

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 771 249**

51 Int. Cl.:

H04N 19/13 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **07.11.2012 E 18173012 (8)**

97 Fecha y número de publicación de la concesión europea: **25.12.2019 EP 3396957**

54 Título: **Procedimiento y dispositivo para proporcionar desplazamientos de compensación para un conjunto de muestras reconstruidas de una imagen**

30 Prioridad:

07.11.2011 GB 201119206

24.02.2012 GB 201203285

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

06.07.2020

73 Titular/es:

**CANON KABUSHIKI KAISHA (100.0%)
30-2 Shimomaruko 3-chome, Ohta-ku
Tokyo 146-8501, JP**

72 Inventor/es:

**FRANCOIS, EDOUARD y
LAROCHE, GUILLAUME**

74 Agente/Representante:

DURAN-CORRETJER, S.L.P

ES 2 771 249 T3

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

DESCRIPCIÓN

Procedimiento y dispositivo para proporcionar desplazamientos de compensación para un conjunto de muestras reconstruidas de una imagen

5 La presente invención se refiere a procedimientos y dispositivos para codificar o decodificar una secuencia de imágenes digitales.

10 La invención se puede aplicar al sector del procesamiento digital de señales y, en particular, al sector de la compresión de vídeo que utiliza compensación del movimiento para reducir redundancias espaciales y temporales en los flujos de vídeo.

15 Muchos formatos de compresión de vídeo, tales como por ejemplo H.263, H.264, MPEG-1, MPEG-2, MPEG-4, SVC, utilizan la transformada de coseno discreta (DCT, discrete cosine transform) basada en bloques y compensación del movimiento para eliminar redundancias espaciales y temporales. Estos se denominan a menudo formatos de vídeo predictivos. Cada cuadro o imagen de la señal de vídeo se divide en segmentos que son codificados y pueden ser decodificados de manera independiente. Un segmento es habitualmente una parte rectangular del cuadro, o varias en general, una parte de un cuadro o un cuadro completo. Además, cada segmento puede ser dividido en macrobloques (MB), y cada macrobloque se divide adicionalmente en bloques, habitualmente bloques de 64x64, 20 32x32, 16x16 ú 8x8 píxeles.

25 En la codificación de vídeo de alta eficiencia (HEVC, High Efficiency Video Coding) se pueden utilizar bloques desde 64x64 hasta 4x4. La división se organiza según una estructura de árbol cuádruple en base a la unidad máxima de codificación (LCU, largest coding unit). Una LCU corresponde a un bloque cuadrado de 64x64. Si es necesario dividir una LCU, un indicador de división indica que la LCU se divide en 4 bloques de 32x32. Del mismo modo, si es necesario dividir cualquiera de estos 4 bloques, el valor del indicador de división se pone a verdadero y el bloque de 32x32 se divide en 4 bloques de 16x16, etc. Cuando el valor del indicador de división se pone a falso, el bloque actual es una unidad de codificación CU. Una CU tiene un tamaño igual a 64x64, 32x32, 16x16 ú 8x8 píxeles.

30 Existen dos familias de modos de codificación para los bloques de codificación de una imagen: modos de codificación basados en predicción espacial, denominada predicción INTRA y modos de codificación basados en predicción temporal (INTER, Merge, Skip). En los modos de predicción tanto espacial como temporal, se calcula un residuo restando la predicción del bloque original.

35 Un bloque INTRA se predice, en general, mediante un proceso de predicción INTRA a partir de los píxeles codificados en su límite causal. En una predicción INTRA, se codifica una dirección de predicción.

40 La predicción temporal consiste en encontrar un cuadro de referencia, bien un cuadro anterior o un cuadro futuro de la secuencia de vídeo, una parte de imagen o área de referencia que es la más cercana al bloque a codificar. Esta etapa es conocida habitualmente como estimación del movimiento. A continuación, se predice el bloque a codificar utilizando el área de referencia en una etapa denominada habitualmente como compensación del movimiento - se codifica la diferencia entre el bloque a codificar y la parte de referencia, junto con un elemento de información del movimiento relativo al vector de movimiento que indica el área de referencia a utilizar para la compensación del movimiento. En la predicción temporal, se codifica por lo menos un vector de movimiento.

45 Para reducir aún más el coste de codificar la información de movimiento, en lugar de codificar directamente un vector de movimiento, suponiendo que el movimiento es homogéneo, el vector de movimiento se puede codificar en términos de una diferencia entre el vector de movimiento y un predictor del vector de movimiento, calculado habitualmente a partir de uno o varios vectores de movimiento de los bloques que rodean el bloque a codificar.

50 En H.264, por ejemplo, los vectores de movimiento se codifican con respecto a un predictor promedio calculado a partir de los vectores de movimiento situados en un entorno causal del bloque a codificar, por ejemplo, a partir de los tres bloques situados por encima y a la izquierda del bloque a codificar. Únicamente la diferencia, denominada como vector de movimiento residual, entre el predictor promedio y el vector de movimiento del bloque actual, es codificada en el flujo de bits para reducir el coste de la codificación.

55 La codificación que utiliza los vectores de movimiento residuales ahorra algo de tasa de bits, pero necesita que el decodificador realice el mismo cálculo del predictor del vector de movimiento para decodificar el valor del vector de movimiento de un bloque a decodificar.

60 Ambos procesos de codificación y decodificación pueden implicar un proceso de decodificación de una imagen codificada. Este proceso se lleva a cabo habitualmente en el lado del codificador para el propósito de la estimación del movimiento futuro, lo que permite que un codificador y un decodificador correspondiente tengan los mismos cuadros de referencia.

65 Para reconstruir el cuadro codificado, el residuo se somete a cuantificación inversa y a la transformada inversa para

- proporcionar el residuo “descodificado” en el dominio de píxeles. A continuación se filtra la primera reconstrucción mediante uno o varios tipos de procesos de filtrado posterior. Estos filtros posteriores se aplican al cuadro reconstruido en el lado del codificador y del descodificador para que se utilice el mismo cuadro de referencia en ambos lados. El objetivo de este filtrado posterior es eliminar los artefactos de compresión. Por ejemplo, H.264/AVC utiliza un filtro anti-efecto bloque. Este filtro puede eliminar los artefactos de efecto bloque debido a la cuantificación DCT del residuo y a la compensación del movimiento de los bloques. En el estándar HEVC actual se utilizan 3 tipos de filtros de bucle: filtro anti-efecto bloque, desplazamiento adaptativo de muestras (SAO, sample adaptive offset) y filtro de bucle adaptativo (ALF, adaptive loop filter).
- La **figura 1** es un diagrama de flujo que muestra las etapas del proceso de filtrado de bucle de una implementación de HEVC conocida. En una etapa inicial -101-, el codificador o descodificador genera la reconstrucción de todo el cuadro. A continuación, en la etapa -102-, se aplica un filtro anti-efecto bloque a esta primera reconstrucción para generar una reconstrucción sin bloques -103-. El objetivo del filtro anti-efecto bloque es eliminar los artefactos de bloque generados por la cuantificación de residuos y la compensación del movimiento de bloques o predicción de bloques INTRA. Estos artefactos son visualmente importantes a tasas de bits bajas. El filtro anti-efecto bloque funciona para suavizar los límites de los bloques según las características de dos bloques contiguos. Se tienen en cuenta el modo de codificación de cada bloque, los parámetros de cuantificación utilizados para la codificación de residuos y las diferencias de píxeles contiguos en el límite. Se aplica el mismo criterio/clasificación para todos los cuadros y no se transmiten datos adicionales. El filtro anti-efecto bloque mejora la calidad visual del cuadro actual eliminando los artefactos de efecto bloque y mejora asimismo la estimación del movimiento y la compensación del movimiento para los siguientes cuadros. De hecho, se eliminan las altas frecuencias del artefacto de bloque y, de este modo, no es necesario compensar estas altas frecuencias con la textura residual de los siguientes cuadros.
- Después del filtro anti-efecto bloque, se filtra la reconstrucción sin bloques mediante un filtro de bucle de desplazamiento adaptativo de muestras (SAO) en la etapa -104-. El cuadro resultante -105- es filtrado a continuación con un filtro de bucle adaptativo (ALF) en la etapa -106- para generar el cuadro reconstruido -107- que se mostrará y utilizará como un cuadro de referencia para los siguientes cuadros Inter.
- El objetivo del filtro de bucle del SAO y del ALF es mejorar la reconstrucción del cuadro enviando datos adicionales a diferencia de un filtro anti-efecto bloque en el que no se transmite ninguna información.
- El principio del filtro de bucle de SAO es clasificar cada píxel en una clase y añadir el mismo valor de desplazamiento al valor de píxel respectivo de cada píxel de la clase. Por lo tanto, se transmite un desplazamiento para cada clase. El filtrado de bucle de SAO proporciona dos tipos de clasificación para un área del cuadro: un desplazamiento de borde y un desplazamiento de banda.
- La clasificación de desplazamiento de borde implica determinar una clase para cada píxel comparando su valor de píxel correspondiente con los valores de píxel de dos píxeles contiguos. Además, los dos píxeles contiguos dependen de un parámetro que indica la dirección de los 2 píxeles contiguos. Estas direcciones son de 0 grados (dirección horizontal), de 45 grados (dirección diagonal), de 90 grados (dirección vertical) y de 135 grados (segunda dirección diagonal). En adelante, estas direcciones son denominadas “tipos” de clasificación de desplazamientos de borde.
- El segundo tipo de clasificación es una clasificación de desplazamiento de banda que depende del valor de píxel. Una clase en desplazamiento de banda del SAO corresponde a un intervalo de valores de píxel. Por lo tanto, se añade el mismo desplazamiento a todos los píxeles que tienen un valor de píxel dentro de un intervalo determinado de valores de píxel.
- Para que sea más adaptativo al contenido del cuadro, se ha propuesto aplicar el filtrado de SAO en base a una estructura de árbol cuádruple para codificar el SAO. Por consiguiente, el área del cuadro que corresponde a un nodo hoja del árbol cuádruple puede o puede no ser filtrado mediante el SAO, de tal manera que únicamente se filtran algunas áreas. Además, cuando se activa el SAO, solamente se utiliza una clasificación de SAO: el desplazamiento de borde o el desplazamiento de banda según los parámetros relacionados transmitidos para cada clasificación. Finalmente, para cada nodo hoja de SAO, se transmiten la clasificación de SAO así como sus parámetros y los desplazamientos de todas las clases.
- La principal ventaja del árbol cuádruple es seguir eficientemente las propiedades locales de la señal. No obstante, esto requiere una codificación dedicada en el flujo de bits. También se puede contemplar otra solución que sustituye la codificación basada en árbol cuádruple de los parámetros de SAO por una codificación a nivel de LCU.
- Una imagen de los datos de vídeo a codificar se puede proporcionar como un conjunto de matrices bidimensionales (también conocidas como canales de color) de los valores de muestra, de las que cada entrada representa la intensidad de un componente de color, tal como una medida del brillo luma y las variaciones de color croma del color neutro de la escala de grises hacia azul o rojo (YUV) o como una medida de la intensidad de los componentes de la luz rojo, verde o azul (RGB). Un modelo YUV define un espacio de color en términos de un componente de luma (Y) y dos de crominancia (UV). En general, Y representa el componente de luma (brillo) y U y V son los componentes de

crominancia (color) o croma.

El filtrado de SAO se aplica habitualmente de modo independiente sobre los componentes luma y ambos componentes croma U y V.

5 Una implementación conocida del desplazamiento de banda de SAO descrita en CHIH-MING FU et al. "Sample adaptive offset for HEVC", Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop On, IEEE, 17 de octubre de 2011, páginas 1 a 5, XP032027547, DOI: 10.1109/MMSP.2011.6093807 ISBN: 978-1-4577-1432-0 divide el intervalo de los valores de píxel en 32 intervalos predefinidos del mismo tamaño, tal como se representa en la figura 2. El valor mínimo del intervalo de los valores de píxel es sistemáticamente 0 y el valor máximo depende de la profundidad de bits de los valores de píxel según la siguiente relación $Max = 2^{Profundidaddebits} - 1$. Por ejemplo, cuando la profundidad de bits es de 8 bits, el valor máximo de un píxel puede ser 255. De este modo, el intervalo de los valores de píxel está entre 0 y 255. Para esta profundidad de bits de 8 bits, cada clase incluye un intervalo de 16 valores de píxel. Además, para un desplazamiento de banda de SAO se consideran 2 grupos de clases. El primero contiene 16 clases sucesivas en el centro del intervalo de los valores de píxel, tal como se representan en color gris en la figura 2. El segundo grupo contiene asimismo 16 clases pero en ambos extremos del intervalo de valores de píxel, tal como se representa mediante sombreado en la figura 2. Para el desplazamiento de banda de SAO de un área del cuadro, el grupo utilizado para la clasificación y los 16 desplazamientos son introducidos en el flujo de bits.

20 La **figura 3** es un diagrama de flujo que muestra las etapas de un procedimiento para seleccionar desplazamientos en un codificador para un área del cuadro actual -303-. El área del cuadro contiene N píxeles. En una etapa inicial -301-, las variables Sum_j y $SumNbPix_j$ se fijan a un valor cero para cada uno de los 16 intervalos. j indica el intervalo o número de clase actual. Sum_j indica la suma de la diferencia entre el valor de píxel en el intervalo j y el valor de sus píxeles originales correspondientes. $SumNbPix_j$ indica el número de píxeles en el intervalo j .

25 En la etapa -302-, la variable de contador i se fija al valor cero. A continuación, en la etapa -304- se extrae el primer píxel del área del cuadro -303-. Se supone que el grupo de SAO actual que está siendo procesado es conocido (primero o segundo según se representan en la **figura 2**). Si se determina en la etapa -305- que el valor de píxel P_i no está en el grupo de SAO actual, entonces el valor de la variable de contador i se incrementa en la etapa -308- para clasificar los píxeles posteriores del área del cuadro -303-. De lo contrario, si se determina en la etapa -305- que el valor de píxel P_i está en el grupo de SAO actual, se encuentra en la etapa -306- el número de intervalo (o número de clase) j correspondiente al valor de P_i . En la etapa -307- posterior, se incrementa la variable $SumNbPix_j$ correspondiente y la diferencia entre P_i y su valor original P_i^{orig} se añade a Sum_j . En la etapa siguiente, se incrementa la variable de contador i para aplicar la clasificación a los demás píxeles del área del cuadro -303-. En la etapa -309- se determina si se han clasificado o no todos los píxeles N del área del cuadro -303- (es decir, si $i \geq N$), en caso afirmativo, se calcula un *Desplazamiento_j* para cada clase en la etapa -310- para generar una tabla de desplazamientos -311- que presenta un desplazamiento para cada clase j como el resultado final del algoritmo de selección del desplazamiento. Este desplazamiento se calcula como el promedio de la diferencia entre los valores de píxel de los píxeles de la clase j y sus respectivos valores de píxel originales. El *Desplazamiento_j* para la clase j se determina mediante la siguiente ecuación:

$$Desplazamiento_j = \frac{Sum_j}{SumNbPix_j} \quad (1)$$

45 La **figura 4** es un diagrama de flujo que muestra las etapas de un proceso de descodificación que aplica los desplazamientos de banda de SAO a los correspondientes grupos de clases. Este proceso se puede aplicar asimismo en el lado del codificador para generar el cuadro de referencia utilizado para la estimación de movimiento y la compensación de los siguientes cuadros.

50 Una etapa inicial -401- del proceso implica la descodificación de los valores de desplazamiento para cada clase de valores de píxel con el fin de generar una tabla de desplazamientos -402-. En el lado del codificador, la tabla de desplazamientos -402- es el resultado del algoritmo de selección mostrado en la figura 3. Por lo tanto, en el lado del codificador, la etapa -401- se sustituye por el algoritmo de selección de desplazamiento de la figura 3.

55 En la etapa -403- se pone a 0 una variable de contador i . El píxel P_i se extrae en la etapa -405- de un área del cuadro -404- que contiene N píxeles. En la etapa -406- se determina si el píxel P_i pertenece o no al grupo de clases actual. Si se determina que el píxel P_i está en el grupo de clases actual, se identifica el número de clase j relacionado y en la etapa -409- se extrae el valor de desplazamiento *Desplazamiento_j* relacionado a partir de la tabla de desplazamientos -402-. El valor de desplazamiento extraído *Desplazamiento_j* se suma a continuación al valor de píxel de P_i en la etapa -410- para generar el valor de píxel filtrado P_i en la etapa -411-. El valor de píxel filtrado se introduce a continuación en el área del cuadro filtrado -415- en la etapa -412- en el píxel correspondiente.

Si en la etapa -406- se determina que el píxel P_i no está en el grupo de desplazamientos de banda de SAO, a continuación en la etapa -412- el valor de píxel de P_i se pone sin filtrado en el área del cuadro filtrado -415-. Después

de la etapa -412-, la variable de contador i se incrementa para filtrar, en caso necesario, los siguientes píxeles del área del cuadro actual -404-. Después de que se ha determinado en la etapa -414- que se han procesado todos los píxeles N del área del cuadro (es decir, $i \geq N$), se reconstruye el área del cuadro filtrado -415- y se puede añadir al cuadro reconstruido de SAO (véase el cuadro -105- de la figura 1).

5 Un inconveniente de los procesos conocidos para la selección de las compensaciones es que no están adaptados a variaciones diferentes en el contenido de píxeles de la imagen y a tipos diferentes de componentes de los píxeles de la imagen.

10 La presente invención se ha concebido para abordar uno o varios de los problemas anteriores.

Según un primer aspecto de la presente invención, se da a conocer un procedimiento para codificar una imagen, según se define en cualquiera de las reivindicaciones 1 a 4.

15 Según un segundo aspecto de la presente invención, se da a conocer un procedimiento para decodificar una imagen, según se define en la reivindicación 5.

Según un tercer aspecto de la presente invención se da a conocer una señal que transporta un conjunto de datos de información para una imagen, representada por un flujo de bits de vídeo, según se define en la reivindicación 8.

20 Según un cuarto aspecto de la presente invención, se da a conocer un dispositivo de codificación para codificar una imagen, según se define en la reivindicación 9.

25 Según un quinto aspecto de la presente invención se da a conocer un dispositivo para decodificar una imagen, según se define en la reivindicación 10.

