, 800, 900, 1000) es un codificador (600, 700, 800, 900, 1000) que cumple con la norma de codificación de video de alta eficiencia, HEVC.

8. Un terminal móvil (1100) que comprende un decodificador (100, 200, 300, 400, 500) según cualquiera de las reivindicaciones 3–4 y/o un codificador (600, 700, 800, 900, 1000) según cualquiera de las reivindicaciones 6–7.

9. Un nodo de red (2) que comprende un decodificador (100, 200, 300, 400, 500) según cualquiera de las reivindicaciones 3–4 y/o un codificador (600, 700, 800, 900, 1000) según cualquiera de las reivindicaciones 6–7.

Fig. 1

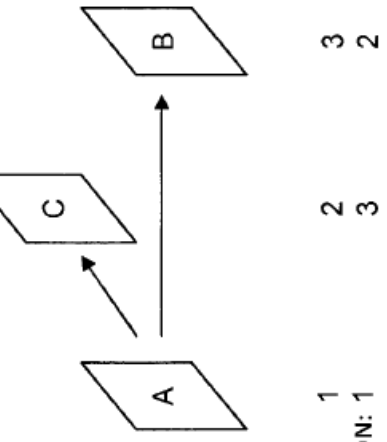

Fig. 2

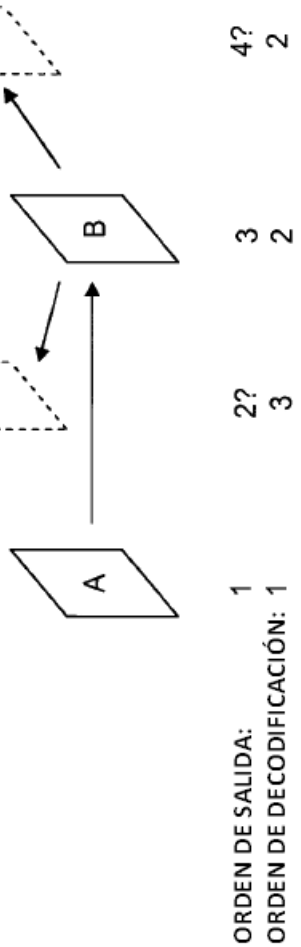

Fig. 4

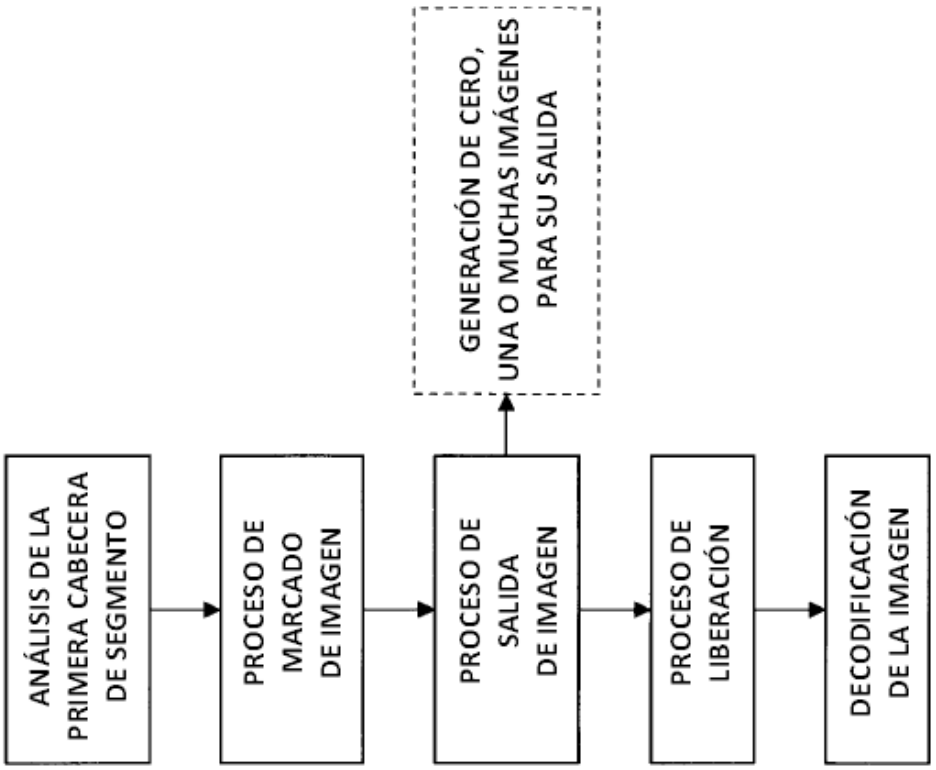
ANÁLISIS DE LA PRIMERA CABECERA DE SEGMENTO
PROCESO DE MARCADO DE IMAGEN
PROCESO DE SALIDA DE IMAGEN
GENERACIÓN DE CERO, UNA O MUCHAS IMÁGENES PARA SU SALIDA
PROCESO DE LIBERACIÓN
DECODIFICACIÓN DE LA IMAGEN

Fig. 5

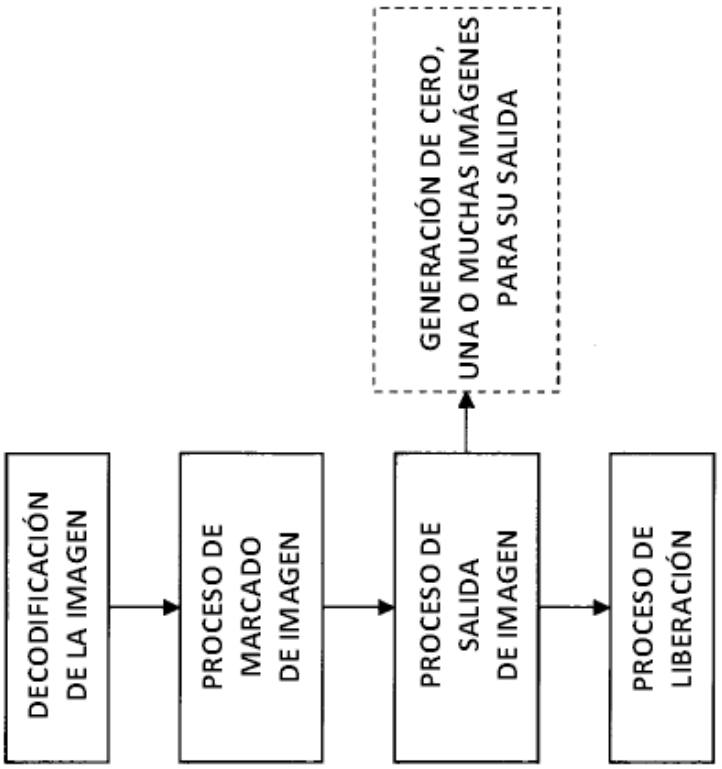
DECODIFICACIÓN DE LA IMAGEN
PROCESO DE MARCADO DE IMAGEN
PROCESO DE SALIDA DE IMAGEN
GENERACIÓN DE CERO, UNA O MUCHAS IMÁGENES PARA SU SALIDA
PROCESO DE LIBERACIÓN

Fig. 3

MAX_NUM_REORDER_PICS[2]=2
MAX_NUM_REORDER_PICS[1]=1
MAX_NUM_REORDER_PICS[0]=0
A
D
C
E
B
CAPA TEMPORAL: 0 2 1 2 0
ORDEN DE SALIDA: 1 2 3 4 5
ORDEN DE DECODIFICACIÓN: 1 4 3 5 2