En el contexto de la presente invención una muestra puede corresponder a un único píxel, con un valor de muestra correspondiente al respectivo valor de píxel. Alternativamente, una muestra puede comprender una serie de píxeles, y el valor de la muestra puede corresponder a un valor de píxel determinado a partir de los valores de píxel de la serie de píxeles.

30 Por lo menos partes de los procedimientos según la invención pueden ser implementados por ordenador. Por consiguiente, la presente invención puede adoptar la forma de una realización completamente de hardware, una realización completamente de software (incluyendo firmware, software residente, microcódigo, etc.) o una realización que combina aspectos de software y de hardware que pueden ser todos denominados, en general, en este documento como un "circuito", "módulo" o "sistema". Además, la presente invención puede adoptar la forma de un producto de programa informático incorporado en cualquier medio de expresión tangible que tenga código de programa utilizable por ordenador incorporado en el medio.

40 Dado que la presente invención se puede implementar en software, la presente invención se puede realizar como código legible por ordenador a proporcionar a un aparato programable en cualquier medio portador adecuado. Un medio portador tangible puede comprender un medio de almacenamiento, tal como un disquete, un CD-ROM, una unidad de disco duro, un dispositivo de cinta magnética o un dispositivo de memoria de estado sólido y similares. Un medio portador transitorio puede incluir una señal, tal como una señal eléctrica, una señal electrónica, una señal óptica, una señal acústica, una señal magnética o una señal electromagnética, por ejemplo, una señal de microondas o de radiofrecuencia.

De este modo, según un sexto aspecto de la presente invención se da a conocer un producto de programa informático, según se define en la reivindicación 6.

50 Según un séptimo aspecto de la presente invención se da a conocer un medio de almacenamiento legible por ordenador, según se define en la reivindicación 7.

A continuación se describirán las realizaciones de la invención, únicamente a modo de ejemplo, y haciendo referencia a los siguientes dibujos en los que:

la figura 1 es un diagrama de flujo que muestra las etapas de un proceso de filtrado de bucle de la técnica anterior;

60 la figura 2 muestra gráficamente una clasificación del desplazamiento de banda adaptativo de las muestras de un proceso de HEVC de la técnica anterior;

la figura 3 es un diagrama de flujo que muestra las etapas de un proceso para determinar los desplazamientos de compensación para un desplazamiento de banda de SAO de la HEVC;

65 la figura 4 es un diagrama de flujo que muestra las etapas de un proceso de filtrado de desplazamiento de banda de SAO de la HEVC;

la figura 5 es un diagrama de bloques que muestra, de modo esquemático, un sistema de comunicación de datos en el que se pueden implementar una o varias realizaciones de la invención;

5 la figura 6 es un diagrama de bloques que muestra los componentes de un dispositivo de procesamiento en el que se pueden implementar una o varias realizaciones de la invención;

la figura 7 es un diagrama de flujo que muestra las etapas de un procedimiento de codificación, según las realizaciones de la invención;

10 la figura 8 es un diagrama de flujo que muestra las etapas de un proceso de filtrado de bucle de acuerdo con una o varias realizaciones de la invención;

15 la figura 9 es un diagrama de flujo que muestra las etapas de un procedimiento de descodificación, según las realizaciones de la invención;

la figura 10 es un diagrama de flujo que muestra las etapas de un procedimiento para determinar la clasificación del desplazamiento de banda de SAO que no incorpora la invención;

20 la figura 11 es un diagrama de flujo que muestra las etapas de un procedimiento para determinar una clasificación equiprobable del procedimiento de la figura 10;

la figura 12 es un diagrama de flujo que muestra las etapas de un procedimiento para determinar una clasificación equiprobable como una alternativa a las etapas de la figura 11;

25 la figura 13 muestra varios tamaños del intervalo útil para la clasificación, según una realización de la invención;

la figura 14 muestra varios tamaños de clases para la clasificación, según una realización de la invención;

30 la figura 15 muestra varios tamaños de las clases en un intervalo útil para la clasificación, según una realización de la invención;

la figura 16 muestra varias posiciones del centro de un intervalo útil de un primer grupo para la clasificación, según una realización de la invención;

35 la figura 17 muestra varias posiciones del centro de un intervalo útil del segundo grupo para la clasificación, según una realización de la invención; y

40 la figura 18 muestra la selección de tasa-distorsión de la clasificación de parámetros, según una realización de la invención.

Las figuras 19a y 19b muestran las posibles posiciones del intervalo útil dentro del intervalo completo, según otra realización de la invención.

45 La figura 20A muestra un pseudocódigo aplicado en la técnica anterior para codificar los parámetros del SAO a nivel de la LCU.

La figura 20B muestra un pseudocódigo mejorado según una realización de la invención, para codificar los parámetros del SAO a nivel de la LCU.

50 La figura 21 es un diagrama de flujo correspondiente al pseudocódigo de la figura 20A.

La figura 22 es un diagrama de flujo correspondiente al pseudocódigo de la figura 20B.

55 La figura 23 es un diagrama de flujo para utilizar en la explicación de la codificación de los parámetros del SAO según otra realización de la presente invención.

La figura 24 muestra un pseudocódigo utilizado para codificar los parámetros del SAO según otra realización más de la presente invención.

60 La figura 25 es un diagrama de flujo correspondiente al pseudocódigo de la figura 24.

La figura 26 es un diagrama de flujo para utilizar en la explicación de la codificación de los parámetros del SAO según otra realización más de la presente invención.

65 La **figura 5** muestra un sistema de comunicación de datos en el que se pueden implementar una o varias

realizaciones de la invención. El sistema de comunicación de datos comprende un dispositivo de transmisión, en este caso un servidor -501-, que puede funcionar para transmitir paquetes de datos de un flujo de datos a un dispositivo receptor, en este caso un terminal cliente -502-, a través de una red de comunicación de datos -500-. La red de comunicación de datos -500- puede ser una red de área amplia (WAN) o una red de área local (LAN). Dicha red puede ser, por ejemplo, una red inalámbrica (Wifi/802.11a o b o g), una red Ethernet, una red de internet o una red mixta compuesta por varias redes diferentes. En una realización determinada de la invención, el sistema de comunicación de datos puede ser un sistema de emisión de televisión digital en el que el servidor -501- envía el mismo contenido de datos a múltiples clientes.

El flujo de datos -504- proporcionado por el servidor -501- puede estar compuesto por datos multimedia que representan datos de vídeo y audio. Los flujos de datos de vídeo y audio pueden ser capturados, en algunas realizaciones de la invención, por el servidor -501- utilizando un micrófono y una cámara respectivamente. En algunas realizaciones, los flujos de datos pueden estar almacenados en el servidor -501- o ser recibidos por el servidor -501- desde otro proveedor de datos, o ser generados en el servidor -501-. El servidor -501- está dotado de un codificador para codificar flujos de vídeo y audio, en particular, para proporcionar un flujo de bits comprimido para su transmisión, que es una representación más compacta de los datos presentados como una entrada al codificador.

Para obtener una mejor relación entre la calidad de los datos transmitidos y la cantidad de datos transmitidos, la compresión de los datos de vídeo puede ser, por ejemplo, según el formato HEVC o el formato H.264/AVC.

El cliente -502- recibe el flujo de bits transmitido y descodifica el flujo de bits reconstruido para reproducir las imágenes de vídeo en un dispositivo de visualización y los datos de audio mediante un altavoz.

Aunque en el ejemplo de la figura 5 se considera un escenario de transmisión, se apreciará que en algunas realizaciones de la invención, la comunicación de datos entre un codificador y un descodificador se puede llevar a cabo utilizando, por ejemplo, un dispositivo de almacenamiento multimedia, tal como un disco óptico.

En una o varias realizaciones de la invención, se transmite una imagen de vídeo con datos representativos de los desplazamientos de compensación para su aplicación a los píxeles reconstruidos de la imagen con el fin de proporcionar píxeles filtrados en una imagen final.

La **figura 6** muestra, de modo esquemático, un dispositivo de procesamiento -600- configurado para implementar, por lo menos, una realización de la presente invención. El dispositivo de procesamiento -600- puede ser un dispositivo, tal como un microordenador, una estación de trabajo o un dispositivo portátil ligero. El dispositivo -600- comprende un bus de comunicaciones -613- conectado a:

- una unidad central de procesamiento -611-, tal como un microprocesador, denominada CPU;
- una memoria de sólo lectura -607-, denominada ROM, para almacenar programas informáticos para implementar la invención;
- una memoria de acceso aleatorio -612-, denominada RAM, para almacenar el código ejecutable del procedimiento de las realizaciones de la invención así como los registros adaptados para registrar las variables y parámetros necesarios para implementar el procedimiento de codificación de una secuencia de imágenes digitales y/o el procedimiento de descodificación de un flujo de bits según las realizaciones de la invención; y
- una interfaz de comunicaciones -602- conectada a una red de comunicaciones -603- sobre la que se transmiten o reciben los datos digitales a procesar.

Opcionalmente, el aparato -600- puede incluir asimismo los siguientes componentes:

- medios de almacenamiento de datos -604-, tales como un disco duro, para almacenar programas informáticos para implementar los procedimientos de una o varias realizaciones de la invención y los datos utilizados o generados durante la implementación de una o varias realizaciones de la invención;
- una unidad de disco -605- para un disco -606-, estando adaptada la unidad de disco para leer datos del disco -606- o escribir datos en dicho disco;
- una pantalla -609- para visualizar datos y/o servir como interfaz gráfica con el usuario, por medio de un teclado -610- o cualesquiera otros medios de señalización.

El aparato -600- puede estar conectado a varios periféricos, tales como, por ejemplo, una cámara digital -620- o un micrófono -608-, estando conectado cada uno a una tarjeta de entrada/salida (no mostrada) para suministrar datos multimedia al aparato -600-.

El bus de comunicaciones proporciona comunicación e interoperabilidad entre los diversos elementos incluidos en el aparato -600- o conectados al mismo. La representación del bus no es limitativa y, en particular, la unidad central de procesamiento puede funcionar para comunicar instrucciones a cualquier elemento del aparato -600-, directamente o por medio de otro elemento del aparato -600-.

El disco -606- puede ser sustituido por cualquier medio de información, tal como, por ejemplo, un disco compacto

(CD-ROM), regrabable o no, un disco ZIP o una tarjeta de memoria y, en términos generales, por medios de almacenamiento de la información que pueden ser leídos por un microordenador o por un microprocesador, integrados o no en el aparato, posiblemente extraíbles y adaptados para almacenar uno o varios programas cuya ejecución permite el procedimiento de codificación de una secuencia de imágenes digitales y/o el procedimiento de descodificación de un flujo de bits, según la invención a implementar.

El código ejecutable puede estar almacenado bien en la memoria de sólo lectura -607-, en el disco duro -604- o en un medio digital extraíble, tal como, por ejemplo, un disco -606-, tal como se ha descrito anteriormente. Según una variante, el código ejecutable de los programas puede ser recibido por medio de la red de comunicación -603-, mediante la interfaz -602-, para ser almacenado en uno de los medios de almacenamiento del aparato -600- antes de ser ejecutado, tal como el disco duro -604-.

La unidad central de procesamiento -611- está adaptada para controlar y dirigir la ejecución de las instrucciones o porciones del código de software del programa o programas según la invención, instrucciones que están almacenadas en uno de los medios de almacenamiento mencionados anteriormente. En el encendido, el programa o programas que están almacenados en una memoria no volátil, por ejemplo, en el disco duro -604- o en la memoria de sólo lectura -607-, son transferidos a la memoria de acceso aleatorio -612-, que contiene a continuación el código ejecutable del programa o programas, así como los registros para almacenar las variables y parámetros necesarios para implementar la invención.

En esta realización, el aparato es un aparato programable que utiliza software para implementar la invención. No obstante, alternativamente, la presente invención puede ser implementada en hardware (por ejemplo, en forma de circuito integrado de aplicación específica o ASIC).

La **figura 7** muestra un diagrama de bloques de un codificador según, por lo menos, una realización de la invención. El codificador está representado por módulos conectados, estando adaptado cada módulo para implementar, por ejemplo, en forma de instrucciones de programación a ejecutar por la CPU -611- del dispositivo -600-, por lo menos una etapa correspondiente de un procedimiento que implementa, por lo menos, una realización de codificación de una imagen de una secuencia de imágenes según una o varias realizaciones de la invención.

Una secuencia original de las imágenes digitales $i0$ a in -701- es recibida como entrada en el codificador -70-. Cada imagen digital está representada por un conjunto de muestras, conocidas como píxeles.

Un flujo de bits -710- es emitido por el codificador -70- después de la implementación del proceso de codificación. El flujo de bits -710- comprende una serie de unidades o segmentos de codificación, comprendiendo cada segmento una cabecera del segmento para transmitir los valores de codificación de los parámetros de codificación utilizados para codificar el segmento y un cuerpo del segmento, que comprende los datos de vídeo codificados.

Las imágenes digitales $i0$ a in -701- de entrada son divididas en bloques de píxeles por el módulo -702-. Los bloques corresponden a porciones de la imagen y pueden tener tamaños variables (por ejemplo, 4x4, 8x8, 16x16, 32x32, 64x64 píxeles). Se selecciona un modo de codificación para cada bloque de entrada. Se proporcionan dos familias de modos de codificación: modos de codificación basados en codificación de predicción espacial (predicción INTRA) y modos de codificación basados en predicción temporal (codificación Inter, Merge, SKIP). Se prueban los posibles modos de codificación.

El módulo -703- implementa un proceso de predicción INTRA, en el que se predice el bloque a codificar determinado mediante un predictor calculado a partir de los píxeles del entorno de dicho bloque a codificar. Se codifica una indicación del predictor INTRA seleccionado y la diferencia entre el bloque determinado y su predictor, para proporcionar un residuo si se selecciona la codificación INTRA.

La predicción temporal se implementa mediante un módulo -704- de estimación de movimiento y un módulo -705- de compensación del movimiento. En primer lugar se selecciona una imagen de referencia de entre un conjunto de imágenes de referencia -716-, y se selecciona mediante el módulo -704- de estimación de movimiento una parte de la imagen de referencia, denominada asimismo parte de imagen o área de referencia, que es el área más próxima al bloque determinado a codificar. El módulo -705- de compensación del movimiento predice a continuación el bloque a codificar utilizando el área seleccionada. La diferencia entre el área de referencia seleccionada y el bloque determinado, denominada asimismo bloque residual, es calculada por el módulo -705- de compensación del movimiento. El área de referencia seleccionada es indicada por un vector de movimiento.

De este modo, en ambos casos (predicción espacial y temporal), se calcula un residuo restando la predicción del bloque original.

En la predicción INTRA implementada por el módulo -703-, se codifica una dirección de predicción. En la predicción temporal, se codifica, por lo menos, un vector de movimiento.

La información relativa al vector de movimiento y el bloque residual es codificada si se selecciona la predicción Inter.

Para reducir más la tasa de bits, suponiendo que el movimiento es homogéneo, se codifica el vector de movimiento mediante su diferencia con respecto a un predictor del vector de movimiento. Los predictores del vector de movimiento de un conjunto de predictores de la información de movimiento se obtienen a partir del campo -718- de vectores de movimiento mediante un módulo -717- de codificación y predicción del vector de movimiento.

5 El codificador -70- comprende, además, un módulo de selección -706- para seleccionar el modo de codificación aplicando un criterio de coste de codificación, tal como un criterio de tasa-distorsión. Para reducir más las redundancias se aplica una transformada (tal como una DCT) mediante el módulo de transformación -707- al bloque residual, los datos transformados obtenidos son cuantificados a continuación por el módulo de cuantificación -708- y sometidos a codificación entrópica mediante el módulo de codificación entrópica -709-. Finalmente, el bloque residual codificado del bloque actual que está siendo codificado se introduce en el flujo de bits -710-.

15 El codificador -70- lleva a cabo asimismo la descodificación de la imagen codificada para producir una imagen de referencia para la estimación del movimiento de las imágenes posteriores. Esto permite que el codificador y el descodificador que reciben el flujo de bits tengan los mismos cuadros de referencia. El módulo -711- de cuantificación inversa lleva a cabo la cuantificación inversa de los datos cuantificados, seguida por una transformada inversa mediante el módulo de transformación inversa -712-. El módulo de predicción INTRA inversa -713- utiliza la información de la predicción para determinar qué predictor utilizar para un bloque determinado y el módulo de compensación del movimiento inversa -714- suma en realidad el residuo obtenido por el módulo -712- al área de referencia obtenida a partir del conjunto de imágenes de referencia -716-.

A continuación se aplica el filtrado posterior mediante el módulo -715- para filtrar el cuadro de píxeles reconstruido. En las realizaciones de la invención, se utiliza un filtro de bucle de SAO en el que se añaden los desplazamientos de compensación a los valores de píxel de los píxeles reconstruidos de la imagen reconstruida.

25 La **figura 8** es un diagrama de flujo que muestra las etapas del proceso de filtrado de bucle según, por lo menos, una realización de la invención. En una etapa inicial -801-, el codificador genera la reconstrucción de todo el cuadro. A continuación, en la etapa -802- se aplica un filtro anti-efecto bloque a esta primera reconstrucción para generar una reconstrucción sin bloques -803-. El objetivo del filtro anti-efecto bloque es eliminar los artefactos de bloque generados por la cuantificación de residuos y la compensación del movimiento de bloques o la predicción INTRA de bloques. Estos artefactos son visualmente importantes a bajas tasas de bits. El filtro anti-efecto bloque funciona para suavizar los límites de los bloques según las características de los dos bloques contiguos. Se tienen en cuenta el modo de codificación de cada bloque, los parámetros de cuantificación utilizados para la codificación de residuos y las diferencias de los píxeles contiguos en el límite. Se aplica el mismo criterio/clasificación a todos los cuadros y no se transmiten datos adicionales. El filtro anti-efecto bloque mejora la calidad visual del cuadro actual eliminando los artefactos de efecto bloque y mejora asimismo la estimación del movimiento y la compensación del movimiento para los siguientes cuadros. De hecho, se eliminan las altas frecuencias del artefacto de bloques, y por lo tanto estas altas frecuencias no necesitan ser compensadas con la textura residual de los siguientes cuadros.

40 Después del filtro anti-efecto bloque, se filtra la reconstrucción sin bloques mediante un filtro de bucle de desplazamiento adaptativo de muestras (SAO) en la etapa -804- en base a una clasificación de píxeles -814- determinada según las realizaciones de la invención. El cuadro resultante -805- puede ser filtrado a continuación con un filtro de bucle adaptativo (ALF) en la etapa -806- para generar el cuadro reconstruido -807- que se presentará y se utilizará como un cuadro de referencia para los siguientes cuadros Inter.

45 En la etapa -804- cada píxel del área del cuadro se clasifica en una clase de la clasificación determinada según su valor de píxel. Una clase corresponde a un intervalo determinado de valores de píxel. El mismo valor de desplazamiento de compensación se añade al valor de píxel de todos los píxeles que tienen un valor de píxel dentro del intervalo determinado de valores de píxel.

50 La determinación de la clasificación de los píxeles para el filtrado de desplazamiento adaptativo de muestras se explicará en más detalle a continuación haciendo referencia a cualquiera de las **figuras 10 a 17**.

55 La **figura 9** muestra un diagrama de bloques de un descodificador -90- que puede ser utilizado para recibir datos desde un codificador según una realización de la invención. El descodificador es representado mediante módulos conectados, estando adaptado cada módulo para implementar, por ejemplo en forma de instrucciones de programa a ejecutar por la CPU -611- del dispositivo -600-, una etapa correspondiente de un procedimiento implementado por el descodificador -90-.

60 El descodificador -90- recibe un flujo de bits -901- que comprende unidades de codificación, estando compuesta cada una de ellas por una cabecera que contiene información sobre los parámetros de codificación y un cuerpo que contiene los datos de vídeo codificados. Tal como se ha explicado con respecto a la **figura 7**, los datos de vídeo codificados son sometidos a codificación entrópica y se codifican los índices de los predictores del vector de movimiento, para un bloque determinado, en un número predeterminado de bits. Los datos de vídeo codificados recibidos se descodifican entrópicamente mediante el módulo -902-. Los datos de residuo son descuantificados a continuación por el módulo -903- y, a continuación, se aplica una transformada inversa mediante el

módulo -904- para obtener los valores de píxel.

Los datos de modo que indican el modo de codificación también son descodificados entrópicamente y en base al modo, se lleva a cabo una descodificación de tipo INTRA o una descodificación de tipo INTER en los bloques codificados de los datos de la imagen.

En el caso del modo INTRA, se determina un predictor INTRA mediante el módulo de predicción INTRA inversa -905- en base al modo de predicción INTRA especificado en el flujo de bits.

Si el modo es INTER, se extrae la información de predicción del movimiento del flujo de bits para encontrar el área de referencia utilizada por el codificador. La información de la predicción del movimiento está compuesta por el índice del cuadro de referencia y el residuo del vector de movimiento. El predictor del vector de movimiento se suma al residuo del vector de movimiento para obtener el vector de movimiento mediante el módulo de descodificación del vector de movimiento -910-.

El módulo de descodificación del vector de movimiento -910- aplica la descodificación del vector de movimiento para cada bloque actual codificado por la predicción de movimiento. Una vez se ha obtenido un índice del predictor del vector de movimiento para el bloque actual, se puede descodificar el valor real del vector de movimiento asociado con el bloque actual y ser utilizado para aplicar la compensación del movimiento inversa mediante el módulo -906-. La parte de la imagen de referencia indicada por el vector de movimiento descodificado se extrae de una imagen de referencia -908- para aplicar la compensación inversa de movimiento -906-. Los datos -911- del campo del vector de movimiento se actualizan con el vector de movimiento descodificado para ser utilizados por la predicción inversa de los siguientes vectores de movimiento descodificados.

Finalmente, se obtiene un bloque descodificado. Se aplica un filtrado posterior mediante el módulo de filtrado posterior -907- de manera similar al módulo de filtrado posterior -715- aplicado al codificador, tal como se describe haciendo referencia a la **figura 7**. Finalmente, el descodificador -90- proporciona una señal de vídeo descodificada -909-.

La **figura 10** es un diagrama de flujo que muestra las etapas de un procedimiento -que no incorpora la presente invención- para clasificar los píxeles reconstruidos de una imagen para la aplicación de desplazamientos de compensación. En este procedimiento, las clases para la clasificación de los píxeles reconstruidos del área del cuadro según su valor de píxel son determinadas en base a la distribución estadística de los valores de píxel reconstruidos del área del cuadro. El centro, el intervalo útil y la cantidad de píxeles por clase se determinan en base a la distribución de los valores de píxel. En este procedimiento, el descodificador puede aplicar exactamente el mismo proceso que el descodificador para la segmentación de la distribución.

En una etapa inicial del proceso, el módulo -1002- escanea un área del cuadro -1001- actual para determinar la distribución estadística de los valores de píxel del área del cuadro -1001- y para generar un histograma -1003- correspondiente. En una implementación particular, este proceso implica la actualización de una tabla que contiene el número de píxeles para cada valor de píxel, es decir, para cada valor de píxel se tabula el número de píxeles que tienen dicho valor de píxel. La tabla contiene un número de celdas igual a MAX , el valor de píxel máximo determinado según la expresión $Max = 2^{Profundidad_{debit}-1}$, en base a la profundidad de bits de los píxeles.

El módulo -1004- determina a continuación el centro del histograma -1003- generado. El intervalo útil de los valores de píxel del histograma es determinado a continuación por el módulo -1006- según la distribución de los valores de píxel representados en el histograma -1003- y, en su caso, en base al centro del histograma. Finalmente, se determinan las clases equiprobables que definen los intervalos de los valores de píxel. De este modo, se proporciona una tabla -1009- que contiene el intervalo de los valores de píxel de cada clase o, alternativamente, una tabla que contiene los valores de píxel de cada píxel. La determinación de las clases equiprobables puede depender de un número de clases -1000- predeterminado.

En una etapa -1004- se pueden utilizar varios algoritmos para determinar el centro del histograma -1003- generado. Por ejemplo, se pueden encontrar el valor mínimo Min_{Hist} y el valor máximo Max_{Hist} del histograma. Para encontrar el valor mínimo Min_{Hist} , las celdas del histograma $Hist_k$ son escaneadas desde el valor de píxel 0 hasta la primera celda $Hist_k$ del histograma distinta de 0. Y para encontrar Max_{Hist} , se escanean las celdas en orden inverso (desde el valor de píxel máximo MAX hasta la primera celda del histograma $Hist_k$ distinta de 0). El centro del histograma $Centro_{Hist}$ se calcula como sigue:

$$Centro_{Hist} = (Max_{Hist} - Min_{Hist}) / 2 + Min_{Hist}$$

En un ejemplo alternativo, se considera que el centro del histograma es el centro medio ponderado de la distribución. Si se considera que el valor de la celda del histograma $Hist_k$ es el número de píxeles que tienen el valor

$$Centro_{Hist} = \frac{\sum_{k=0}^{MAX} k \times Hist_k}{N}$$

donde N es el número de píxeles en el área del cuadro actual.

- 5 En la etapa -1006-, una técnica potencial para determinar el intervalo útil del histograma generado es seleccionar Min_{Hist} y Max_{Hist} descritos anteriormente para ambos extremos del intervalo útil.

En otro ejemplo, el valor mínimo del histograma $Min_{Intervalo}$ se determina escaneando desde 0 hasta el primer $Hist_k$ que tiene un valor superior a un umbral α . Del mismo modo, $Max_{Intervalo}$ se determina mediante el escaneado inverso desde el valor de píxel máximo MAX hasta el primer $Hist_k$ que es superior a un umbral α . El umbral α puede ser un valor predeterminado. Alternativamente, el umbral α puede depender del número de píxeles del área del cuadro y/o del tipo de componente de la señal de entrada (croma y luma).

15 En una realización concreta, se puede considerar que el número de clases es conocido en los lados del codificador y del descodificador. El número de clases de los valores de píxel pueden depender, por ejemplo, del número de píxeles en el área del cuadro actual según cada componente (luma, croma U y V).

Para generar clases equiprobables, se define el número de píxeles $NbPix_{Intervalo}$ en el intervalo útil -1007-. El número de píxeles en el intervalo útil $NbPix_{Intervalo}$ se determina escaneando cada celda del histograma $Hist_k$ desde $k=Min_{Intervalo}$ hasta $k=Max_{Intervalo}$. A continuación, se divide el número determinado de píxeles en el intervalo útil, $NbPix_{Intervalo}$, por el número de clases -1000- para determinar el número óptimo de píxeles en cada clase $NbPix_{clases}$.

La **figura 11** es un diagrama de flujo que muestra las etapas de un algoritmo para determinar las clases equiprobables del procedimiento de la figura 10. En una etapa inicial -1101-, el número de clases j se fija a 0 y el valor de píxel actual k se fija a $Min_{Intervalo}$. Para una clasificación equiprobable, una clase se identifica por su intervalo de los valores de píxel. De este modo, el número de clase j se identifica por su intervalo [Min_j ; Max_j] desde su valor mínimo de píxel Min_j hasta su valor máximo de píxel Max_j .

En la etapa -1103-, el valor mínimo del píxel Min_j de la clase actual con índice j se fija al valor de píxel actual k. A continuación, $SumNbPix_j$ se fija a 0 en la etapa -1104-. $SumNbPix_j$ corresponde al número de píxeles en el intervalo j. A continuación, el número de píxeles que tienen el valor de píxel k ($Hist_k$) se suma a $SumNbPix_j$ en la etapa -1105-. En la etapa -1106- se determina si la suma del número de píxeles para la clase actual j $SumNbPix_j$ es superior o no al número de píxeles de las clases $NbPix_{clases}$. Si no se satisface esta condición, el valor k se incrementa en la etapa -1107- y el número de píxeles $Hist_k$ para el valor de píxel k se suma a $SumNbPix_j$ en la etapa -1105-. Si se determina que $SumNbPix_j > NbPix_{clases}$ o si k alcanza el valor máximo del intervalo útil $Max_{Intervalo}$, el valor máximo para la clase actual j es igual al valor actual de k en la etapa -1108-. En esta fase, se define la clase j - es decir, se ha determinado el intervalo [Min_j ; Max_j] de la clase j. La variable k se incrementa en la etapa -1109- para evitar obtener el mismo valor de píxel en más de una clase. Además, la variable j se incrementa asimismo en la etapa -1110- para definir el intervalo de los valores de píxel para la siguiente clase. Si la variable j es superior al número de clases $NbPix_{clases}$, entonces se puede considerar que todas las clases se han definido en la etapa -1112-.

Como consecuencia el codificador determinará el valor de desplazamiento para cada clase j, tal como se describe en relación con la figura 3 y se transmite al descodificador. El codificador y el descodificador filtrarán el área del cuadro, tal como se describe haciendo referencia a la figura 4.

Se debe observar que en el algoritmo de la figura 11, el número de clases $NbClases$ no depende de los valores de píxel debido a que el número de clases es predeterminado en base a un valor de sintaxis. Por consiguiente, en este algoritmo, el análisis del desplazamiento de banda del SAO es independiente de la descodificación de los demás cuadros. Se debe observar que el análisis del desplazamiento de banda de SAO incluye el análisis de cada desplazamiento.

En otro algoritmo para determinar la clasificación equiprobable, el número de clases se puede determinar de acuerdo con la distribución de los valores de píxel en el histograma generado. De hecho, cuando la amplitud del intervalo útil es alta o baja, el número de clases debería tener un impacto en la eficiencia de la codificación. Por consiguiente, se puede proporcionar una clasificación más adaptable para determinar el número de píxeles en cada clase así como el número de los valores de píxel.

La **figura 12** es un diagrama de flujo que muestra las etapas de un algoritmo adicional para proporcionar una clasificación más adaptable. Este diagrama de flujo está basado en el diagrama de flujo del algoritmo de la figura 11, donde los módulos que terminan con los mismos números realizan funciones equivalentes. Sin embargo, los módulos de decisión -1206- y -1211- de este algoritmo funcionan con condiciones de prueba diferentes a las condiciones de prueba manejadas por los módulos correspondientes -1106- y -1111- de la figura 11.

En este algoritmo, el módulo de decisión -1206- detiene el bucle en base a los valores k y selecciona Max_j para la clase j , si $SumNbPix_j > NbPix_{clases}$ o si k alcanza el valor máximo del intervalo útil $Max_{Intervalo}$ o si $k - Min_j$ es estrictamente menor que el intervalo máximo para una clase ($MaxClase_{Intervalo}$). $k - Min_j$ corresponde al número de valores de píxel en el intervalo actual de la clase j . $MaxClase_{Intervalo}$ es un número máximo predeterminado de valores de píxel en el intervalo. Este intervalo puede depender de la profundidad de bits, del número de píxeles N del área del cuadro y del tipo de señal (luma, croma U y V). Por ejemplo, cuando la profundidad de bits es 8, $MaxClase_{Intervalo}$ para el componente luma podría ser igual a 16, tal como en una implementación de la HEVC.

La ventaja del algoritmo de la figura 12 en comparación con el de la figura 11, es la de su eficacia de codificación para una distribución de los valores de píxel con gran amplitud. Este algoritmo es más adaptable a la distribución.

Se puede observar que en este algoritmo, el número determinado de clases depende de los valores de píxel, y por lo tanto el análisis del cuadro actual depende de la decodificación de los cuadros anteriores. Para ser más robusto con respecto a los errores de transmisión, el número de clases $NbClases$ se introduce en el flujo de bits. La transmisión de dichos datos tiene un impacto insignificante sobre la eficiencia de la codificación.

La principal ventaja de los algoritmos de clasificación de las figuras 10 a 12 es que la clasificación está adaptada a la distribución de los valores de píxel. Además del centro, el intervalo útil y el tamaño de cada clase y su cantidad no tienen que ser transmitidos. Por consiguiente, al igual que en la implementación de la HEVC conocida, ningún dato adicional además de los datos representativos del desplazamiento de cada clase tiene que ser transmitido para la clasificación determinada.

Una realización de la invención para determinar una clasificación y que implica la señalización de parámetros de la clasificación se describirá a continuación haciendo referencia a la **figura 13**. El objetivo de la presente realización de la clasificación es proporcionar una clasificación óptima de la distribución de los valores de píxel. La diferencia en comparación con los algoritmos descritos anteriormente es que la clasificación no está determinada directamente en base a la distribución de los valores de píxel sino en base a un criterio de tasa-distorsión. En la presente realización, el codificador selecciona la clasificación (grupo de clases), mejor adaptada a la distribución de los valores de píxel, de entre las potenciales clasificaciones predefinidas (grupos de clases predeterminados). Esta selección se basa en el criterio de tasa-distorsión. Al igual que en los algoritmos descritos anteriormente, se determina el centro, el intervalo útil y el tamaño de las clases del histograma generado que representa la distribución de los valores de píxel. En la presente realización estos parámetros son transmitidos en el flujo de bits. Para minimizar el impacto de la transmisión de dichos datos, los tamaños de las clases y los intervalos relacionados son seleccionados entre valores predefinidos. Por consiguiente, el codificador introduce en el flujo de bits el centro de la clasificación seleccionada, el índice relacionado con la clasificación seleccionada y los tamaños de las clases de la clasificación.

Para proporcionar una adaptación a la distribución de los valores de píxel, se definen varios tamaños de intervalos de valores de píxel, tal como se representa en la figura 13. En la figura 13, el intervalo completo de valores de píxel se divide en 32 subintervalos. Para un primer grupo de clases relacionadas con los valores de píxel situados en el centro del intervalo de los valores de píxel, se representan 4 ejemplos -1301-, -1302-, -1303-, -1304-. El primer ejemplo -1301- contiene 26 intervalos de los 32 intervalos potenciales. De este modo, el intervalo útil -1301- representa el 13/16 del intervalo completo. Del mismo modo, -1302- representa únicamente 8 intervalos de los 32 intervalos potenciales, es decir, 1/4 del intervalo completo, -1303- representa 1/8 de todo el intervalo y -1304- 1/16 de todo el intervalo. Para el esquema propuesto, se pueden considerar todos los tamaños posibles desde todo el intervalo completo hasta un intervalo correspondiente solamente a un valor de píxel. El número de intervalos útiles posibles debe ser predeterminado en función de la eficiencia de la codificación, o predeterminado para el número de píxeles en el área del cuadro.

La **figura 13** muestra asimismo varios ejemplos de tamaños para el segundo grupo de clases en relación con los valores de píxel situados en los bordes del intervalo de los valores de píxel. El segundo grupo incluye dos subgrupos de clases, uno situado hacia cada borde del histograma. Los ejemplos -1305-, -1306-, -1307-, -1308- representan respectivamente el mismo número de valores de píxel que los ejemplos -1301-, -1302-, -1303-, -1304- del primer grupo.

En ciertas realizaciones de la invención, el tamaño de las clases, es decir, el intervalo de los valores de píxel por clase, no es fijo, en comparación con los procedimientos de la técnica anterior. La **figura 14** muestra ejemplos de varios tamaños -1401- a -1406-. En este ejemplo, los tamaños de las clases son desde 32 píxeles -1401- hasta solamente 1 píxel -1406-. Estos tamaños de clases se podrían combinar con todos los posibles intervalos útiles tal como se ha descrito anteriormente en relación con la figura 13. En esta realización, se considera que todas las clases tienen el mismo tamaño para un intervalo específico de valores de píxel. Por lo tanto, para un grupo, se introducen en el flujo de bits los datos representativos del tamaño del intervalo útil y el tamaño de las clases.

En otra realización, los tamaños de las clases para un intervalo útil determinado son adaptados según la posición de la clase en el intervalo útil. Más precisamente, los tamaños de las clases son adaptados a la distribución de los valores de píxel. En otra realización, estos tamaños son predeterminados para cada intervalo útil según la

distribución de valores de píxel. De hecho, el histograma de la distribución de los valores de píxel corresponde, en general, a una distribución gaussiana. Cuanto más cerca del centro del histograma esté un valor de píxel, más cantidad de píxeles tendrán un valor de píxel cercano a este valor. Esto significa que una celda de histograma $Hist_k$ próxima al centro tiene un valor (número de píxeles correspondientes) mayor que una celda de histograma $Hist_k$ en ambos extremos del intervalo útil del histograma.

La **figura 15** muestra ejemplos de las dos realizaciones descritas para los tamaños de las clases. El ejemplo -1501- representa un tamaño fijo de 8 valores de píxel para un intervalo útil de 32 valores de píxel. -1502- representa un tamaño fijo de 4 valores de píxel para el mismo tamaño de intervalo útil.