Fig. 6

MAX_NUM_REORDER_PICS[0]=0
RAP
A
B
C
ORDEN DE SALIDA: 1 2 3
ORDEN DE DECODIFICACIÓN: 1 2 3

Fig. 7

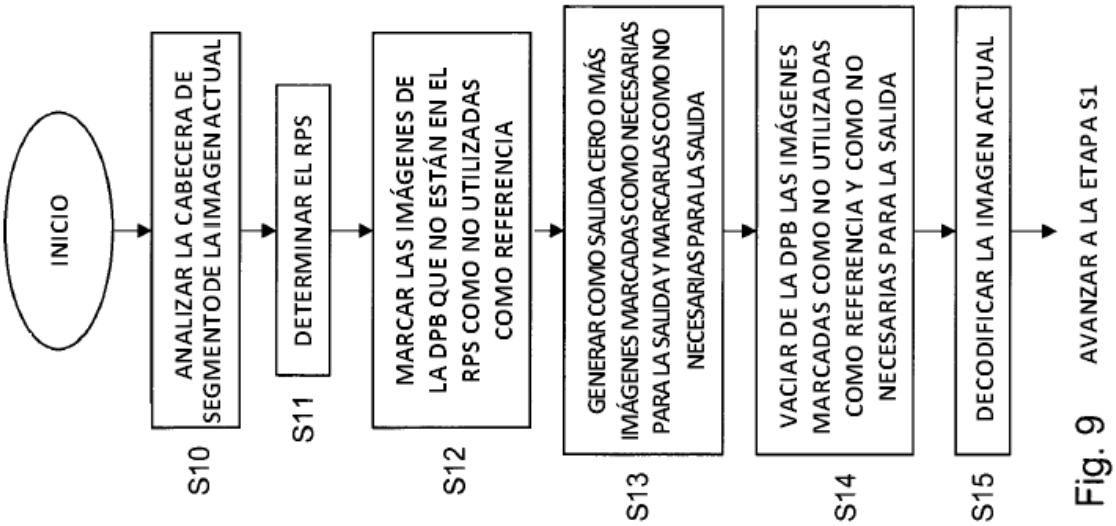
INICIO
ANALIZAR LA CABECERA DE SEGMENTO DE LA IMAGEN ACTUAL
S10
DETERMINAR EL RPS
S11
MARCAR LAS IMÁGENES DE LA DPB QUE NO ESTÁN EN EL RPS COMO NO UTILIZADAS COMO REFERENCIA
S12
GENERAR COMO SALIDA CERO O MÁS IMÁGENES MARCADAS COMO NECESARIAS PARA LA SALIDA Y MARCARLAS COMO NO NECESARIAS PARA LA SALIDA
S13
VACIAR DE LA DPB LAS IMÁGENES MARCADAS COMO NO UTILIZADAS COMO REFERENCIA Y COMO NO NECESARIAS PARA LA SALIDA
S14
DECODIFICAR LA IMAGEN ACTUAL
S15
AVANZAR A LA ETAPA S1

Fig. 9

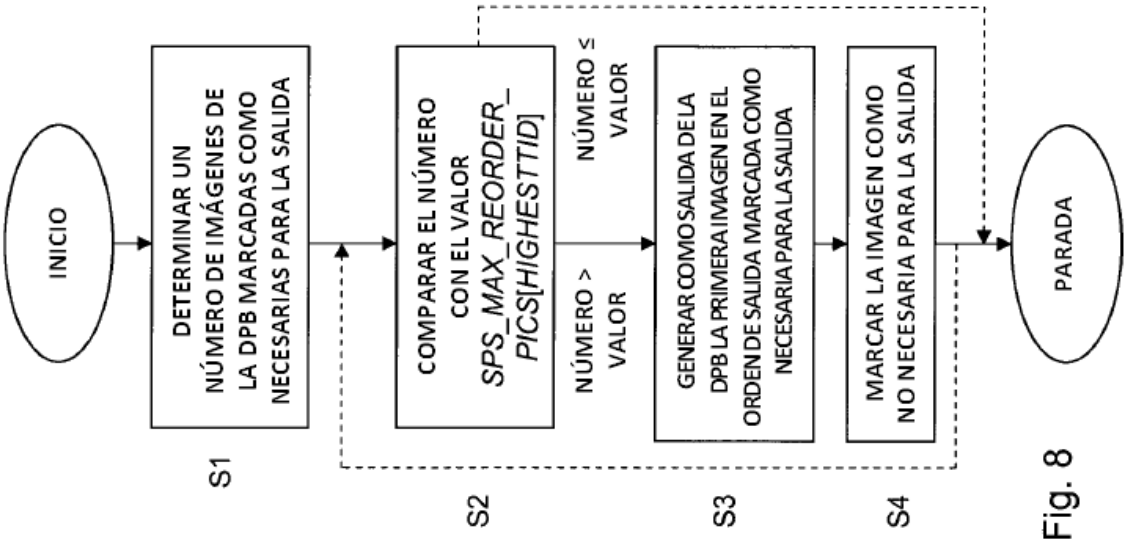
INICIO
DETERMINAR UN NÚMERO DE IMÁGENES DE LA DPB MARCADAS COMO NECESARIAS PARA LA SALIDA
S1
COMPARAR EL NÚMERO CON EL VALOR SPS_MAX_REORDER_PICS[HIGHESTTID]
S2
NÚMERO > VALOR
NÚMERO ≤ VALOR
GENERAR COMO SALIDA DE LA DPB LA PRIMERA IMAGEN EN EL ORDEN DE SALIDA MARCADA COMO NECESARIA PARA LA SALIDA
S3
MARCAR LA IMAGEN COMO NO NECESARIA PARA LA SALIDA
S4
PARADA