El ejemplo -1503- muestra la otra realización para tamaños adaptativos de las clases para un intervalo actual de 32 valores de píxel. En este ejemplo, las clases en ambos extremos del intervalo útil son mayores, es decir, tiene un intervalo más amplio de valores de píxel que las clases del centro con 8 valores de píxel y 2 valores de píxel, respectivamente. Entre estas clases, las otras 2 clases tienen un intervalo de 4 valores de píxel.

Los tamaños de las clases para el segundo grupo pueden estar adaptados asimismo a la distribución de los valores de píxel. El objetivo del segundo grupo de la actual implementación de la HEVC es explotar solamente los dos extremos del histograma. De hecho, ambos extremos del histograma contienen los valores extremos que a menudo están relacionados con altas frecuencias en las que el error (debido a la codificación con pérdidas) es habitualmente mayor en comparación con bajas frecuencias. Del mismo modo que en el primer grupo, se pueden probar varios tamaños de clases para los intervalos útiles del segundo grupo. En este caso, para los dos subgrupos del segundo grupo, las subdivisiones -1501- y -1502- pueden ser comparadas con el criterio de tasa-distorsión.

Además, se puede aplicar la realización en la que los tamaños de las clases están adaptados. El ejemplo -1504- muestra los tamaños de clases adaptados propuestos para el primer intervalo (izquierda) del segundo grupo. Y el ejemplo -1505- muestra los tamaños de clases adaptados propuestos para el segundo subgrupo (derecha) del segundo grupo. En este caso, las clases contienen más valores de píxel en ambos extremos que las clases cerca del centro.

El objetivo del segundo grupo es explotar ambos extremos del histograma; por consiguiente, en ocasiones es útil utilizar una adaptación inversa de los tamaños para el segundo grupo. En este caso, el ejemplo -1504- se utiliza para el segundo subgrupo (derecha) y el ejemplo -1505- se utiliza para el primer subgrupo (izquierda) del segundo grupo. En esta realización, las clases contienen menos valores de píxel en ambos extremos que las clases cercanas al centro. En este caso, el objetivo no es producir una clasificación de clases equiprobable sino encontrar una mejor segmentación de ambos extremos del segundo grupo.

Dado que la distribución estadística de los valores de píxel no está necesariamente centrada en el medio del intervalo completo de los valores de píxel, se debería determinar un centro de la distribución en base al intervalo útil y ser transmitido en el flujo de bits con los datos de la imagen. La **figura 16** muestra un ejemplo de un intervalo completo con diferentes posiciones de centro para un intervalo útil correspondiente a un cuarto del intervalo completo. Al contrario que en el ejemplo -1302- de la **figura 13**, para los cuatro ejemplos de la **figura 16**, -1601-, -1602-, -1603-, -1604-, el centro del intervalo útil no está situado en el centro del intervalo completo. Esta solución permite adaptar la clasificación seleccionada a la distribución de los valores de píxel.

El centro determinado se puede codificar a continuación para su transmisión en el flujo de bits. Se pueden contemplar varias técnicas para codificar los datos.

Si se considera que la profundidad de bits del área del cuadro actual es de 8 bits, el número de posiciones que podrían considerarse para el valor del centro corresponde a 256 menos el tamaño del mínimo intervalo útil. Por ejemplo, en comparación con la figura 13, el tamaño mínimo del intervalo útil es igual a 2, y estas 2 clases pueden contener, por lo menos 1 píxel. Por lo tanto, para este ejemplo específico, el centro puede adoptar un valor entre 1 y 254, de modo que se pueden considerar 254 posiciones para el centro.

Otra solución es cuantificar el valor del centro. En una realización, el centro se codifica según el tamaño de las clases. De este modo, por ejemplo, si el tamaño de las clases (o el tamaño mínimo de todas las clases de un intervalo útil actual cuando se utiliza el esquema de tamaño de clases adaptado) es igual al valor de un píxel, el centro no se cuantifica y puede ser todas las posibles posiciones de centro para el intervalo útil actual. Si el tamaño de las clases es 16 valores de píxel, tal como se representa en la **figura 16**, únicamente se consideran los valores de píxel cada 16 valores de píxel. Por lo tanto, en la **figura 16**, el centro para los ejemplos -1601-, -1602-, -1603- y -1604- es 9, 23, 27 y 6, respectivamente. En otra realización, pueden ser consideradas solamente las posiciones del centro iguales a un múltiplo del tamaño máximo de las clases definidas en el algoritmo. De este modo, el centro es igual a un valor de píxel dividido por el tamaño máximo de las clases. Esto ofrece una reducción en términos del número de bits a transmitir.

Además, teóricamente, el centro más probable es el centro del intervalo completo. De este modo, los datos transmitidos para determinar la posición del centro en el lado del descodificador es la diferencia entre el centro del

intervalo completo y el centro del intervalo útil de la clasificación actual. Por lo tanto, por ejemplo en la **figura 16**, los datos transmitidos con respecto al centro para los ejemplos -1601-, -1602-, -1603-, -1604- son respectivamente $16-9=7$, $16-23=-7$, $16-27=-11$, $16-6=10$.

5 Para el segundo grupo, no es necesario codificar el centro del histograma. De este modo, se pueden considerar varios esquemas para codificar el desplazamiento de los dos subgrupos para el segundo grupo. Las realizaciones propuestas en la cuantificación del valor del centro descritas para el primer grupo se pueden extender fácilmente a las realizaciones propuestas para el segundo grupo.

10 En las realizaciones de la invención, la posición del intervalo útil (clasificación seleccionada) puede ser especificada con la misma precisión o granularidad en el intervalo completo, es decir, independientemente de la posición de la clasificación dentro del intervalo completo. Este es el caso en los ejemplos -1601- a -1604- mostrados en la figura 16, donde las posiciones (posiciones de centro) son 9, 23, 27 y 6. El intervalo completo está identificado de 0 a 32. Existen 32 posiciones posibles y la granularidad es la misma en todo el intervalo.

15 No obstante, es posible asimismo, tal como se muestra en las figuras 19a y 19b proporcionar más posiciones posibles en una parte del intervalo completo que en otra parte del intervalo completo. En otras palabras, la granularidad de la posición varía en función de donde esté la clasificación dentro del intervalo completo. Estas realizaciones proponen una cuantificación desigual del intervalo completo (en este caso, identificado de 0 a 32) con una granularidad variable para posicionar más precisamente el centro de la clasificación (intervalo útil) en las partes más importantes (o más probables) del intervalo completo. Asimismo, la cuantificación desigual permite que el número de bits necesarios para señalar la posición de la clasificación esté limitado, aún proporcionando una precisión adecuada en las partes importantes del intervalo completo. Esta granularidad más fina se podría aplicar, por ejemplo, en medio del intervalo completo, tal como se representa en la figura 19a. En esta figura, las posibles posiciones del centro corresponden a los índices que están representados por una línea continua en negrita. El intervalo entre dos posibles posiciones de centro es menor en el medio del intervalo completo que en los extremos. De este modo, la posición del centro se puede ajustar de manera más precisa en el medio del intervalo completo que en los extremos del intervalo completo.

30 En la figura 19b el intervalo entre dos posiciones de centro posibles es menor en ambos extremos del intervalo completo que en el medio. Por ejemplo, esta realización puede ser particularmente útil en caso de tener valores de muestra importantes en los valores extremos de la distribución.

Más en general, se podría aplicar una cuantificación más fina en cualquier punto del intervalo completo.

35 Cuando se utiliza cuantificación variable, tal como se ha descrito anteriormente, el intervalo de la clasificación (tamaño del intervalo útil) puede ser fijo para todas las posiciones. Por ejemplo, el intervalo de la clasificación puede comprender cuatro clases, cada una compuesta por 8 valores de píxel.

40 Asimismo, es posible hacer que el intervalo de la clasificación/los tamaños de las clases varíen con la posición, de manera que en la figura 19a, el intervalo de la clasificación es, por ejemplo, de 8 valores de píxel en las posiciones 12 a 20, 16 valores de píxel en las posiciones 10 y 26, y 32 valores de píxel en las posiciones 2 y 28.

45 La cuantificación variable, tal como se describe en este caso, puede ser utilizada independientemente del procedimiento aplicado para determinar el intervalo de la clasificación. Este procedimiento puede utilizar, por ejemplo, las propiedades de la distribución estadística de los valores de muestra o utilizar un criterio de tasa-distorsión.

50 La cuantificación variable podría estar predeterminada tanto en el codificador como en el descodificador. Por ejemplo, el codificador y el descodificador podrían asignar índices a las posibles posiciones del centro (o posiciones de la izquierda), por ejemplo, en la figura 19a la posición 2 es el índice 0, la posición 6 es el índice 1, la posición 10 es el índice 2, la posición 12 es el índice 3, la posición 13 es el índice 4, etc. A continuación, es suficiente que el codificador transmita al descodificador el índice de la clasificación seleccionada. Alternativamente, la información sobre la cuantificación variable se podría determinar en el codificador y ser señalada al descodificador mediante un flujo de bits.

55 En un ejemplo particular que no incorpora la invención, se puede considerar que el centro del histograma es siempre el centro del intervalo completo. De este modo, en este caso solamente se considera un desplazamiento. Ambos grupos se escalan en el centro con el mismo desplazamiento. Por consiguiente, sólo es necesario codificar un dato: el desplazamiento del primer intervalo del segundo grupo. Los ejemplos -1701-, -1702-, -1703- y -1704- de la **figura 17** son ejemplos de dichos desplazamientos. En los ejemplos -1701-, -1702-, -1703-, -1704- los desplazamientos son respectivamente 4, 6, 10 y 0. El desplazamiento puede estar codificado directamente sin predicción.

60 En **otro ejemplo**, ambos subgrupos del segundo grupo tienen una posición independiente en el intervalo completo, tal como se representa en los ejemplos -1705-, -1706-, -1707- y -1708-. Se pueden considerar dos modos de codificación.

En el primero, el centro de un primer grupo inexistente es codificado con el tamaño del intervalo útil de este primer grupo inexistente.

5 El segundo modo de codificación de ambos grupos independientemente es transmitir 2 desplazamientos desde los dos extremos del intervalo completo (uno para cada grupo). De este modo, para los ejemplos -1705-, -1706-, -1707- y -1708-, el desplazamiento transmitido es respectivamente 11 y $32-28=4$ para -1705-, 21 y 0 para -1706-, 3 y $32-16=32$ para -1707-, 7 y $32-31=1$ para -1708-.

10 La **figura 18** es un diagrama de flujo que muestra las etapas de un algoritmo de selección de la tasa-distorsión según una realización de la invención. Con fines explicativos simplificados, se considera solamente la selección para el primer grupo sin tamaños de clases adaptados. La selección para las demás realizaciones descritas anteriormente se puede adaptar fácilmente.

15 En una etapa inicial -1801-, se calculan las estadísticas del área del cuadro actual. Esto implica la determinación de las variables $Hist_k$ y Sum_k para todos los valores de píxel k . $Hist_k$ corresponde al número de píxeles que tienen un valor de píxel igual al valor k y Sum_k corresponde a la suma de las diferencias entre todos los píxeles que tienen un valor de píxel igual al valor k y sus valores de píxel originales. El algoritmo incluye 3 bucles en tres parámetros: el tamaño de las clases S , el tamaño del intervalo R y el centro C . El primer bucle comprueba cada tamaño de clase posible en la etapa -1803-. Por ejemplo, el tamaño definido en la **figura 14**. El desplazamiento para cada subintervalo del intervalo completo se calcula en la etapa -1804-. Por ejemplo, si la profundidad de bits es 8 y el tamaño de las clases es 16, entonces se calcula la distorsión y el desplazamiento para los 32 intervalos posibles en el intervalo completo. Por propiedades, el desplazamiento y la distorsión son calculados mediante la combinación lineal de $Hist_k$ y Sum_k para todos los valores de k en el intervalo actual. A continuación, para cada intervalo R -1805- posible, y cada centro C -1806- posible, se evalúa un coste de la tasa-distorsión en la etapa -1807-. Esta evaluación se basa en el criterio de tasa-distorsión. Cuando se prueban todos los centros C -1808-, todos los intervalos -1809- y todos los tamaños -1810-, se seleccionan los mejores parámetros S , R , C en la etapa -1811- en base al mejor coste de la tasa-distorsión. Las ventajas de este segundo esquema para generar una clasificación equiprobable incluye una reducción de la complejidad y una mejora en la eficiencia de la codificación. La selección de la clasificación del centro, intervalo y tamaño de las clases ofrece una selección de la tasa-distorsión óptima en comparación con las realizaciones en las que la clasificación está basada en la distribución estadística de los valores de píxel. Por supuesto, esta realización proporciona una mejora en términos de eficiencia de la codificación en comparación con la actual implementación de la HEVC. Este esquema es **menos** complejo en el lado del descodificador comparado con el anterior dado que no es necesario determinar la distribución de los píxeles en el descodificador. Además, este esquema puede ser menos complejo que las técnicas conocidas de la HEVC debido a que se utilizan menos clases en algunos grupos.

El algoritmo representado en la **figura 18** realiza una selección completa basada en tasa-distorsión de todos los parámetros del desplazamiento de banda: el tamaño de las clases S , el intervalo R , la posición de un valor representativo del centro C . Para limitar la complejidad, es posible fijar algunos parámetros. En una implementación particular del algoritmo de la figura 18, el tamaño S y el intervalo R se fijan en valores determinados conocidos por el codificador y el descodificador. Por ejemplo S , podría representar 8 valores de píxel y R podría representar 32 valores de píxel correspondientes a 4 clases de 8 píxeles. Como consecuencia, el único parámetro a optimizar es el valor representativo del centro C .

45 Dado que las realizaciones de la invención tienen en cuenta el reparto de los valores de píxel en todo el intervalo de valores de píxel en la determinación de la clasificación de los píxeles, la clasificación se puede adaptar en correspondencia con diferentes distribuciones de valores de píxel. En particular, se puede adaptar la clasificación según el tipo de componente de los píxeles. Por ejemplo, en el caso de un conjunto de píxeles del componente croma, los valores de píxel tienden a ser menores en comparación con los valores de píxel de los píxeles luma croma. Además, los valores de píxel croma U tienen una distribución diferente a la de los valores de píxel croma V que tienen valores de píxel más concentrados y relativamente mayores. Además, en el caso de píxeles del componente croma, la distribución de los valores de píxel tiende a estar más concentrada alrededor de los valores pico de los píxeles en comparación con los píxeles luma croma que proporcionan una distribución extendida más ampliamente.

60 Tal como se ha visto anteriormente, para evitar la determinación de la distribución de los píxeles en el lado del descodificador, se transmiten los parámetros S , R y C en el flujo de bits además del tipo de SAO (no SAO, desplazamiento de borde o desplazamiento de banda) y los valores de desplazamiento de compensación. Cuando el tamaño de la clase y el intervalo son fijos, solamente se transmite C para permitir que un descodificador recupere el centro del intervalo.

En el caso de S y R fijos, una solución conocida para codificar los parámetros del SAO consiste en aplicar el pseudocódigo de la **figura 20A**, descrito en forma de diagrama de flujo en la **figura 21**.

65 El proceso se inicia con la determinación de los parámetros del SAO, incluyendo el tipo de SAO (almacenado en la

palabra de código `sao_type_idx`), el valor representativo del centro del intervalo útil (almacenado en la palabra de código `sao_band_position`) cuando se utiliza el tipo de desplazamiento de banda y los desplazamientos del SAO (almacenados en las palabras de código `sao_offset`). En la figura 20A, *cdx* representa el índice del componente de color al que se aplica el SAO, *rx* y *ry* representan la posición del área en que se aplica el SAO e *i* es el índice de la clase de los valores de muestra.

La codificación de los parámetros del SAO se inicia a continuación en la etapa -2003- con la codificación del tipo de SAO utilizando un código Exp Golomb sin signo (*ue(v)*) (es decir, un código de longitud variable sin signo). Si el tipo de SAO es de tipo 5 (*desplazamiento de banda*), la codificación continúa con la codificación del valor representativo de la posición del centro del intervalo útil utilizando un código de longitud fija sin signo de tamaño 5 (*u(5)*) en la etapa -2017-. A continuación, la codificación de los cuatro desplazamientos correspondientes a las cuatro clases contenidas en el intervalo se realiza de modo iterativo en las etapas -2019- a -2025-. En este caso, cada desplazamiento se codifica utilizando un código Exp Golomb con signo (*se(v)*) (es decir, un código de codificación de longitud variable con signo (VLC)). El proceso de codificación finaliza a continuación con la etapa -2027-.

Si el tipo de SAO no es un *desplazamiento de banda*, se comprueba primero si el tipo de SAO es un *no SAO* (*no SAO* significa que no se aplica ningún desplazamiento a las muestras involucradas). Si se ha seleccionado *no SAO*, el proceso de codificación se detiene en la etapa -2027-.

De lo contrario, se continúa con la codificación iterativa de los cuatro desplazamientos de borde en las etapas -2007- a -2013-. De nuevo, el proceso de codificación se detiene en la etapa -2027-.

Los códigos VLC se utilizan, generalmente, cuando el intervalo de valores a representar es relativamente elevado pero algunos valores de este intervalo son más probables que otros. Los valores más probables reciben a continuación un código corto, mientras que los valores menos probables reciben un código largo. El principal inconveniente de estos códigos es que provocan una complejidad de descodificación mayor que los códigos de longitud fija (FLC). De hecho, un código VLC debe leerse bit a bit dado que no se conoce el tamaño final del código mientras que todos los bits de un código FLC se pueden leer directamente dado que su tamaño es conocido.

En las **figuras 20B** y **22**, se propone una alternativa a este proceso de codificación sustituyendo los códigos VLC por los códigos FLC.

Este proceso de codificación se inicia con la etapa -2201- que es idéntica a la etapa -2001-. En la etapa -2203-, la codificación VLC de la palabra de código `sao_type_idx` se sustituye por una codificación FLC. En este caso, son necesarios 3 bits para codificar los 6 valores posibles del tipo de SAO (es decir, el tipo “no SAO”, los 4 tipos de “desplazamiento de borde” y el tipo de “desplazamiento de banda”). A continuación se comprueba si el tipo de SAO es “no SAO”. En este caso, no se codifica nada más y el proceso finaliza con la etapa -2215-. De lo contrario, se comprueba si el tipo de SAO es “desplazamiento de banda”. En caso afirmativo, se codifica el valor representativo de la posición del centro del intervalo en la palabra de código `SAO_band_position` en forma de un código FLC sin signo de tamaño 5. De hecho, en este ejemplo, con un tamaño de clase de 8 valores de muestra y un intervalo compuesto por cuatro clases, son posibles 28 posiciones diferentes para un intervalo completo de 256 valores.