Fig. 8

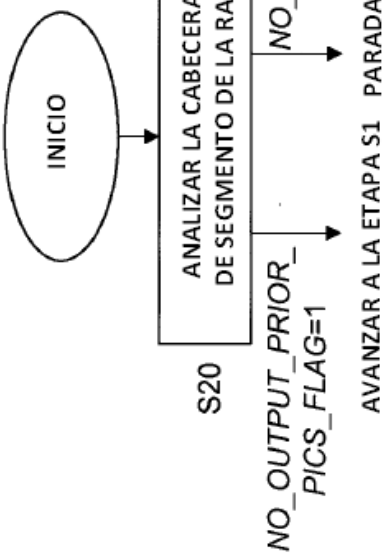

Fig. 10

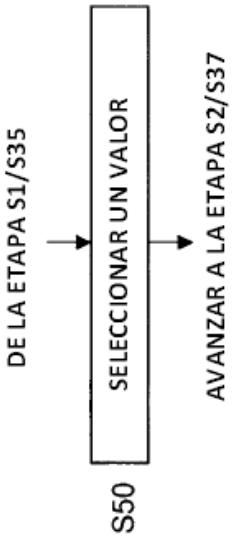

Fig. 13

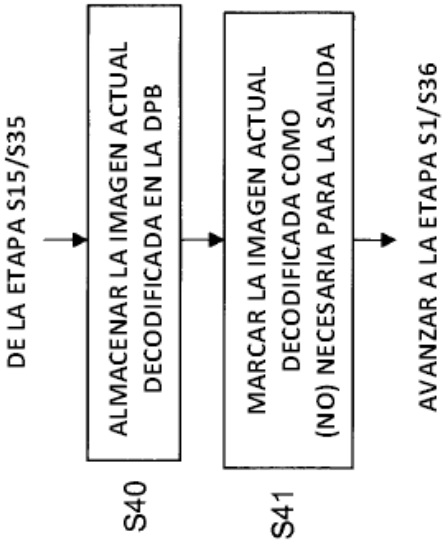

Fig. 12

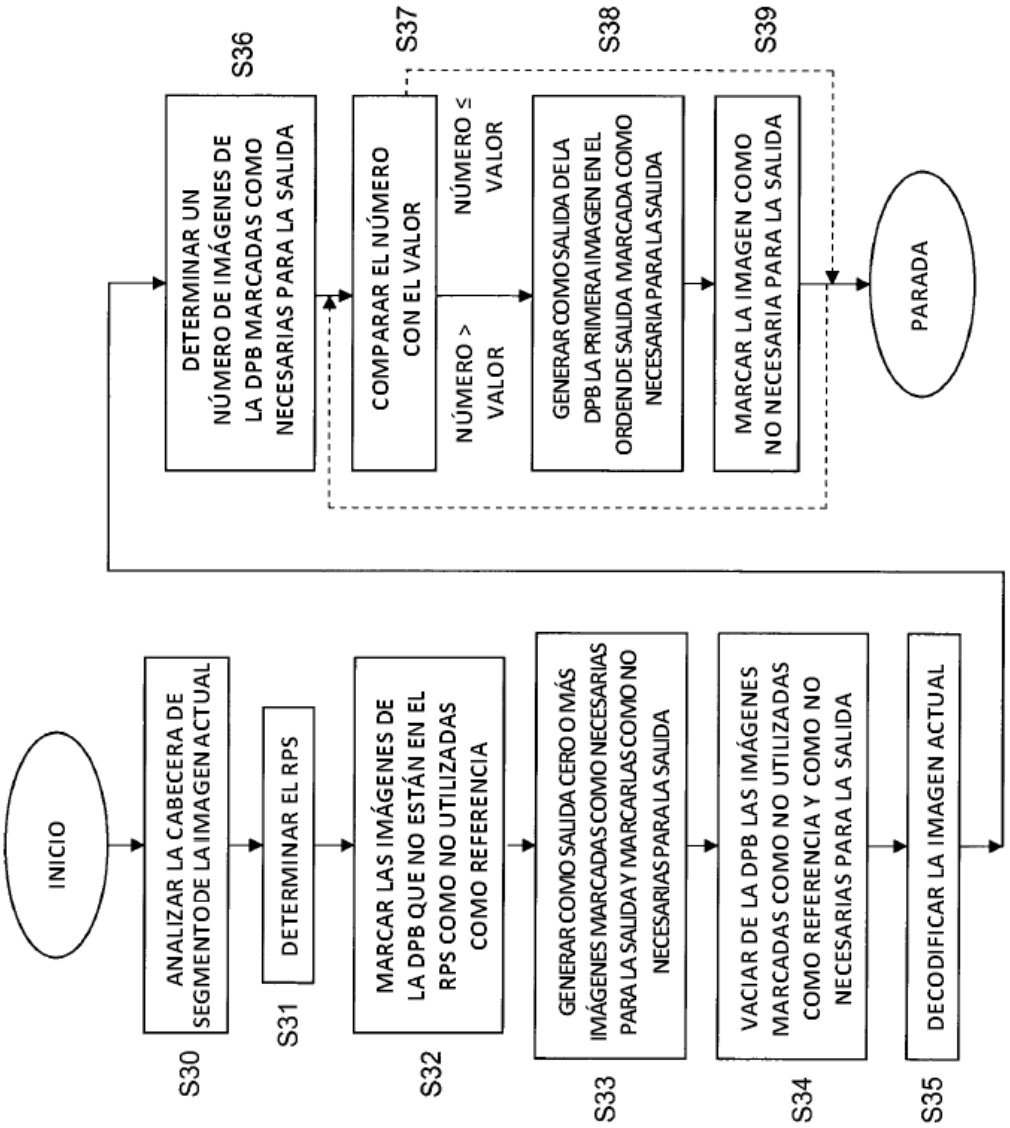
INICIO
ANALIZAR LA CABECERA DE SEGMENTO DE LA IMAGEN ACTUAL
S30
DETERMINAR EL RPS
S31
MARCAR LAS IMÁGENES DE LA DPB QUE NO ESTÁN EN EL RPS COMO NO UTILIZADAS COMO REFERENCIA
S32
GENERAR COMO SALIDA CERO O MÁS IMÁGENES MARCADAS COMO NECESARIAS PARA LA SALIDA Y MARCARLAS COMO NO NECESARIAS PARA LA SALIDA
S33
VACIAR DE LA DPB LAS IMÁGENES MARCADAS COMO NO UTILIZADAS COMO REFERENCIA Y COMO NO NECESARIAS PARA LA SALIDA
S34
DECODIFICAR LA IMAGEN ACTUAL
S35
DETERMINAR UN NÚMERO DE IMÁGENES DE LA DPB MARCADAS COMO NECESARIAS PARA LA SALIDA
S36
COMPARAR EL NÚMERO CON EL VALOR
S37
NÚMERO > VALOR
NÚMERO ≤ VALOR
GENERAR COMO SALIDA DE LA DPB LA PRIMERA IMAGEN EN EL ORDEN DE SALIDA MARCADA COMO NECESARIA PARA LA SALIDA
S38
MARCAR LA IMAGEN COMO NO NECESARIA PARA LA SALIDA
S39
PARADA

Fig. 11

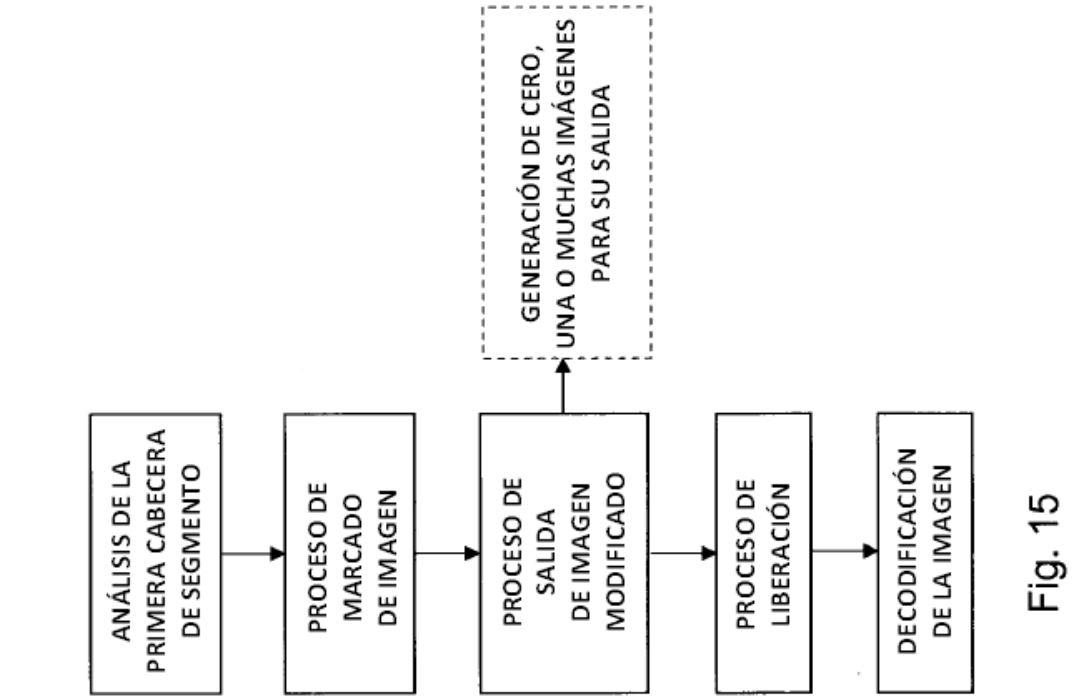
ANÁLISIS DE LA PRIMERA CABECERA DE SEGMENTO
PROCESO DE MARCADO DE IMAGEN
PROCESO DE SALIDA DE IMAGEN MODIFICADO
GENERACIÓN DE CERO, UNA O MUCHAS IMÁGENES PARA SU SALIDA
PROCESO DE LIBERACIÓN
DECODIFICACIÓN DE LA IMAGEN

Fig. 15

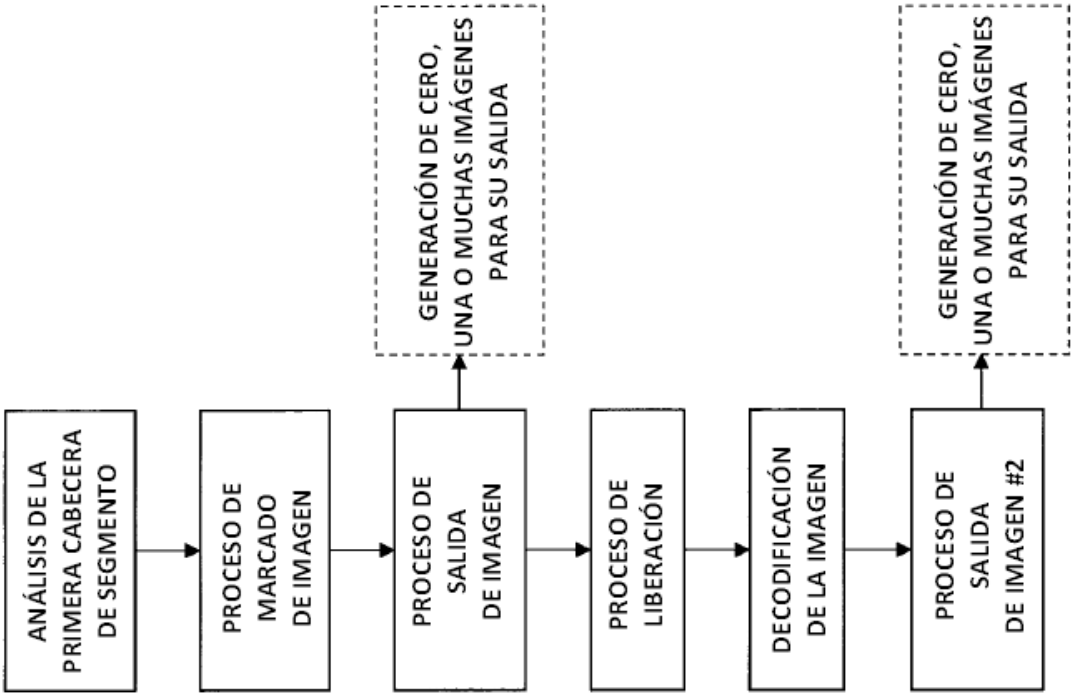
ANÁLISIS DE LA PRIMERA CABECERA DE SEGMENTO
PROCESO DE MARCADO DE IMAGEN
PROCESO DE SALIDA DE IMAGEN
GENERACIÓN DE CERO, UNA O MUCHAS IMÁGENES PARA SU SALIDA
PROCESO DE LIBERACIÓN
DECODIFICACIÓN DE LA IMAGEN
PROCESO DE SALIDA DE IMAGEN #2
GENERACIÓN DE CERO, UNA O MUCHAS IMÁGENES PARA SU SALIDA

Fig. 14

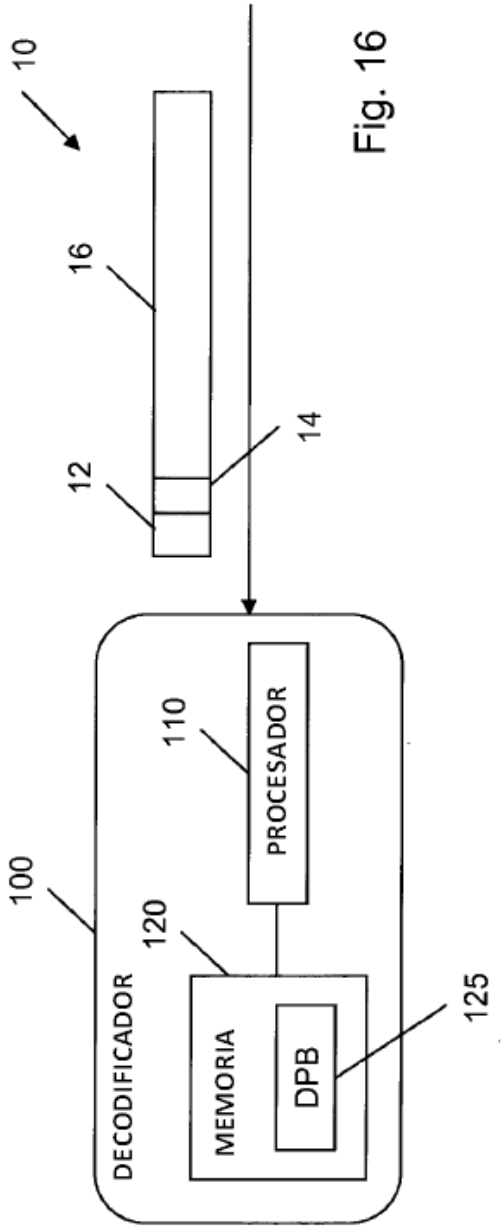
10
16
12
14
100
DECODIFICADOR
110
PROCESADOR
120
MEMORIA
DPB
125

Fig. 16

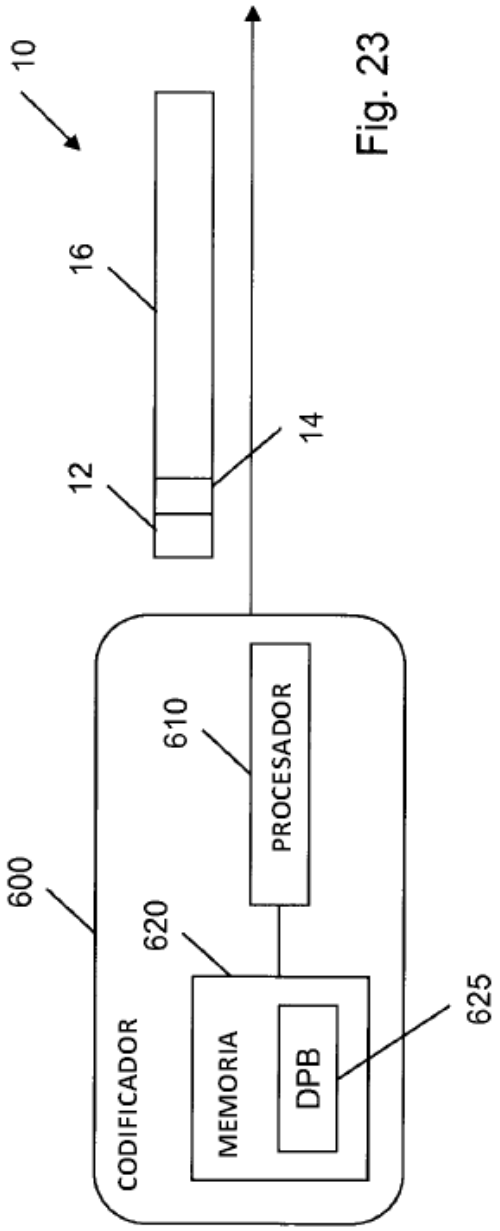
10
16
12
14
600
CODIFICADOR
610
PROCESADOR
620
MEMORIA
DPB
625

Fig. 23

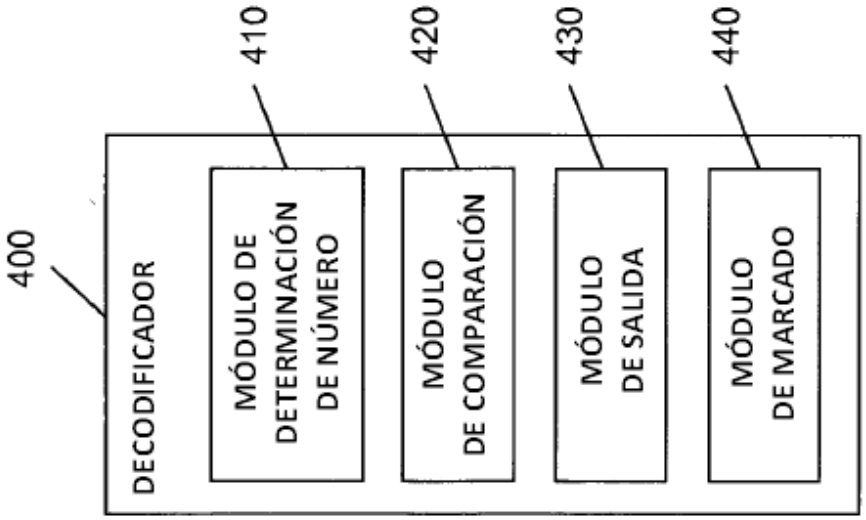
400
DECODIFICADOR
410
MÓDULO DE DETERMINACIÓN DE NÚMERO
420
MÓDULO DE COMPARACIÓN
430
MÓDULO DE SALIDA
440
MÓDULO DE MARCADO