A esta etapa le sigue la codificación de los cuatro desplazamientos del SAO en las etapas -2211- a -2213-. En este caso, un código FLC sustituye el código VLC de la etapa -2023- y -2011-. En lugar de utilizar un código VLC de un máximo de 5 bits que cubren los valores de desplazamiento enteros de -31 a 32, en este caso se utiliza un código FLC de 2 bits de tamaño que puede codificar únicamente 4 valores diferentes, en general (-2, -1, 1, 2). Reducir el número de valores posible tiene el efecto de concentrar la codificación en los valores de desplazamiento utilizados más frecuentemente.

El proceso se detiene después de la codificación del desplazamiento en la etapa -2215-.

Se debe observar que, en otra realización, el intervalo representado por los desplazamientos se puede ampliar codificando una cabecera de la imagen, una cabecera del segmento o una cabecera de la LCU, un factor de multiplicación a aplicar a los desplazamientos obtenidos por el código de 2 bits. Por ejemplo, con un factor de multiplicación igual a 4, los desplazamientos codificados (-2, -1, 1, 2), pasan a ser (-8, -4, 4, 8). El factor de multiplicación también puede estar estandarizado (fijo) o deducirse de otra LCU. Por ejemplo, se puede suponer que se aplica a la LCU actual el factor de multiplicación aplicable a una LCU anterior.

De manera similar en otra realización, se podría aplicar un valor de desplazamiento, codificado en una cabecera de la imagen, una cabecera del segmento o una cabecera de la LCU, a los desplazamientos obtenidos por el código de 2 bits. Por ejemplo, con un valor de desplazamiento de 5, los desplazamientos codificados (-2, -1, 1, 2) pasar a ser (3, 4, 6, 7). De nuevo, el valor de desplazamiento puede asimismo estar estandarizado (fijado) o deducirse de otra LCU. Por ejemplo, se puede suponer que se aplica a la LCU actual el valor de desplazamiento aplicable a una LCU anterior.

Las pruebas han mostrado que tener menos valores de desplazamiento posibles no reduce significativamente el

rendimiento del procedimiento del SAO. Parece que la pérdida inducida por la eliminación de algunos valores de desplazamiento se compensa con la eliminación del importante coste en tasa de bits de los valores de desplazamiento menos probables.

5 Las pruebas adicionales han mostrado asimismo que el número de valores de desplazamiento diferentes se puede reducir adicionalmente a 3 e incluso 2 desplazamientos (lo que requiere codificar únicamente un bit) sin una pérdida significativa en el rendimiento.

10 En la **figura 23**, se propone una mejora adicional del proceso de codificación. Se considera en este caso que los valores de desplazamiento utilizados en el caso del tipo de desplazamiento de borde pueden ser deducidos directamente del tipo de desplazamiento de borde. En este caso no se requiere ninguna codificación de los valores de desplazamiento de borde. Como recordatorio, se asocia cada tipo de desplazamiento de borde con 4 clases en función de la dirección de la señal, y cada clase tiene un valor de desplazamiento asociado. Esta realización está motivada por las pruebas que muestran que, en general, para un tipo de desplazamiento de borde determinado y una clase determinada, los valores de desplazamiento son próximos entre sí y, generalmente, iguales. Como consecuencia, se propone fijar para cada tipo de desplazamiento de borde un conjunto de 4 valores de desplazamiento. Por ejemplo, se propone la siguiente asociación:

- 20 - desplazamiento de borde vertical: (-2, -1, 1, 2)
- desplazamiento de borde horizontal: (-2, -1, 1, 3)
- desplazamiento de borde de la primera diagonal (-3, -2, -1, 1)
- desplazamiento de borde de la segunda diagonal (-1, 1, 2, 3)

25 Las etapas -2301- y -2303- de la figura 23 son idénticas a las etapas -2201- y -2203- ya explicadas haciendo referencia a la figura 22. En las etapas -2305- y -2307-, se comprueba si el tipo de SAO es “desplazamiento de borde” o “no SAO” respectivamente. En ambos casos, no se codifica ningún desplazamiento. En el caso en que el tipo de SAO es “desplazamiento de borde”, cuando se lee el valor a partir del tipo de desplazamiento de borde, un descodificador deducirá los valores de desplazamiento del tipo de desplazamiento de borde gracias a la asociación conocida con los valores de desplazamiento fijos.

30 En la realización de la figura 23, si el tipo de SAO es “desplazamiento de banda”, se codifica en la etapa -2309- el valor representativo de la posición del centro del intervalo y los cuatro valores de desplazamiento se codifican de modo iterativo en las etapas -2311- a -2317-. El proceso de codificación finaliza con la etapa -2319-.

35 En la realización de las **figuras 24 y 25** se aplica otra modificación al proceso de codificación descrito en las figuras 20A y 21. Tal como ya se ha mencionado en la realización anterior, se utiliza un código FLC de 5 bits para codificar la información representativa de la posición del centro del intervalo (`sao_band_position`), mientras que se utilizan únicamente 28 posiciones diferentes. En esta situación, cuatro códigos FLC con longitud de 5 bits cada uno siguen inutilizados. Se propone en este caso aprovechar estos cuatro códigos FLC sobrantes para eliminar la palabra de código utilizada para codificar el tipo de SAO. Se utilizará una nueva palabra de código, `SAO_band_position_and_EO` para codificar conjuntamente las posiciones del intervalo y los tipos de desplazamiento de borde. Esta nueva palabra de código tiene asimismo con una longitud de 5 bits.

45 Tal como es habitual, el proceso se inicia en la etapa -2501- con la definición de los parámetros del SAO. A este proceso le sigue en la etapa -2503- la codificación de un indicador de 1 bit de longitud (`SAO_LCU_flag`) que indica si se utiliza o no el SAO. Si no se utiliza el SAO (etapa -2505-), el proceso se detiene (etapa -2507-).

50 Si se utiliza el SAO, se comprueba en la etapa -2509- qué tipo de SAO se utiliza. Si el tipo de SAO es “desplazamiento de banda”, entonces en la etapa -2513- se utilizan los primeros 28 códigos de la palabra de código `SAO_band_position_and_EO` para codificar el valor representativo de la posición del centro del intervalo. Si el tipo de SAO es “desplazamiento de borde”, entonces en la etapa -2511- se utilizan los últimos cuatro códigos de la palabra de código `SAO_band_position_and_EO` para codificar el tipo de desplazamiento de borde (vertical, horizontal, primera diagonal o segunda diagonal). A las etapas -2511- o -2513- les sigue la codificación de los cuatro valores de desplazamiento en las etapas -2515- a -2521-.

55 A este respecto, aunque las palabras de código sobrantes se utilizan en la presente realización para codificar el tipo de desplazamiento de borde, se apreciará que las palabras de código sobrantes se pueden utilizar alternativamente para otros propósitos. Cualquier otra información que tenga que ser enviada del codificador al descodificador puede ser codificada utilizando las palabras de código sobrantes.

60 En la **figura 26** se propone otra realización para determinar los desplazamientos a aplicar en el caso del desplazamiento de banda. Esta realización está motivada por pruebas que muestran que en el caso del desplazamiento de banda, la mayoría de los desplazamientos tienen amplitudes bajas en valor absoluto. De hecho, los valores de desplazamiento son, en general, iguales a -2, -1, 1 y 2. Cuando el número de clases en un intervalo se reduce a 4, por ejemplo, tal como en el ejemplo de las figuras 20A a 25, el número de grupos diferentes de 4 valores de desplazamiento también se reduce. En el ejemplo anterior con 4 valores de desplazamiento diferentes y 4

clases, el número de grupos diferentes es de $4^4=256$. En las realizaciones de las figuras 21, 22, 23 y 25, se utilizan 8 bits para codificar los valores de desplazamiento (4 valores de desplazamiento, codificado cada uno utilizando 2 bits). En este caso, se considera que todos los grupos de 4 desplazamientos tienen la misma probabilidad de ser seleccionados. Sin embargo, algunos de estos grupos son menos probables que otros. Eliminando los grupos menos probables, es posible reducir el número de bits requerido para codificarlos. Como consecuencia, en lugar de codificar 4 valores de desplazamiento diferentes utilizando 2 bits para cada valor de desplazamiento, se propone asignar índices a los diferentes grupos de 4 valores de desplazamiento y codificar el índice, siendo codificado el índice utilizando menos de 8 bits gracias a la eliminación de los grupos menos probables. Las probabilidades de los grupos se podrían determinar aplicando el SAO a un conjunto de secuencias de entrenamiento y calculando estadísticas en los grupos. Se podría predeterminar una tabla que reúna todos los grupos posibles ordenados según su probabilidad de ser seleccionados, y ser conocida por el codificador y el descodificador. En esta tabla, cada grupo de desplazamientos se asociará a un valor de índice. El número de bits asignado al índice de los grupos a codificar podría ser fijo (estandarizado) o fijo para una secuencia, un cuadro, un segmento o una LCU y ser codificado en las cabeceras correspondientes. Un subconjunto de grupos de la tabla correspondientes a los grupos más probables estará determinado por el codificador y el descodificador en función del número de bits asignados a la codificación del índice.

En la **figura 26** se describe una realización que representa la selección del mejor grupo. El proceso se inicia con la etapa -2601- en la que se inicializa la variable j . Esta variable j se aumenta progresivamente para permitir probar todos los grupos posibles de desplazamientos. En la realización propuesta, se consideran grupos de 4 desplazamientos pero se podrían considerar otros números de desplazamientos. En la etapa -2603- se comprueba si se han probado todos los grupos (por ejemplo, NbOffsetGroup podría ser igual a 128). En caso afirmativo, el proceso se detiene y se codifica una palabra de código de menos de 8 bits correspondiente al grupo seleccionado. De lo contrario, el proceso continúa con la inicialización de una variable i que permite probar todas las clases en un intervalo (etapa -2605-). En la etapa -2606-, la variable $\text{SumDiff}(j)$ que representa la suma de la diferencia entre las muestras originales y las muestras codificadas filtradas por SAO correspondientes al grupo de desplazamiento j se inicializa a 0. En este caso, solamente se consideran 4 clases, pero son posibles otros números de clases coherentes con el número de desplazamientos. En la etapa -2607-, si aún no se han probado algunas clases, se inicializa una variable k que permite probar todas las muestras posibles en un intervalo de muestras correspondientes a la clase i . Con las etapas -2611- a -2619- se calcula la suma de los valores absolutos de las diferencias entre las muestras codificadas filtradas con el desplazamiento de la clase i en el grupo de desplazamientos j y las muestras originales en la clase i considerada. En este caso, $\text{orig}(k)$ es la media de los valores de muestra originales correspondientes al valor codificado $\text{enc}(k)$. $\text{Filter}(i, j)$ es el valor de desplazamiento correspondiente a la clase i , en el grupo de desplazamiento j . $\text{SumDiff}(j)$ es la suma de las diferencias calculadas en todas las clases que componen un intervalo (en este caso, 4 clases).

En el bucle que comprende las etapas -2621- a -2626- se comparan todas las sumas calculadas de las diferencias y se selecciona el índice del grupo que tiene la suma mínima. El índice seleccionado corresponde al grupo de desplazamientos que permite, cuando se aplica a las muestras codificadas, minimizar la diferencia entre las muestras filtradas y la muestra original.

Durante el proceso de codificación sintáctica, la codificación de los valores de desplazamiento, tal como se representa por ejemplo, mediante las etapas -2211- a -2213- de la figura 22, es sustituida por la codificación del índice correspondiente al grupo de desplazamientos seleccionado.

La clasificación de los píxeles del componente luma se lleva a cabo independientemente de la clasificación de los píxeles del componente croma U o V y, de este modo, cada clasificación se puede adaptar correspondientemente, de tal modo que cada clase tenga un número similar de píxeles.

Por lo tanto, los procedimientos de las realizaciones de la invención dan a conocer un enfoque de clasificación más flexible que puede ser adaptado para proporcionar una clasificación más óptima independientemente tanto para señales luma como croma, conduciendo de este modo a una mejora en la eficiencia de la codificación.

Aunque la presente invención se ha descrito en lo anterior haciendo referencia a realizaciones particulares, la presente invención no está limitada a las realizaciones particulares y serán evidentes para un experto en la técnica modificaciones que quedan dentro del alcance de la presente invención.

Por ejemplo, aunque las realizaciones anteriores se han descrito con respecto a los píxeles de una imagen y sus valores de píxel correspondientes, se apreciará que dentro del contexto de la invención, se puede considerar un grupo de píxeles conjuntamente con un valor de píxel del grupo correspondiente. Por lo tanto, una muestra puede corresponder a uno o varios píxeles de una imagen.

A los expertos en la materia se les ocurrirán muchas otras modificaciones y variaciones al hacer referencia a las realizaciones ilustrativas anteriores, que se proporcionan solamente a modo de ejemplo y no están destinadas a limitar el alcance de la invención, que está determinado exclusivamente mediante las reivindicaciones adjuntas. En particular, las diferentes características de las diferentes realizaciones se pueden intercambiar, donde proceda.

En las reivindicaciones, la expresión “que comprende” no excluye otros elementos o etapas, y el artículo indefinido “un” o “una” no excluye una pluralidad. El mero hecho de que las diferentes características se enumeren en reivindicaciones dependientes diferentes entre sí no indica que no se pueda utilizar una combinación ventajosa de estas características.

5

REIVINDICACIONES

1. Procedimiento de codificación que utiliza codificación de vídeo de alta eficiencia de una imagen compuesta por una serie de muestras, teniendo cada muestra un valor de muestra, comprendiendo el procedimiento
- 5 codificar (702-708) las muestras;
descodificar (711-714) las muestras codificadas para proporcionar muestras reconstruidas;
realizar un filtrado de bucle de desplazamiento adaptativo de muestras (715;804) sobre las muestras reconstruidas, comprendiendo el filtrado de bucle aplicar desplazamientos de compensación a los valores de muestra de las respectivas muestras reconstruidas, estando asociado cada desplazamiento de compensación con un intervalo de
- 10 valores de muestra;
proporcionar los desplazamientos de compensación mediante:
seleccionar, en base a un criterio de tasa-distorsión, un grupo de clases (814) entre una serie de grupos de clases predeterminados, teniendo cada uno de dichos grupos de clases predeterminados un intervalo y estando compuestos por una serie de clases, definiendo cada una un intervalo de valores de muestra dentro de dicho
- 15 intervalo, clase en la que se pone una muestra si su valor de muestra está dentro del intervalo de la clase involucrada, e incluyendo los grupos de clases predeterminados grupos de clases que tienen respectivas posiciones de centro diferentes dentro de un intervalo completo de los valores de muestra; y mediante
asociar con cada clase del grupo de clases (814) seleccionado un desplazamiento de compensación para su aplicación al valor de la muestra de cada muestra de la clase involucrada,
generar (709) un flujo de bits (710) de muestras codificadas; y
- 20 transmitir, en el flujo de bits (710), datos de clasificación relacionados con el grupo de clases (814) seleccionado;
caracterizado por que dichas posiciones de centro respectivas diferentes incluyen posiciones de centro diferentes **de** un centro de dicho intervalo completo y **por que** los datos de clasificación comprenden datos representativos de dicha posición de centro del grupo de clases (814) seleccionado, dentro del intervalo completo de valores de
- 25 muestra, siendo codificados los datos de clasificación de acuerdo con el intervalo de las clases.
2. Procedimiento, según la reivindicación 1, en el que las muestras del conjunto pueden ser, por lo menos, de un primer tipo de componente o un segundo tipo de componente, siendo luma uno de dichos primer y segundo tipos de componentes y siendo croma el otro de dicho primer y segundo tipos de componentes, y en el que la serie de clases se determina en función del tipo de componente del conjunto de muestras.
- 30
3. Procedimiento, según la reivindicación 1 o 2, en el que el desplazamiento de compensación para cada clase se determina a partir de un promedio de la diferencia entre el valor de la muestra de cada muestra reconstruida de la clase y el valor de la muestra respectiva de la imagen original correspondiente.
- 35
4. Procedimiento, según cualquier reivindicación anterior, que comprende, además transmitir, en el flujo de bits, datos codificados representativos de los desplazamientos de compensación asociados respectivamente con la serie de clases del grupo de clases (814) seleccionado.
- 40
5. Procedimiento de descodificación de una imagen codificada utilizando la codificación de vídeo de alta eficiencia y compuesta por una serie de muestras, teniendo cada muestra un valor de muestra, comprendiendo el procedimiento recibir los valores de muestra codificados;
recibir los datos de clasificación codificados;
recibir los desplazamientos de compensación codificados;
- 45 descodificar los datos de clasificación e identificar, en base a los datos de clasificación descodificados, un grupo de clases (814) seleccionado por un codificador entre una serie de grupos de clases predeterminados, teniendo cada uno de dichos grupos de clases predeterminados un intervalo y estando compuesto por una serie de clases, definiendo cada una un intervalo de valores de muestra dentro de dicho intervalo, clase en la que se pone una muestra si su valor de muestra está dentro del intervalo de la clase involucrada, e incluyendo los grupos de clases predeterminados grupos de clases que tienen respectivas posiciones de centro diferentes dentro de un intervalo completo de los valores de muestra;
descodificar (902-906) las muestras codificadas para proporcionar los valores de muestra reconstruida y descodificar los desplazamientos de compensación codificados;
- 50 asociar los desplazamientos de compensación descodificados respectivamente con las clases del grupo de clases (814) seleccionado; y
llevar a cabo el filtrado de bucle de desplazamiento adaptativo de muestras (907; 804) sobre los valores de muestras reconstruidas, comprendiendo el filtrado de bucle aplicar el desplazamiento de compensación descodificado asociado a cada clase del grupo de clases seleccionado, a los valores de muestra reconstruidas dentro del intervalo de la clase involucrada,
- 55 **caracterizado por que** dichas posiciones de centro respectivas diferentes incluyen posiciones de centro diferentes de un centro de dicho intervalo completo y **por que** los datos de clasificación comprenden datos representativos de dicha posición de centro del grupo de clases (814) seleccionado, dentro del intervalo completo de valores de muestra, siendo codificados los datos de clasificación de acuerdo con el intervalo de las clases.
- 60
6. Producto de programa informático para un aparato programable, comprendiendo el producto de programa informático una secuencia de instrucciones para implementar un procedimiento según cualquiera de las
- 65

reivindicaciones 1 a 5 cuando es cargado y ejecutado por el aparato programable.

7. Medio de almacenamiento legible por ordenador que almacena instrucciones de un programa informático para implementar un procedimiento según cualquiera de las reivindicaciones 1 a 5.

8. Señal (710) que transporta un conjunto de datos de información para una imagen codificada utilizando la codificación de vídeo de alta eficiencia y representada por un flujo de bits de vídeo, comprendiendo la imagen un conjunto de muestras reconstruibles, teniendo cada muestra reconstruible un valor de muestra, comprendiendo el conjunto de datos de información: datos de clasificación relacionados con un grupo de clases (814) seleccionado por un codificador entre una serie de grupos de clases predeterminados, teniendo cada uno de dichos grupos de clases predeterminados un intervalo de grupo y estando compuesto por una serie de clases, definiendo cada una un intervalo de valores de muestra dentro de dicho intervalo de grupo, clase en la que se pone una muestra si su valor de muestra está dentro del intervalo de la clase involucrada, e incluyendo los grupos de clases predeterminados grupos de clases que tienen respectivas posiciones de centro diferentes dentro de un intervalo completo de los valores de muestra, y estando asociada cada clase de la serie de clases del grupo de clases (814) seleccionado con un desplazamiento de compensación para su aplicación, en el filtrado de bucle adaptativo de muestras (804), a valores de muestras de las muestras reconstruibles dentro del intervalo de la clase involucrada, **caracterizada por que** dichas posiciones de centro respectivas diferentes incluyen posiciones de centro diferentes de un centro de dicho intervalo completo y **por que** los datos de clasificación comprenden datos representativos de dicha posición de centro del grupo de clases (814) seleccionado, dentro del intervalo completo de valores de muestra, siendo codificados los datos de clasificación según el intervalo de las clases.

9. Dispositivo de codificación para codificar utilizando la codificación de vídeo de alta eficiencia una imagen compuesta por una serie de muestras, teniendo cada muestra un valor de muestra, comprendiendo el dispositivo un codificador (702-708) para codificar las muestras; un descodificador (711-714) para descodificar las muestras codificadas para proporcionar las muestras reconstruidas; un filtro de bucle de desplazamiento adaptativo de muestras (715; 804) para filtrar las muestras reconstruidas, comprendiendo el filtro de bucle de desplazamiento adaptativo de muestras medios (804) de aplicación de desplazamientos para aplicar desplazamientos de compensación a los valores de muestra de muestras reconstruidas respectivas, estando asociado cada desplazamiento de compensación con un intervalo de valores de muestra; un proveedor de desplazamientos de compensación, para proporcionar los desplazamientos de compensación, que comprende:

medios para seleccionar, en base a un criterio de tasa-distorsión, un grupo de clases (814) entre una serie de grupos de clases predeterminados, teniendo cada uno de dichos grupos de clases predeterminados un intervalo y estando compuesto por una serie de clases, definiendo cada una un intervalo de valores de muestra, clase en la que se pone una muestra si su valor de muestra está dentro del intervalo de la clase involucrada, e incluyendo los grupos de clases predeterminados grupos de clases que tienen posiciones de centro respectivas diferentes dentro de un intervalo completo de los valores de muestra; y que comprenden, además, medios para asociar con cada clase del grupo de clases (814) seleccionado un desplazamiento de compensación para su aplicación al valor de la muestra de cada muestra de la clase involucrada; un generador de flujo de bits para generar (709) un flujo de bits (710) de muestras codificadas; y medios para transmitir, en el flujo de bits (710), datos de clasificación en relación con el grupo de clases (814) seleccionado; **caracterizado por que** dichas posiciones de centro respectivas diferentes incluyen posiciones de centro diferentes de un centro de dicho intervalo completo y **por que** los datos de clasificación comprenden datos representativos de dicha posición de centro del grupo de clases (814) seleccionado, dentro del intervalo completo de valores de muestra, siendo codificados los datos de clasificación según el intervalo de las clases.

10. Dispositivo para descodificar una imagen codificada utilizando codificación de vídeo de alta eficiencia y compuesta por una serie de muestras, teniendo cada muestra un valor de muestra, comprendiendo el dispositivo medios para recibir valores de muestra codificados; medios para recibir datos de clasificación codificados; medios para recibir desplazamientos de compensación codificados; medios para descodificar los datos de clasificación y para identificar, en base a los datos de clasificación descodificados, un grupo de clases (814) seleccionado por un codificador entre una serie de grupos de clases predeterminados, teniendo cada uno de dichos grupos de clases predeterminado un intervalo y estando compuesto por una serie de clases, definiendo cada una un intervalo de valores de muestra, clase en la que se pone una muestra si su valor de muestra está dentro del intervalo de la clase involucrada, e incluyendo los grupos de clases predeterminados grupos de clases que tienen posiciones de centro respectivas diferentes dentro de un intervalo completo de los valores de muestra; medios para descodificar las muestras codificadas (902-906) con el fin de proporcionar valores de muestra reconstruidas y para descodificar los desplazamientos de compensación codificados; medios para asociar los desplazamientos de compensación descodificados respectivamente con las clases del

- grupo de clases (814) seleccionado; y
medios para realizar el filtrado de bucle de desplazamiento adaptativo de muestras (907; 804) sobre los valores de muestra reconstruidas, comprendiendo el filtrado de bucle aplicar el desplazamiento de compensación descodificado asociado con cada clase del grupo de clases seleccionado a los valores de muestra reconstruida dentro del intervalo de la clase involucrada,
- 5 **caracterizado por que** dichas posiciones de centro respectivas diferentes incluyen posiciones de centro diferentes de un centro de dicho intervalo completo y **por que** los datos de clasificación comprenden datos representativos de dicha posición de centro del grupo de clases (814) seleccionado, dentro del intervalo completo de los valores de muestra, siendo codificados los datos de clasificación según el intervalo de las clases.

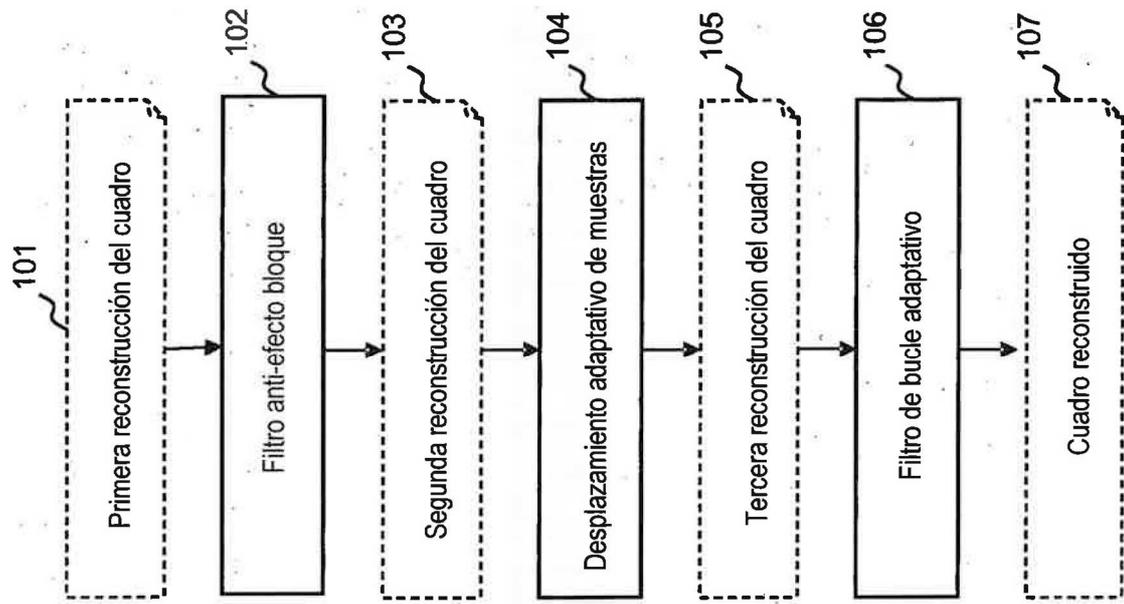
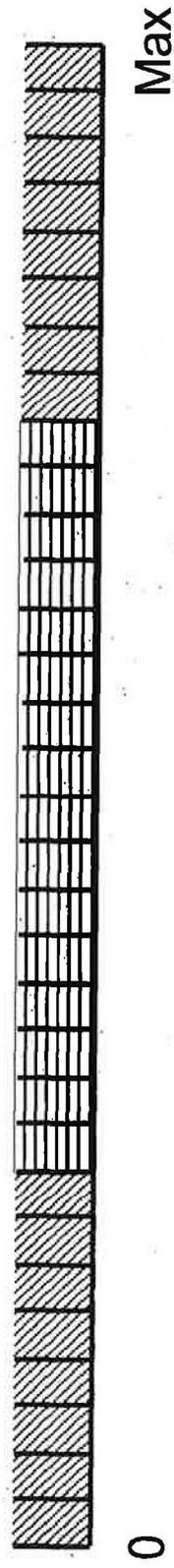