Fig. 19

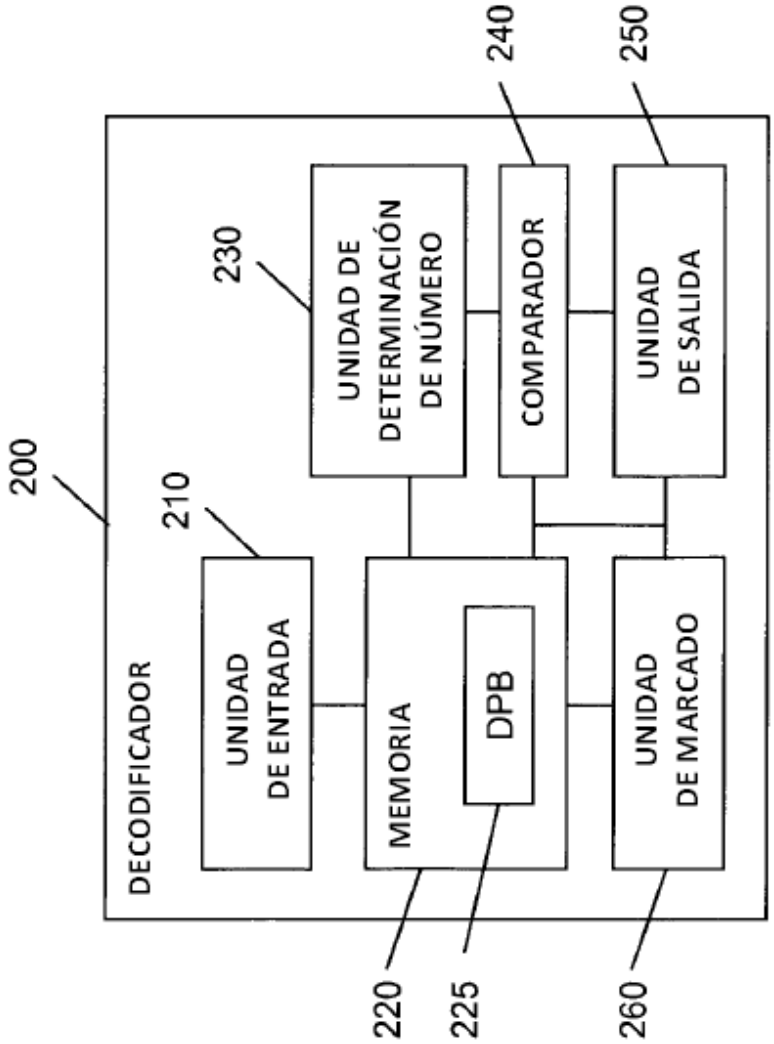
200
DECODIFICADOR
210
UNIDAD DE ENTRADA
220
MEMORIA
225
DPB
230
UNIDAD DE DETERMINACIÓN DE NÚMERO
240
COMPARADOR
250
UNIDAD DE SALIDA
260
UNIDAD DE MARCADO

Fig. 17

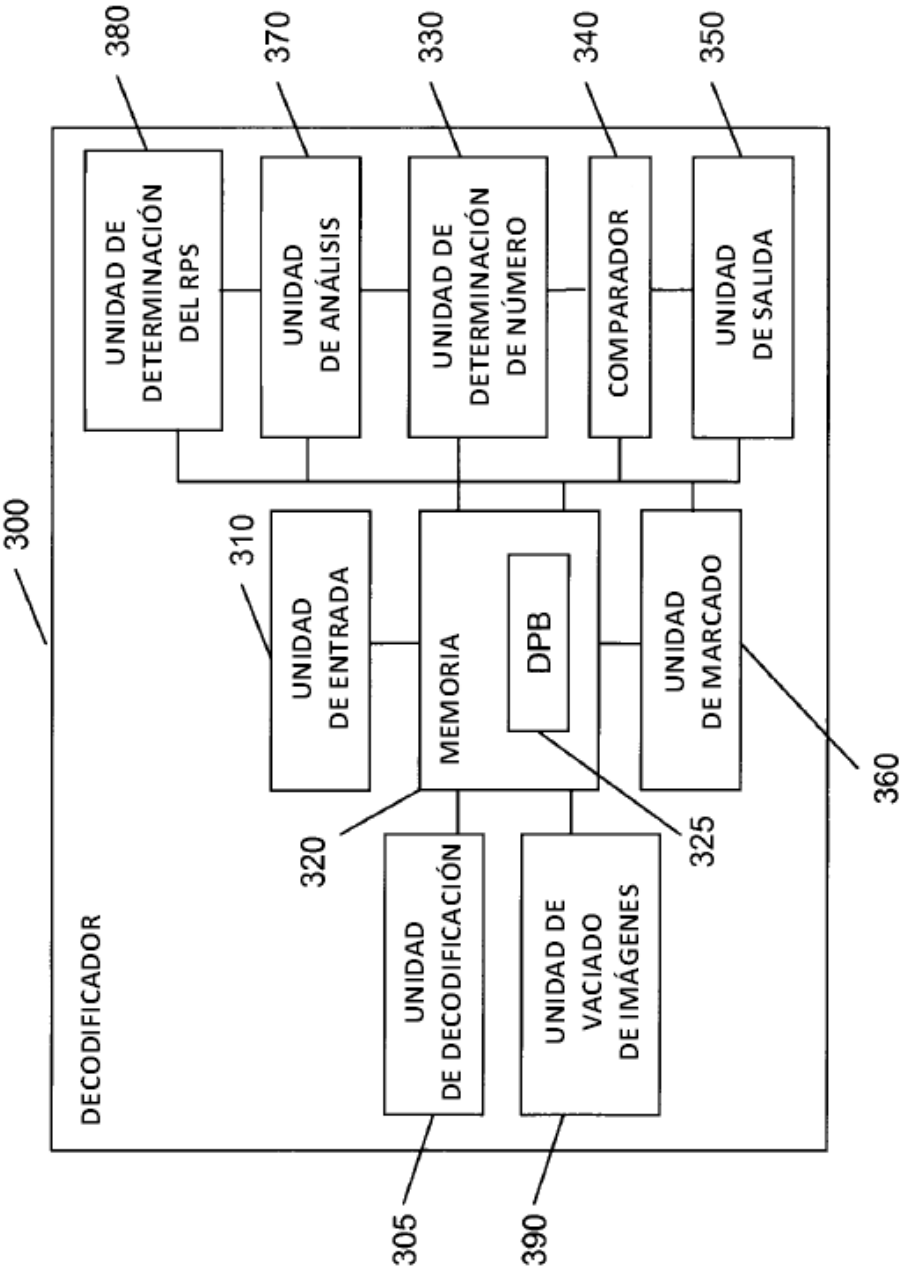
DECODIFICADOR
300
UNIDAD DE DETERMINACIÓN DEL RPS
380
UNIDAD DE ANÁLISIS
370
UNIDAD DE DETERMINACIÓN DE NÚMERO
330
COMPARADOR
340
UNIDAD DE SALIDA
350
UNIDAD DE ENTRADA
310
MEMORIA
DPB
UNIDAD DE MARCADO
360
320
325
UNIDAD DE DECODIFICACIÓN
UNIDAD DE VACIADO DE IMÁGENES
305
390