Figura 1

10

Figura 2



20

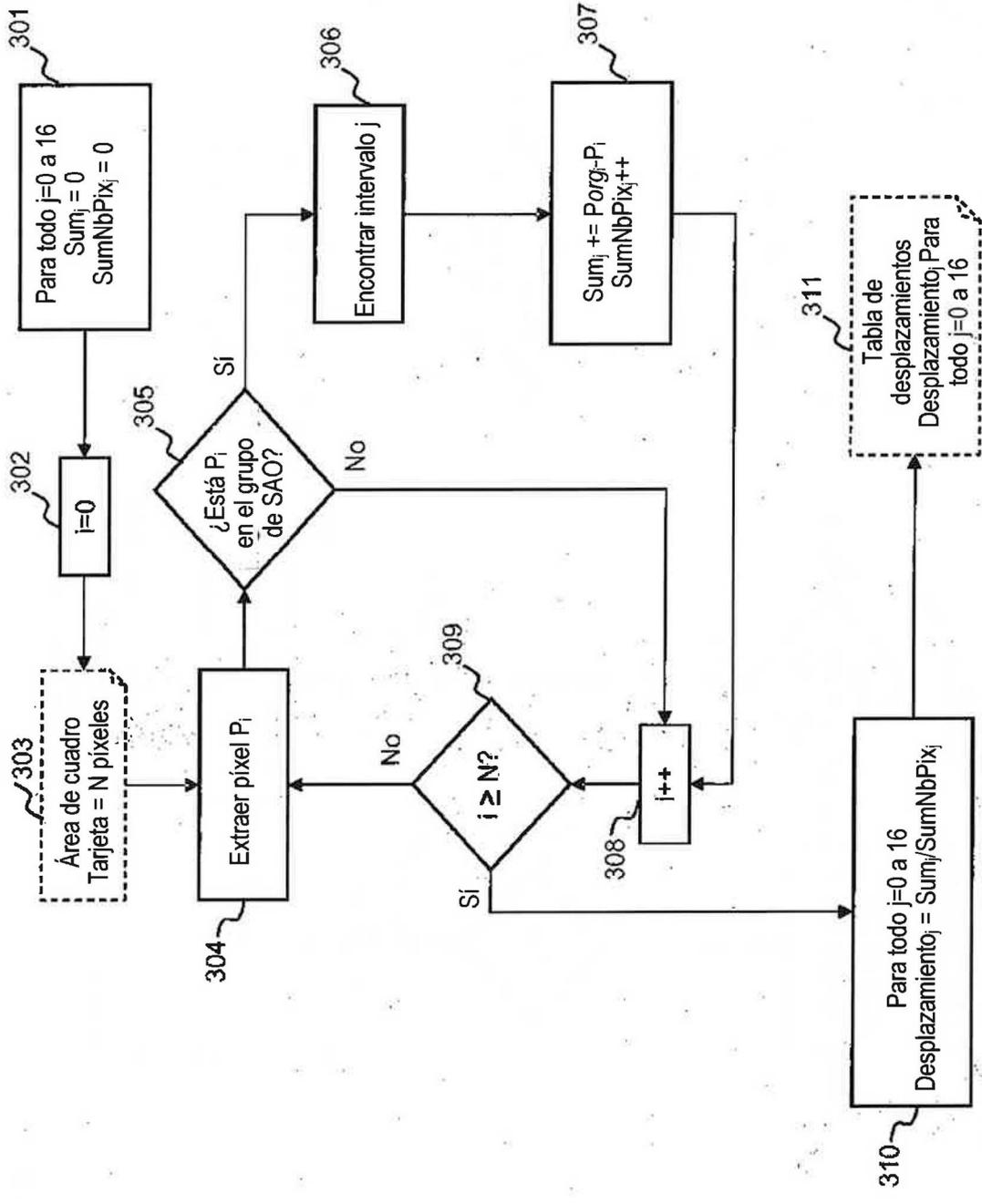


Figura 3

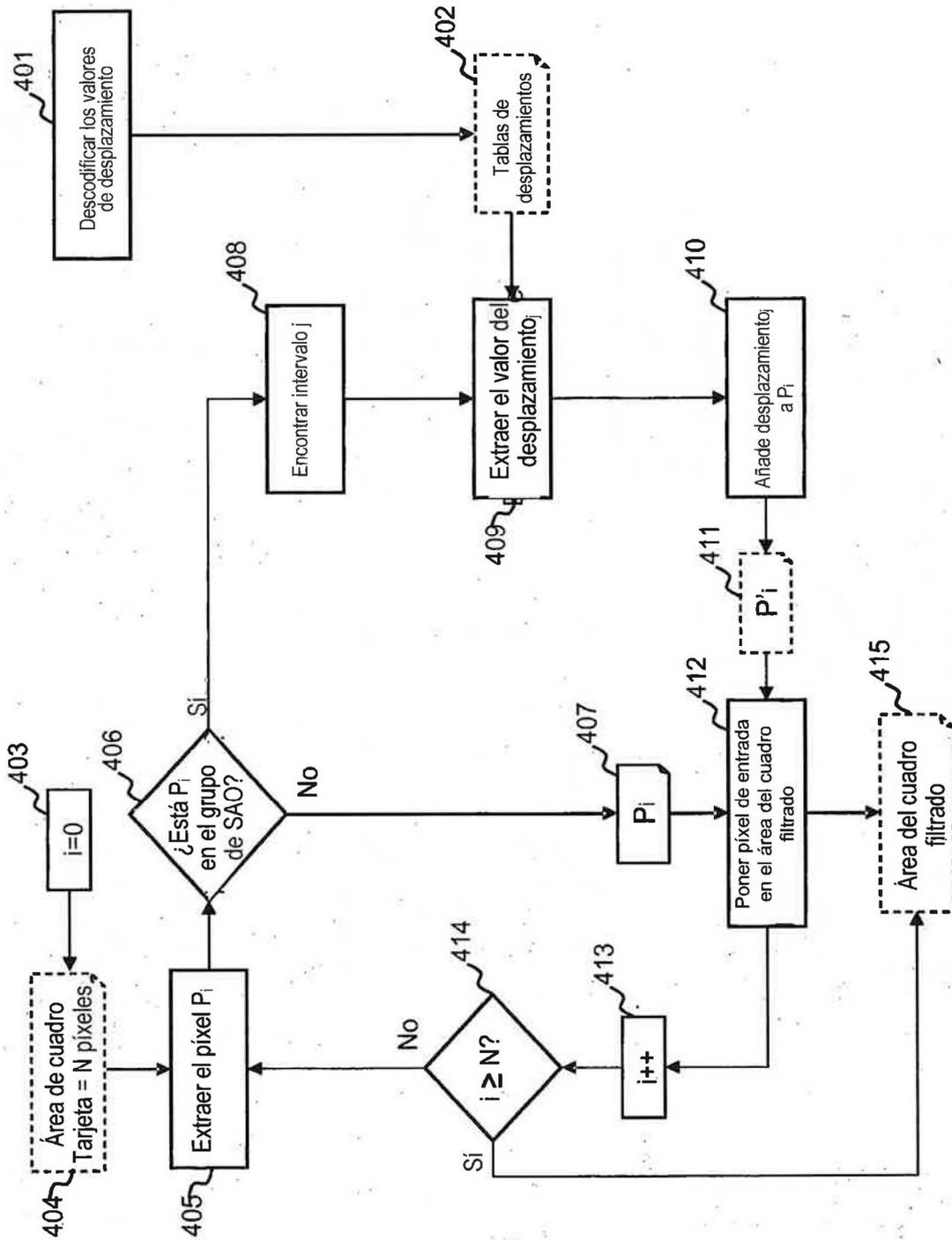


Figura 4

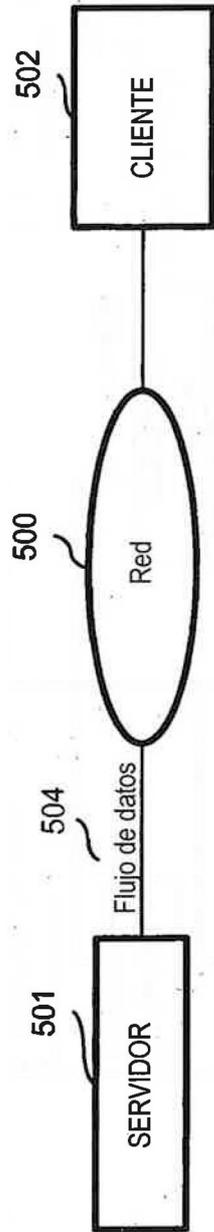


Fig. 5

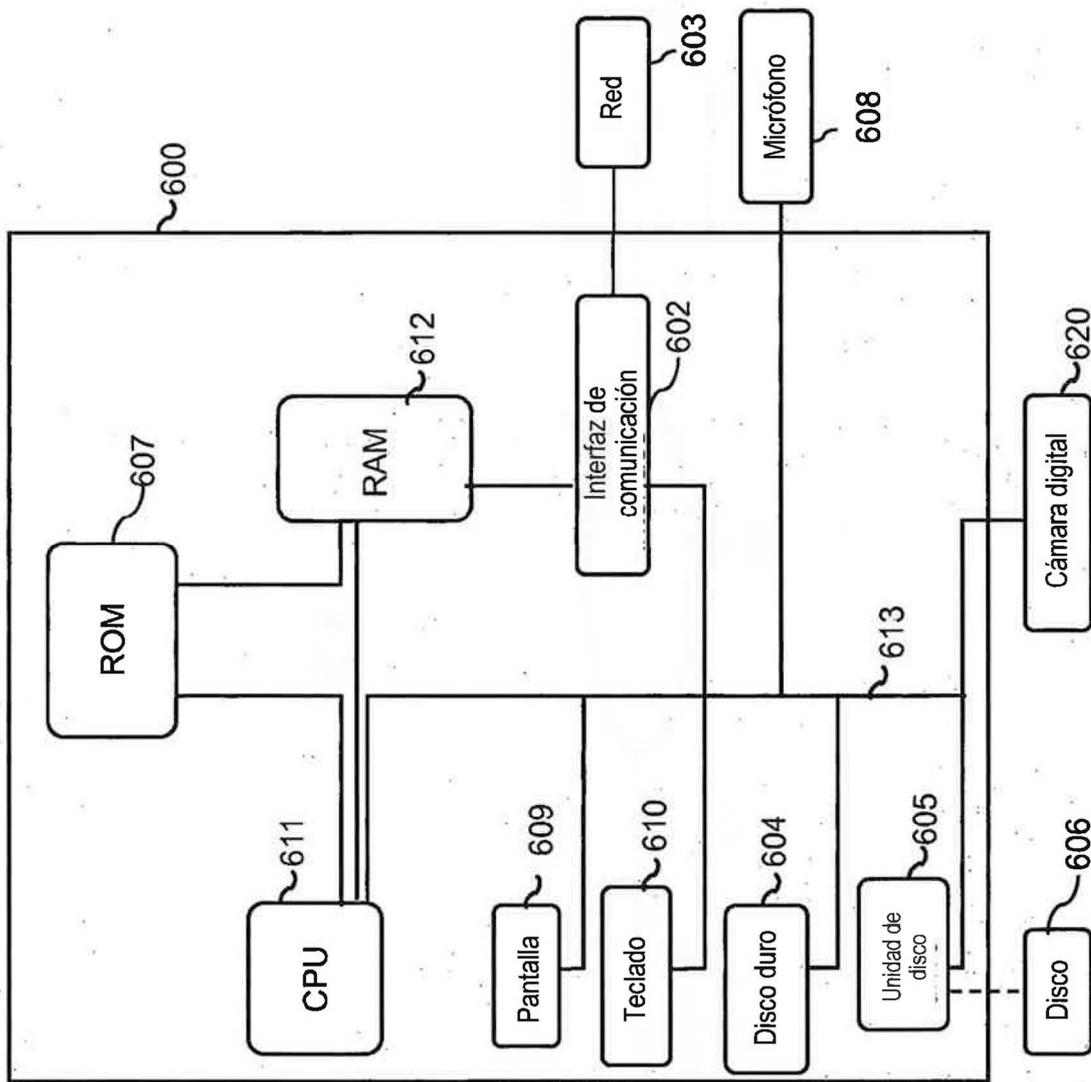


Figura 6

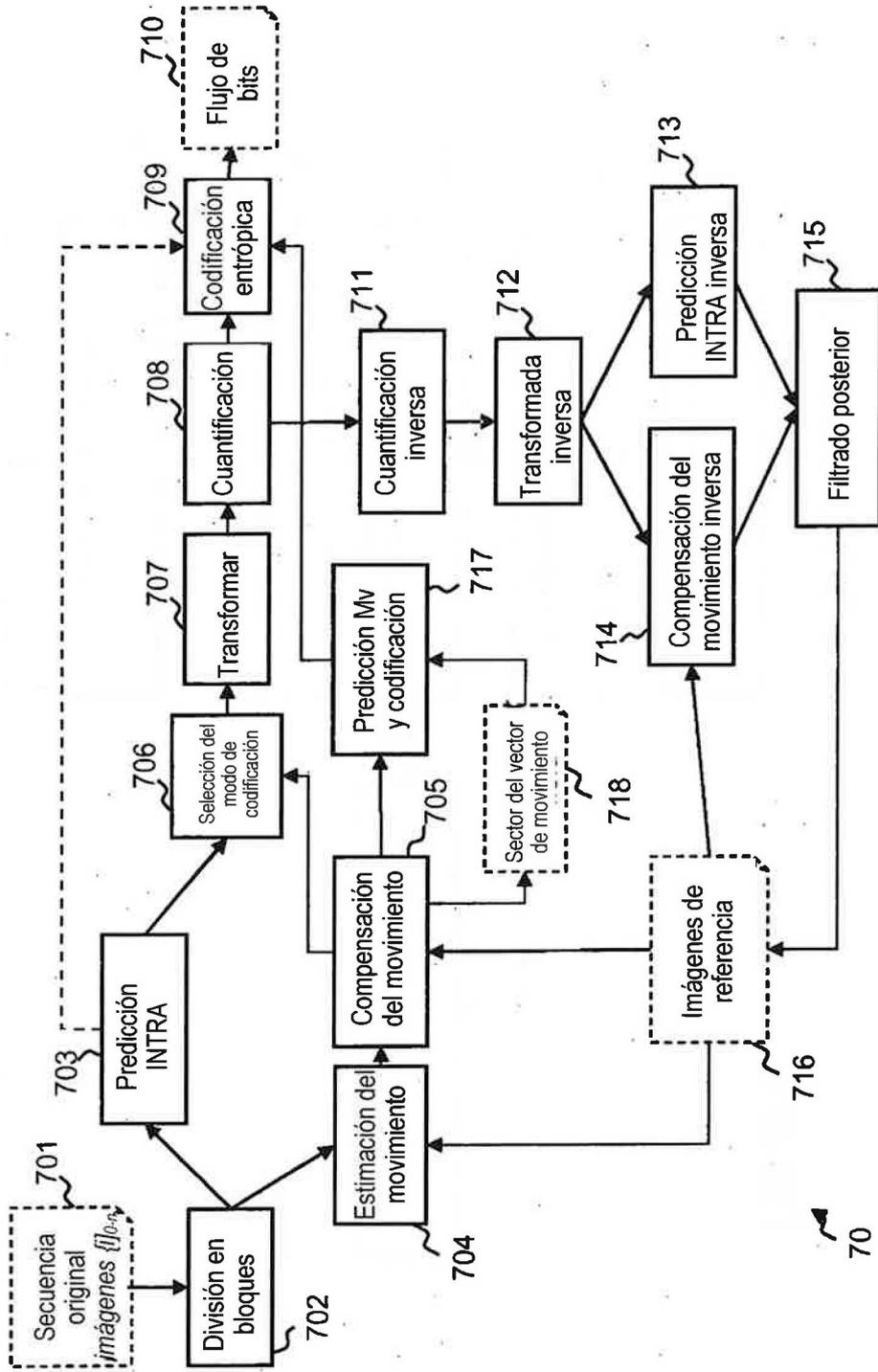


Figura 7

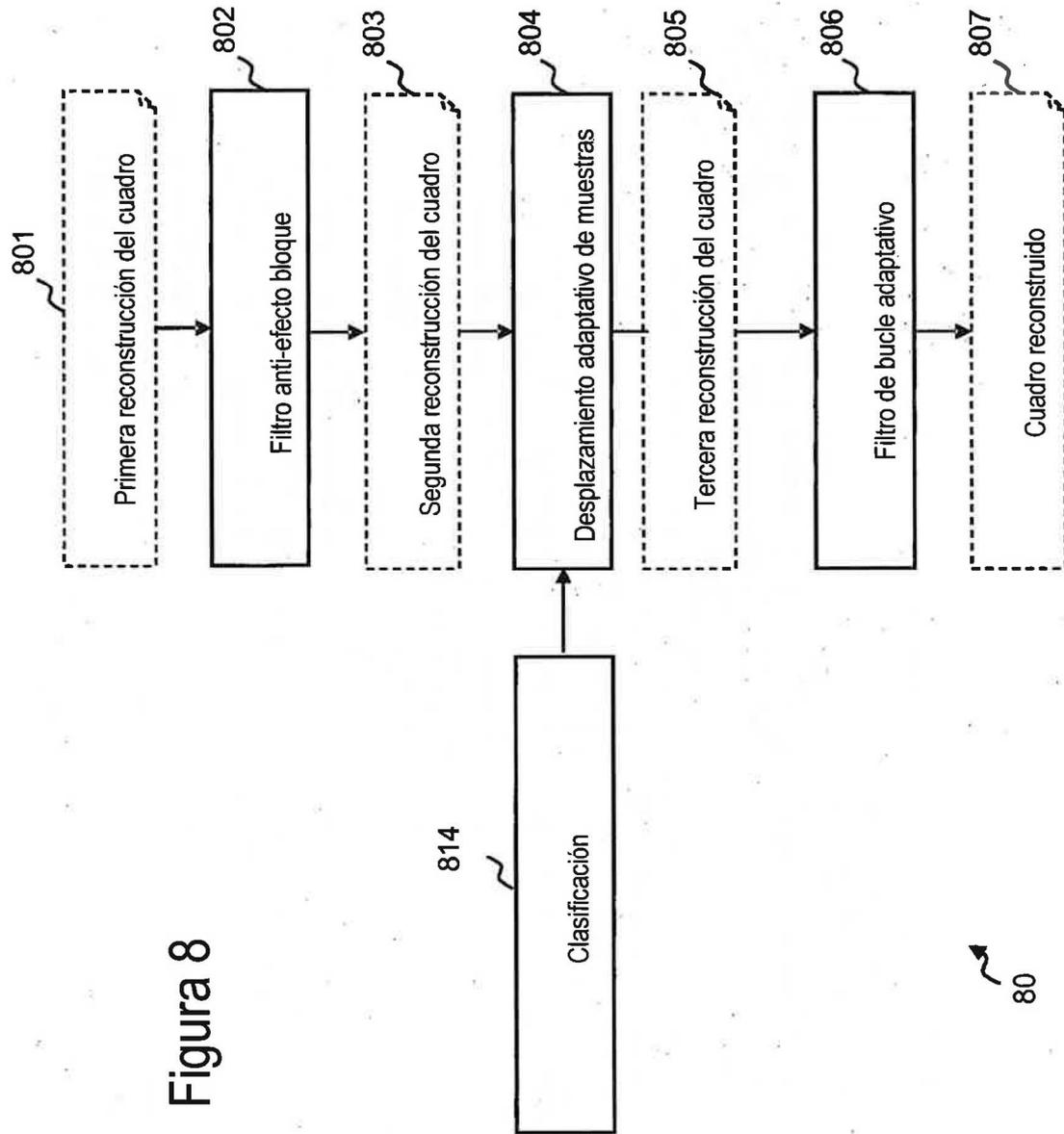


Figura 8