Fig. 18

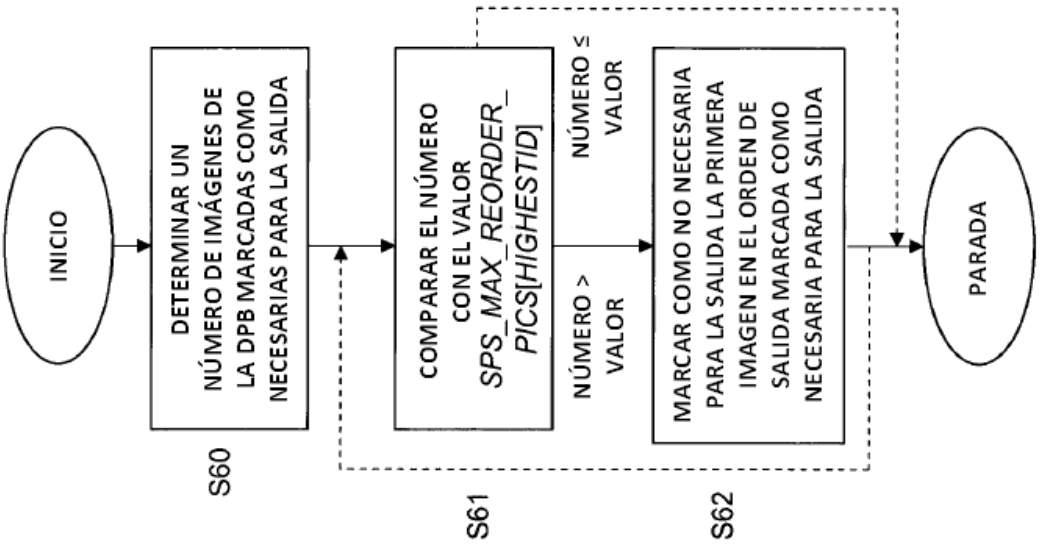
INICIO
DETERMINAR UN NÚMERO DE IMÁGENES DE LA DPB MARCADAS COMO NECESARIAS PARA LA SALIDA
S60
COMPARAR EL NÚMERO CON EL VALOR SPS_MAX_REORDER_PICS[HIGHESTID]
S61
NÚMERO > VALOR
NÚMERO ≤ VALOR
MARCAR COMO NO NECESARIA PARA LA SALIDA LA PRIMERA IMAGEN EN EL ORDEN DE SALIDA MARCADA COMO NECESARIA PARA LA SALIDA
S62
PARADA

Fig. 21

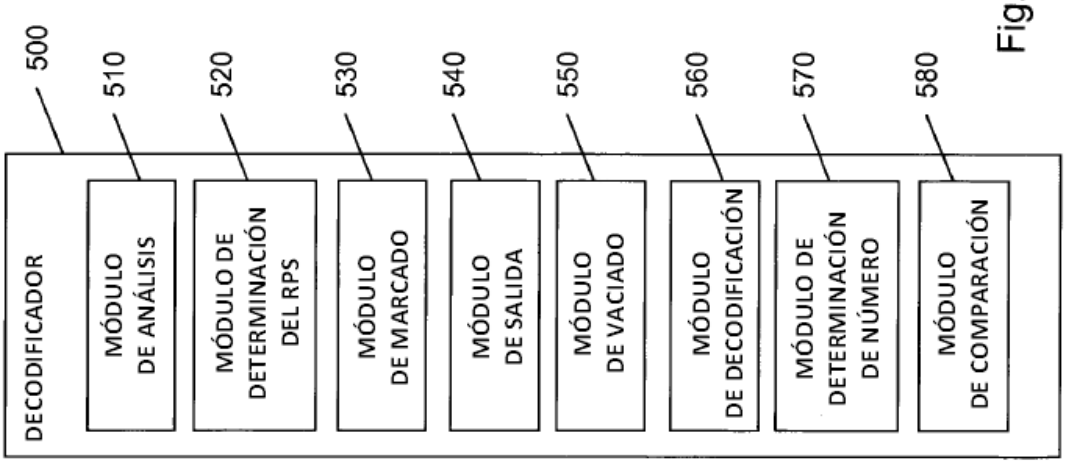
DECODIFICADOR
500
MÓDULO DE ANÁLISIS
510
MÓDULO DE DETERMINACIÓN DEL RPS
520
MÓDULO DE MARCADO
530
MÓDULO DE SALIDA
540
MÓDULO DE VACIADO
550
MÓDULO DE DECODIFICACIÓN
560
MÓDULO DE DETERMINACIÓN DE NÚMERO
570
MÓDULO DE COMPARACIÓN
580

Fig. 20

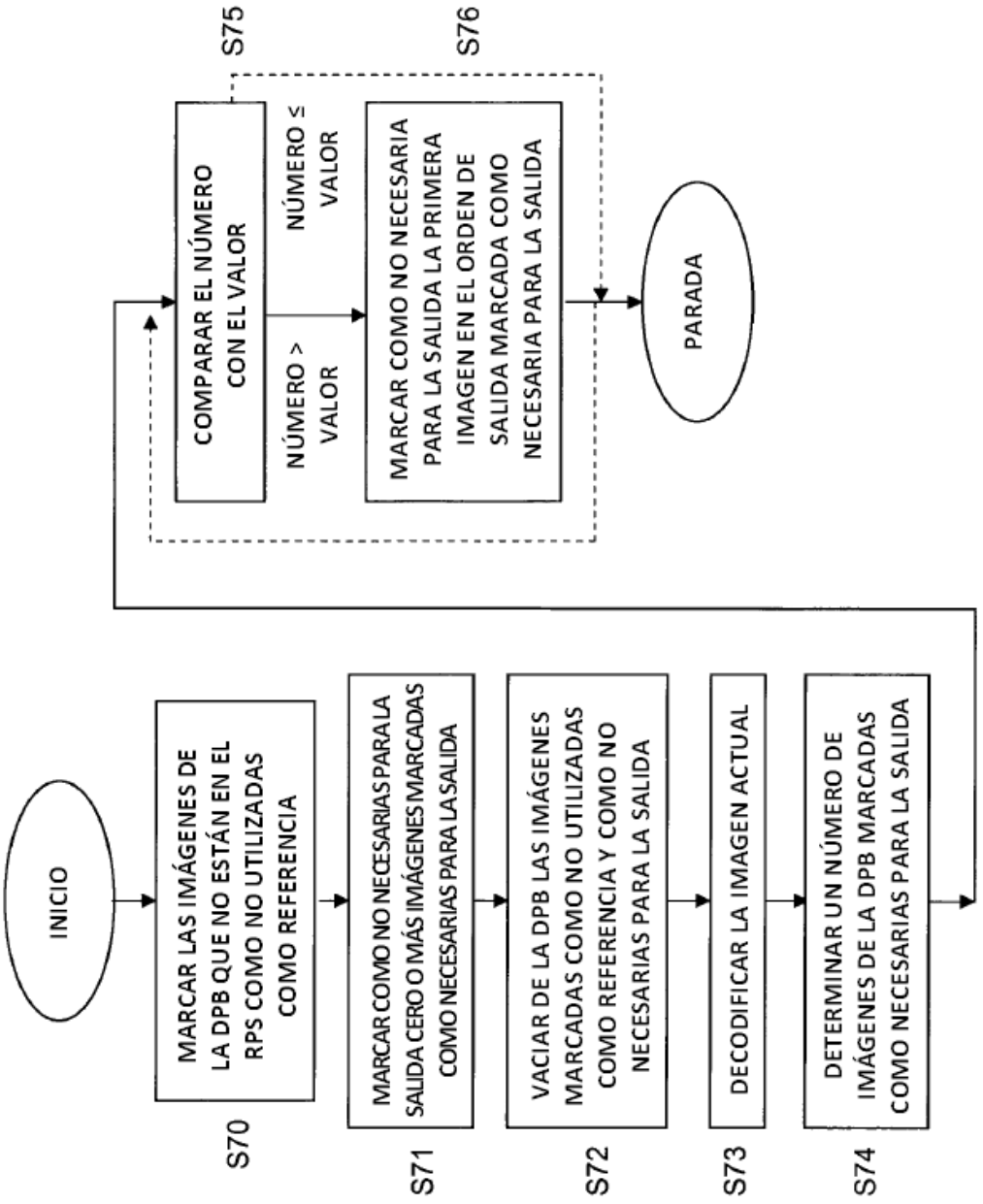
INICIO
MARCAR LAS IMÁGENES DE LA DPB QUE NO ESTÁN EN EL RPS COMO NO UTILIZADAS COMO REFERENCIA
S70
MARCAR COMO NO NECESARIAS PARA LA SALIDA CERO O MÁS IMÁGENES MARCADAS COMO NECESARIAS PARA LA SALIDA
S71
VACIAR DE LA DPB LAS IMÁGENES MARCADAS COMO NO UTILIZADAS COMO REFERENCIA Y COMO NO NECESARIAS PARA LA SALIDA
S72
DECODIFICAR LA IMAGEN ACTUAL
S73
DETERMINAR UN NÚMERO DE IMÁGENES DE LA DPB MARCADAS COMO NECESARIAS PARA LA SALIDA
S74
S75
COMPARAR EL NÚMERO CON EL VALOR
NÚMERO > VALOR
NÚMERO ≤ VALOR
S76
MARCAR COMO NO NECESARIA PARA LA SALIDA LA PRIMERA IMAGEN EN EL ORDEN DE SALIDA MARCADA COMO NECESARIA PARA LA SALIDA
PARADA

Fig. 22

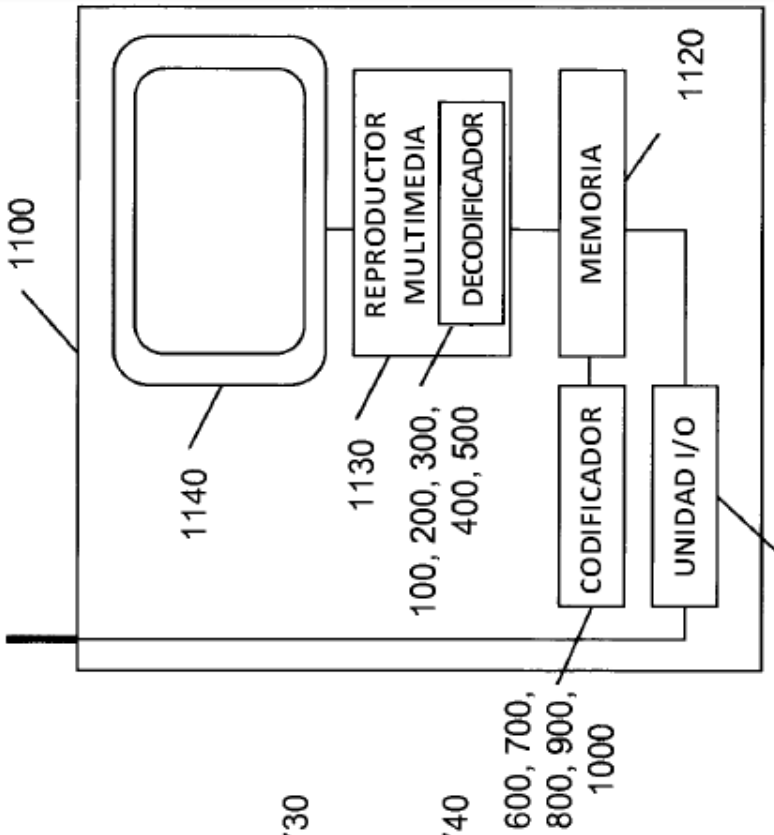
1100
1140
REPRODUCTOR MULTIMEDIA
DECODIFICADOR
1130
100, 200, 300, 400, 500
MEMORIA
1120
CODIFICADOR
UNIDAD I/O
1110
600, 700, 800, 900, 1000

Fig. 28

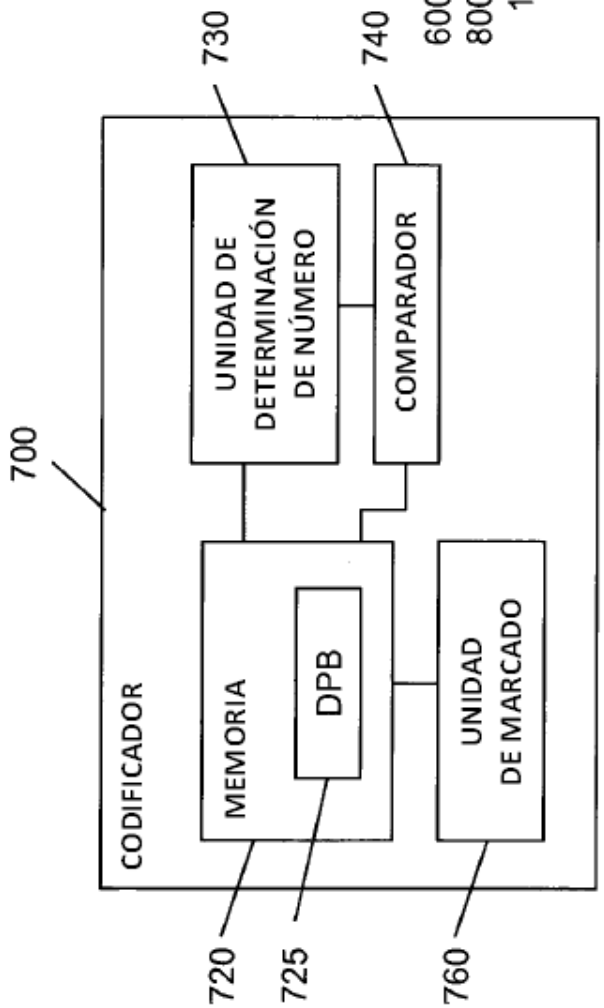
700
CODIFICADOR
730
UNIDAD DE DETERMINACIÓN DE NÚMERO
740
COMPARADOR
MEMORIA
DPB
UNIDAD DE MARCADO
720
725
760

Fig. 24

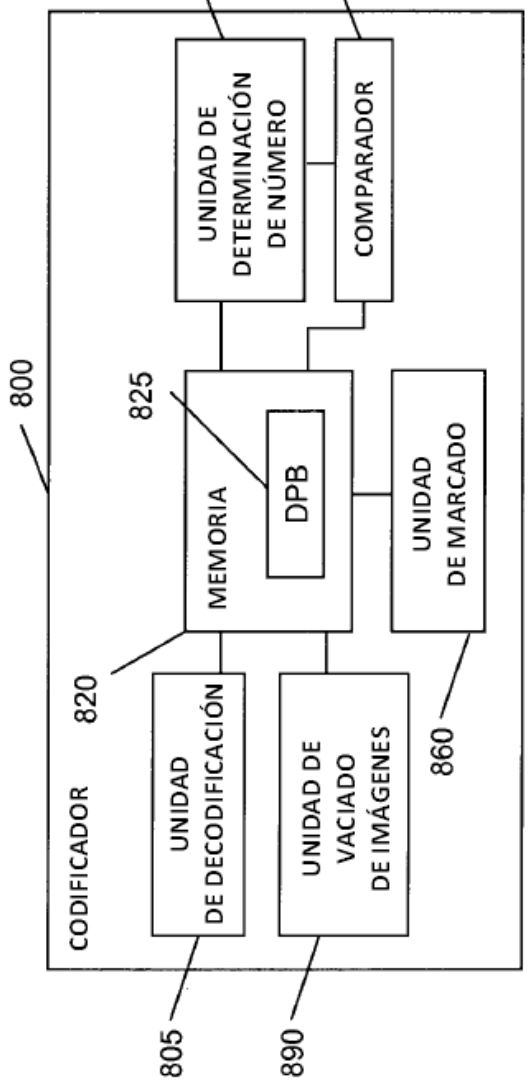
800
CODIFICADOR
820
825
830
840
UNIDAD DE DECODIFICACIÓN
MEMORIA
DPB
UNIDAD DE DETERMINACIÓN DE NÚMERO
COMPARADOR
UNIDAD DE MARCADO
UNIDAD DE VACIADO DE IMÁGENES
805
860
890

Fig. 25

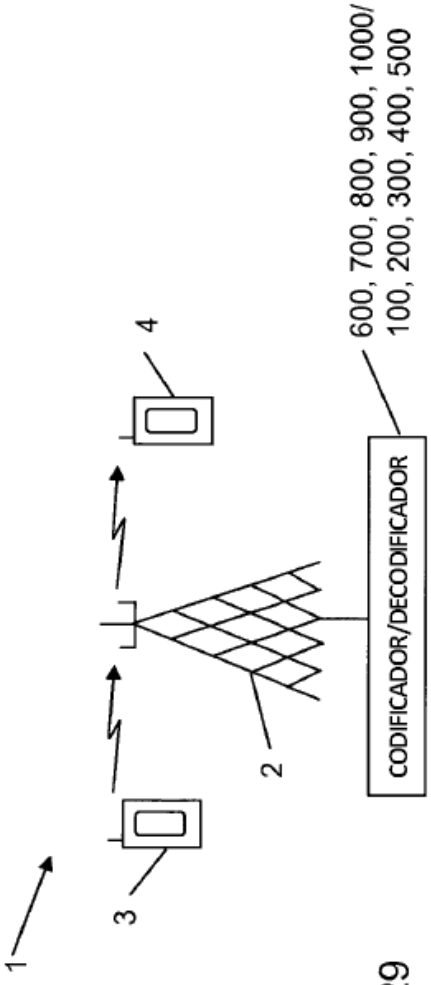
1
2
3
4
600, 700, 800, 900, 1000/
100, 200, 300, 400, 500
CODIFICADOR/DECODIFICADOR

Fig. 29

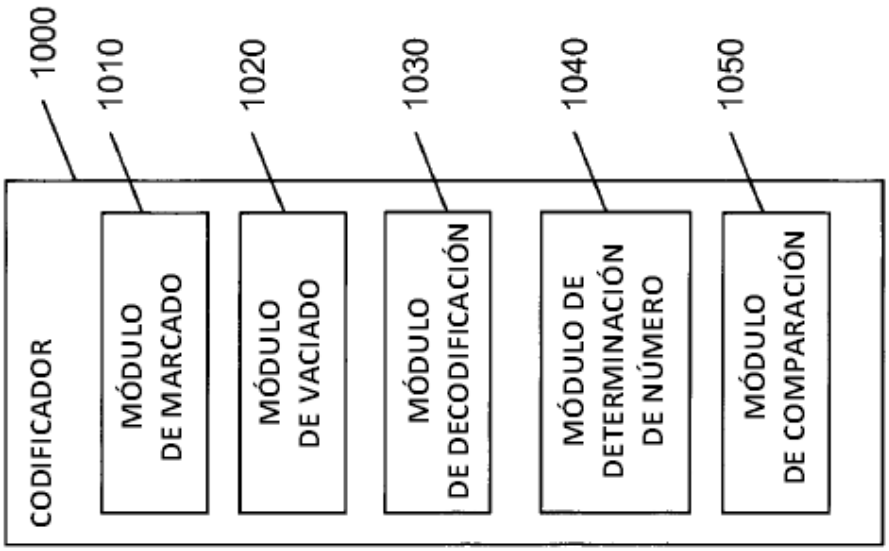
1000
CODIFICADOR
1010
MÓDULO DE MARCADO
1020
MÓDULO DE VACIADO
1030
MÓDULO DE DECODIFICACIÓN
1040
MÓDULO DE DETERMINACIÓN DE NÚMERO
1050
MÓDULO DE COMPARACIÓN

Fig. 27

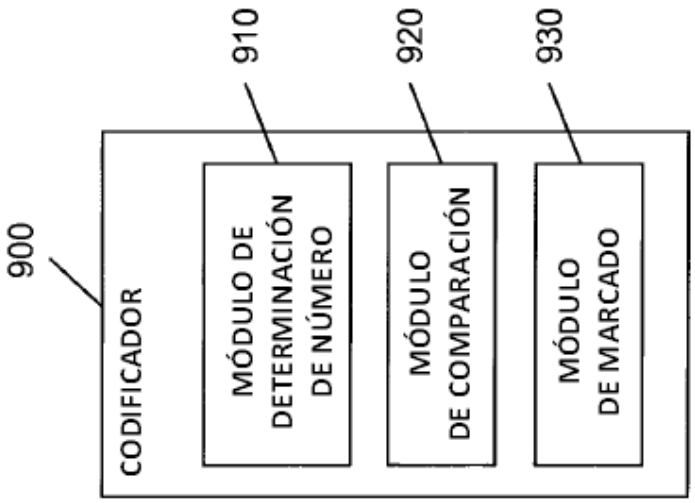
900
CODIFICADOR
910
MÓDULO DE DETERMINACIÓN DE NÚMERO
920
MÓDULO DE COMPARACIÓN
930
MÓDULO DE MARCADO

Fig. 26

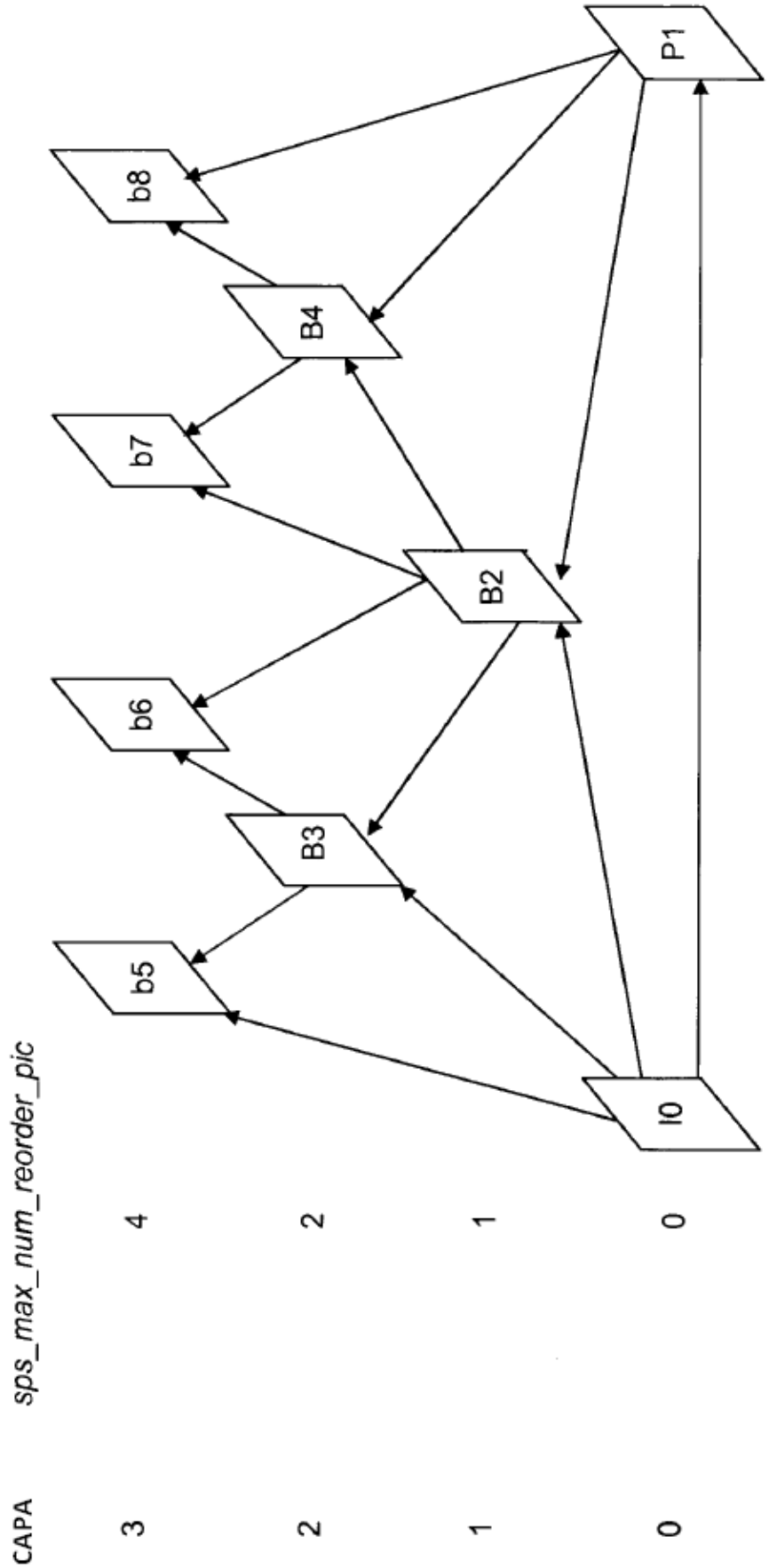

Fig. 30