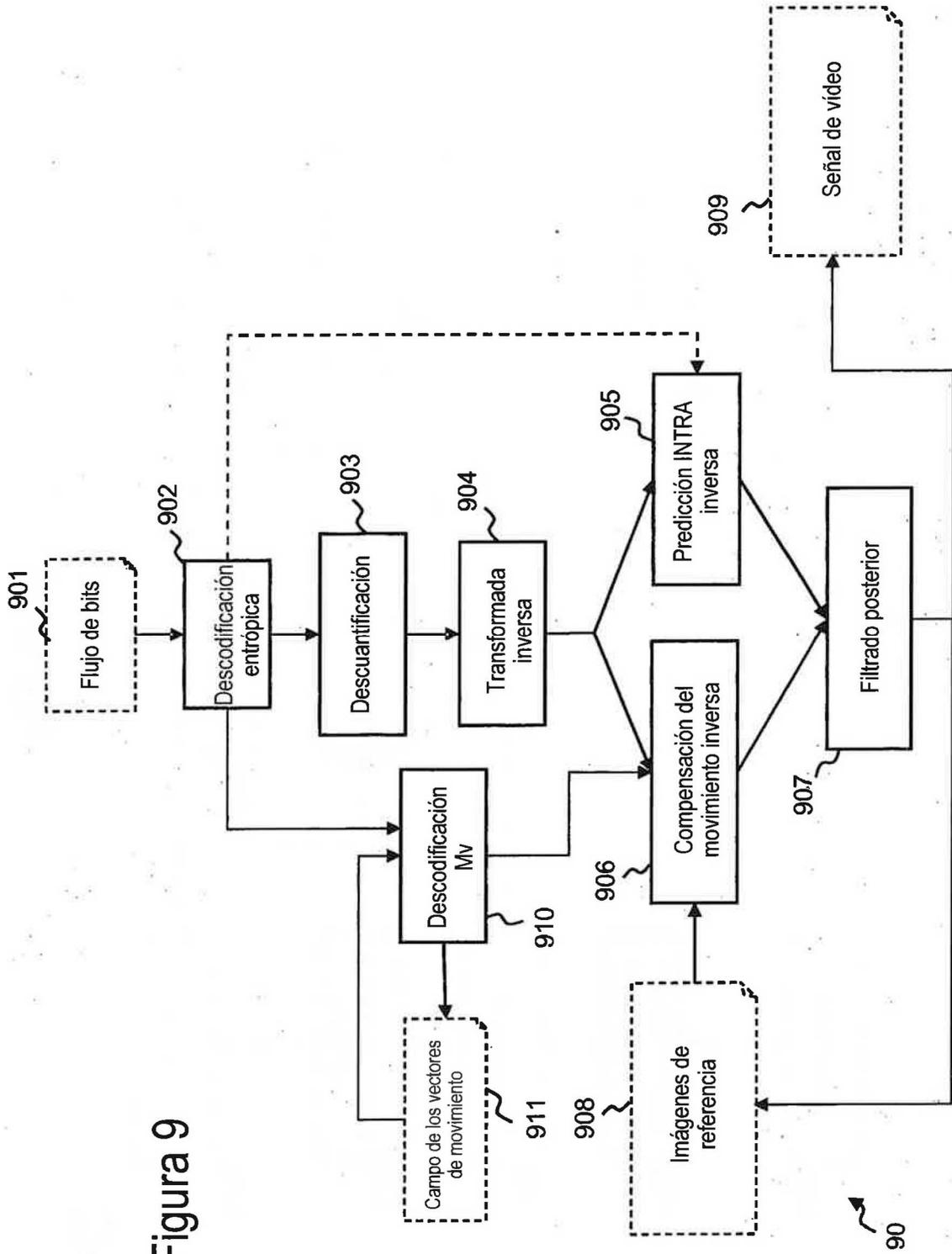


Figura 9

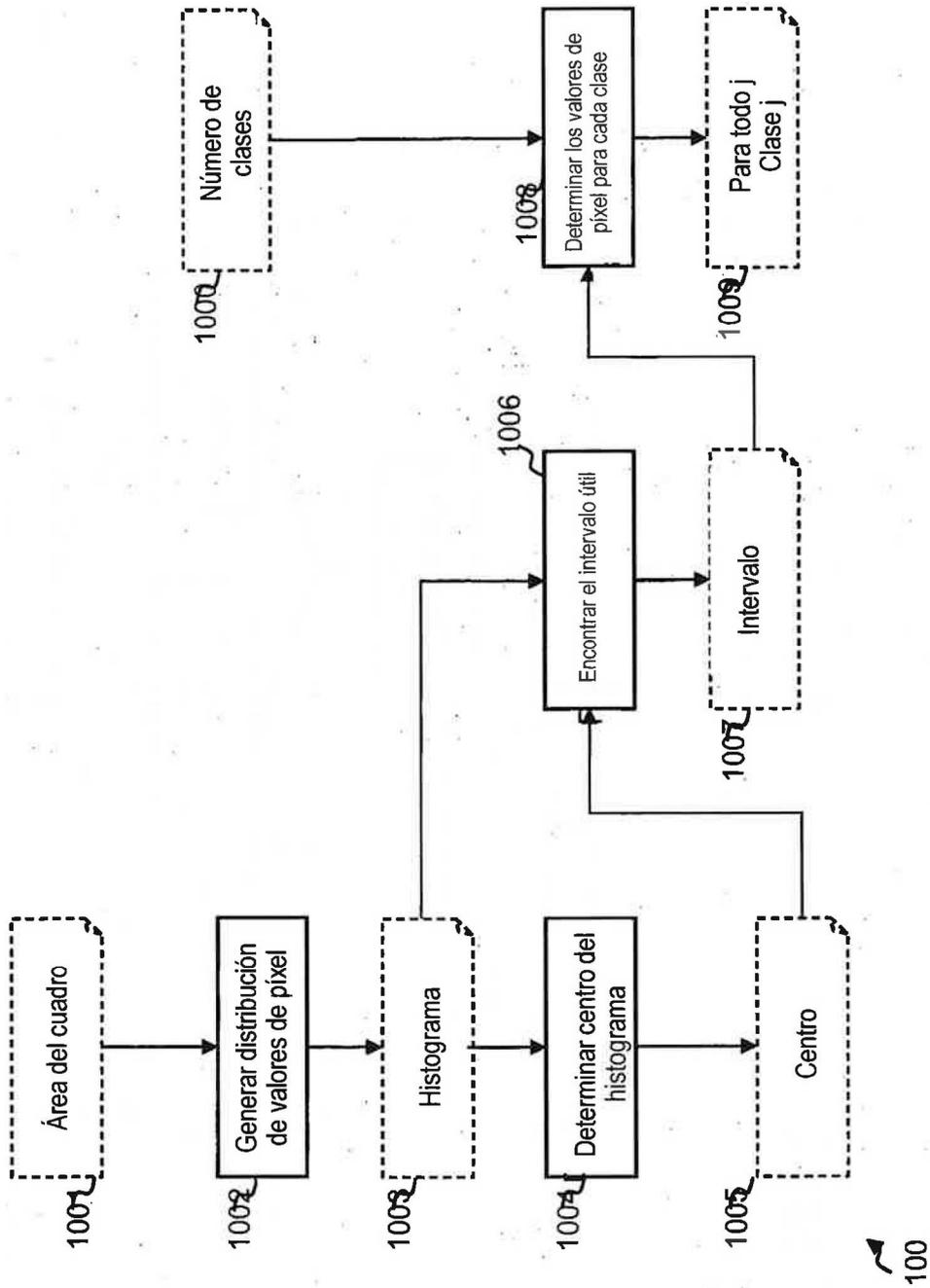


Figura 10

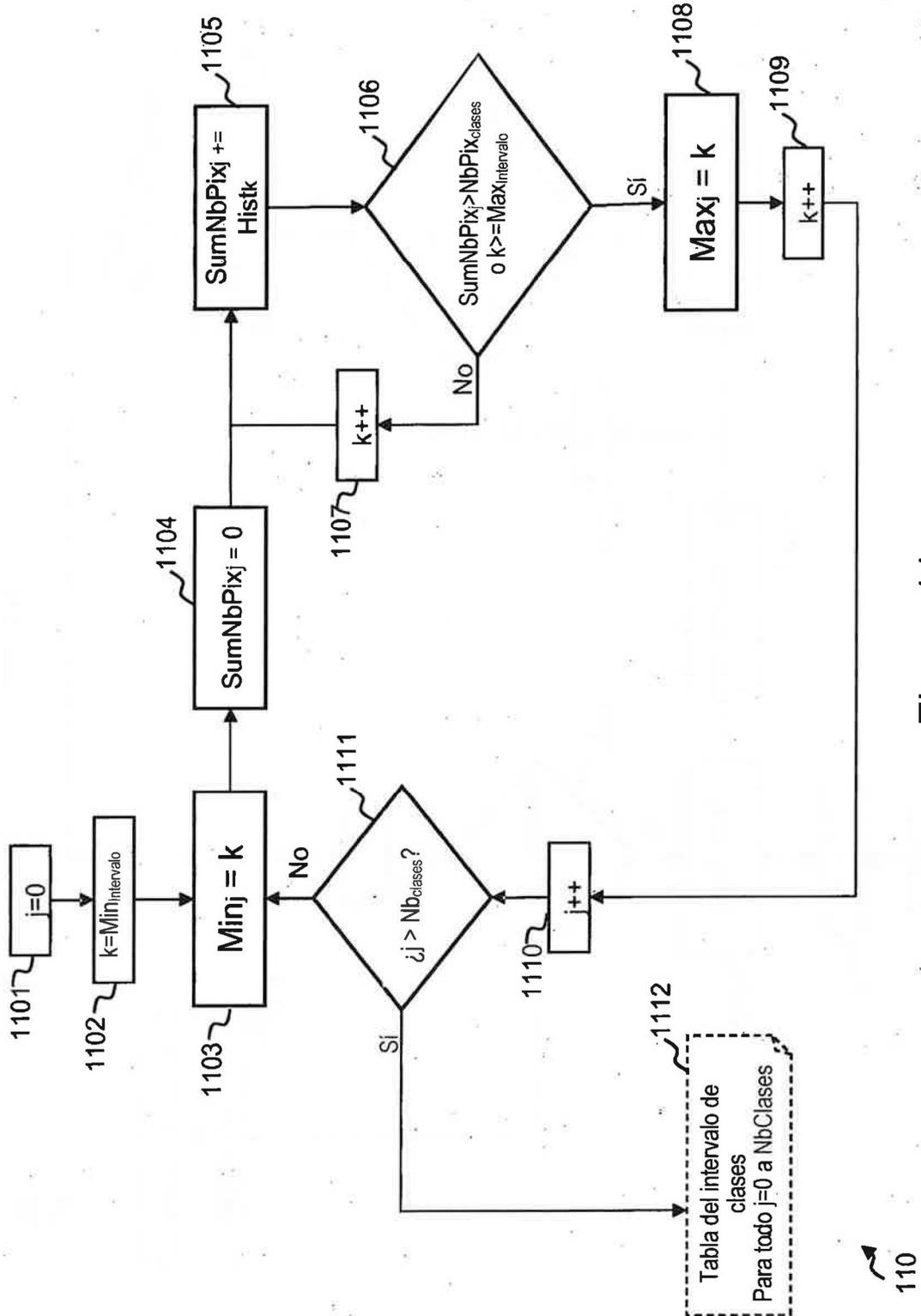


Figura 11

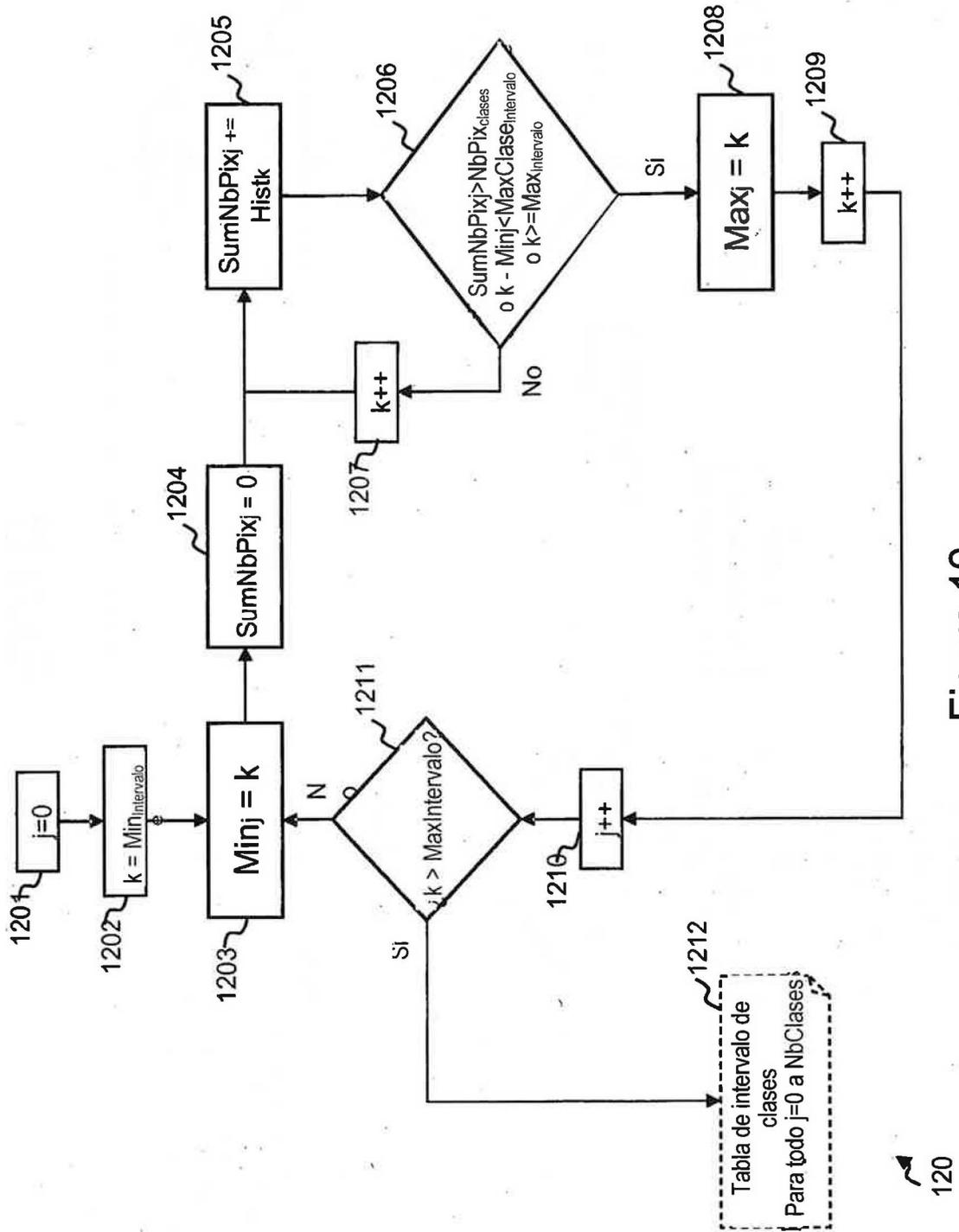


Figura 12

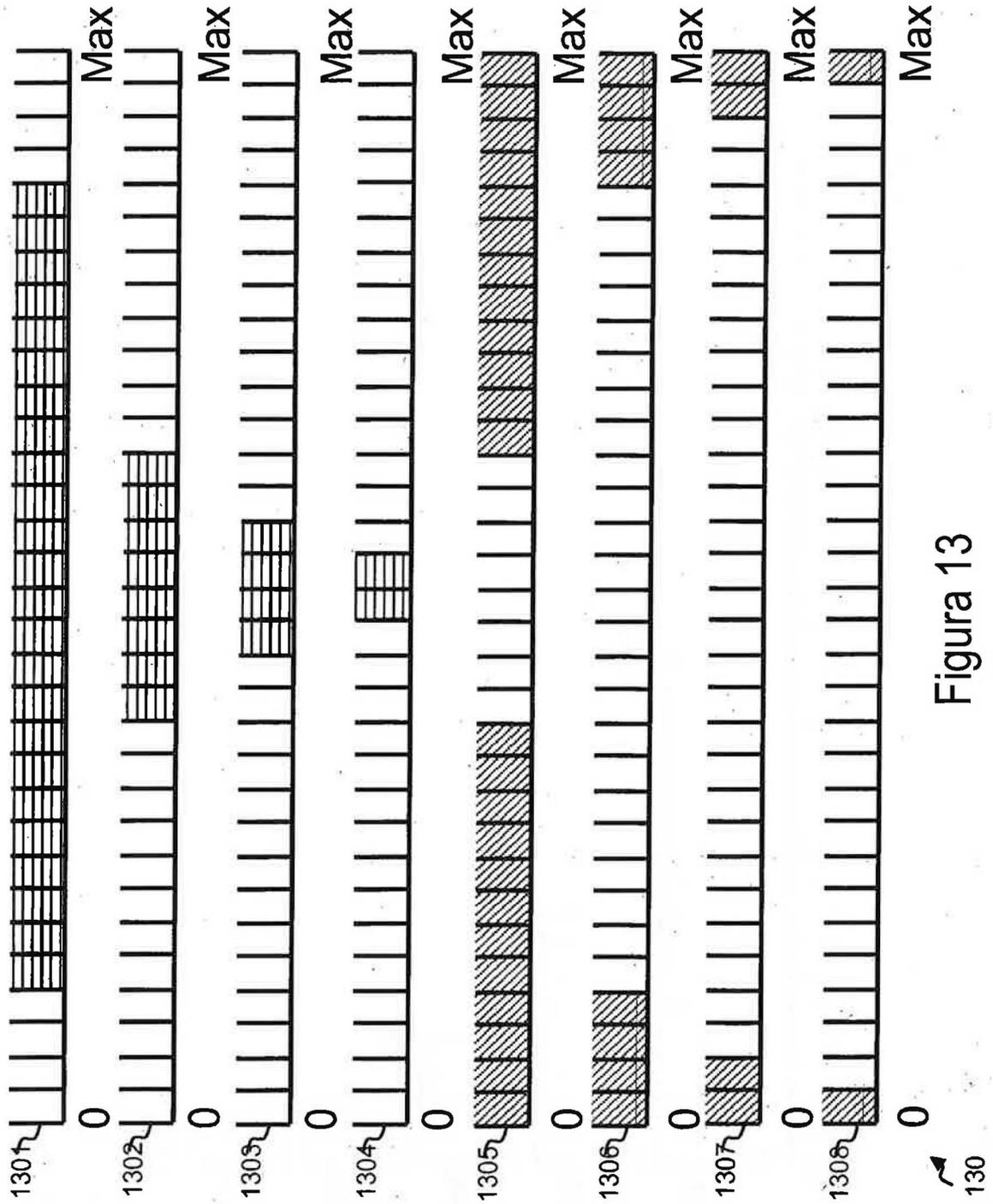
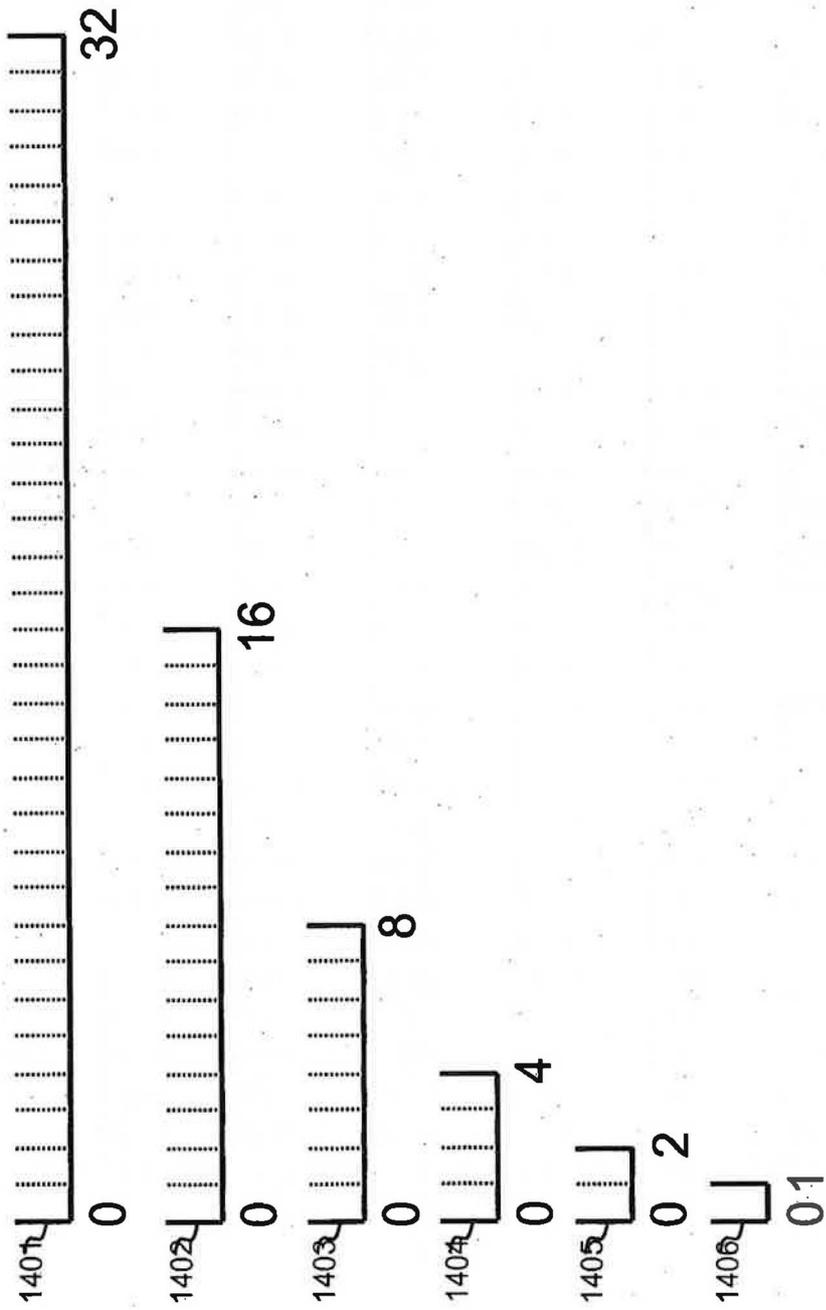
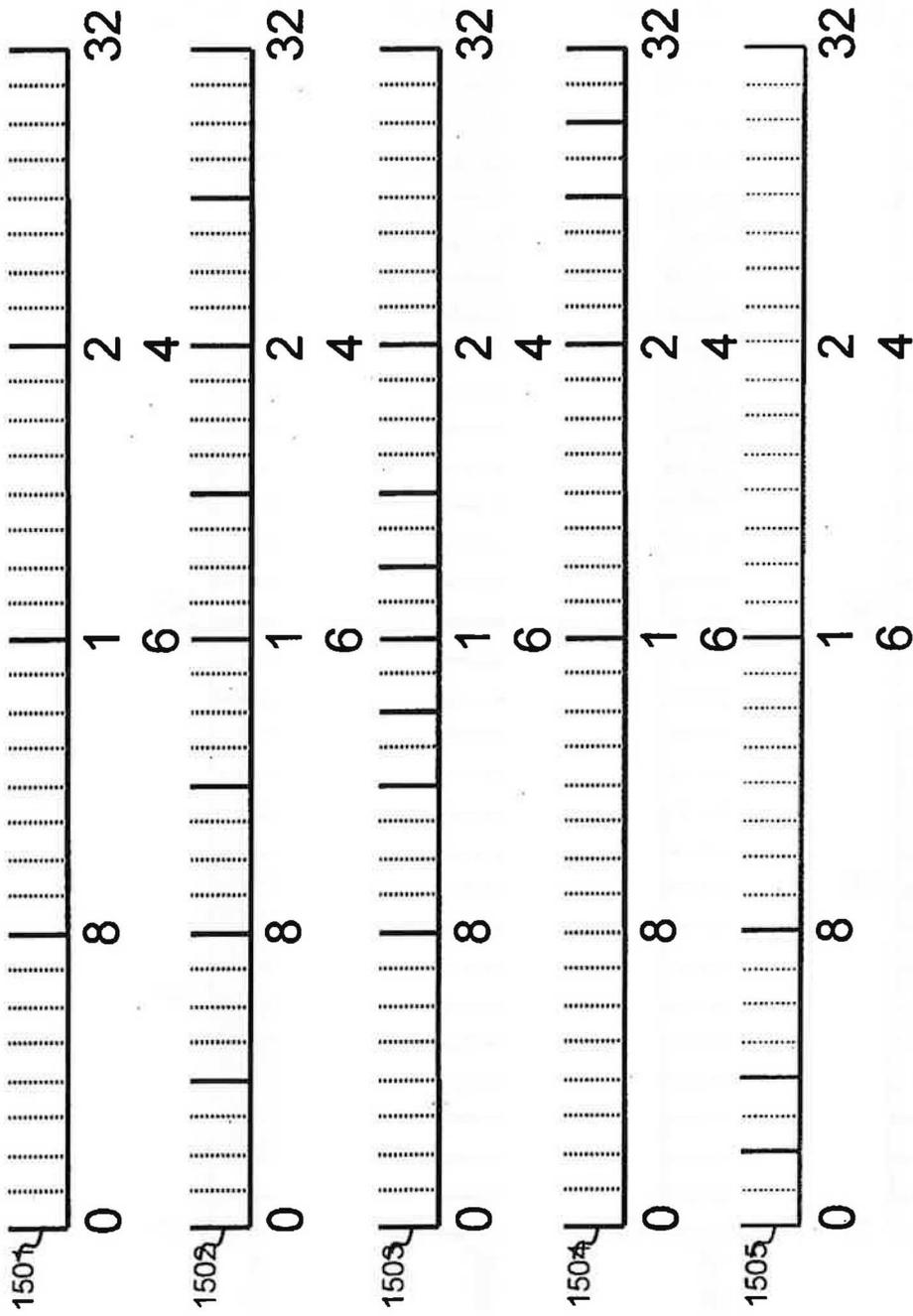


Figura 13



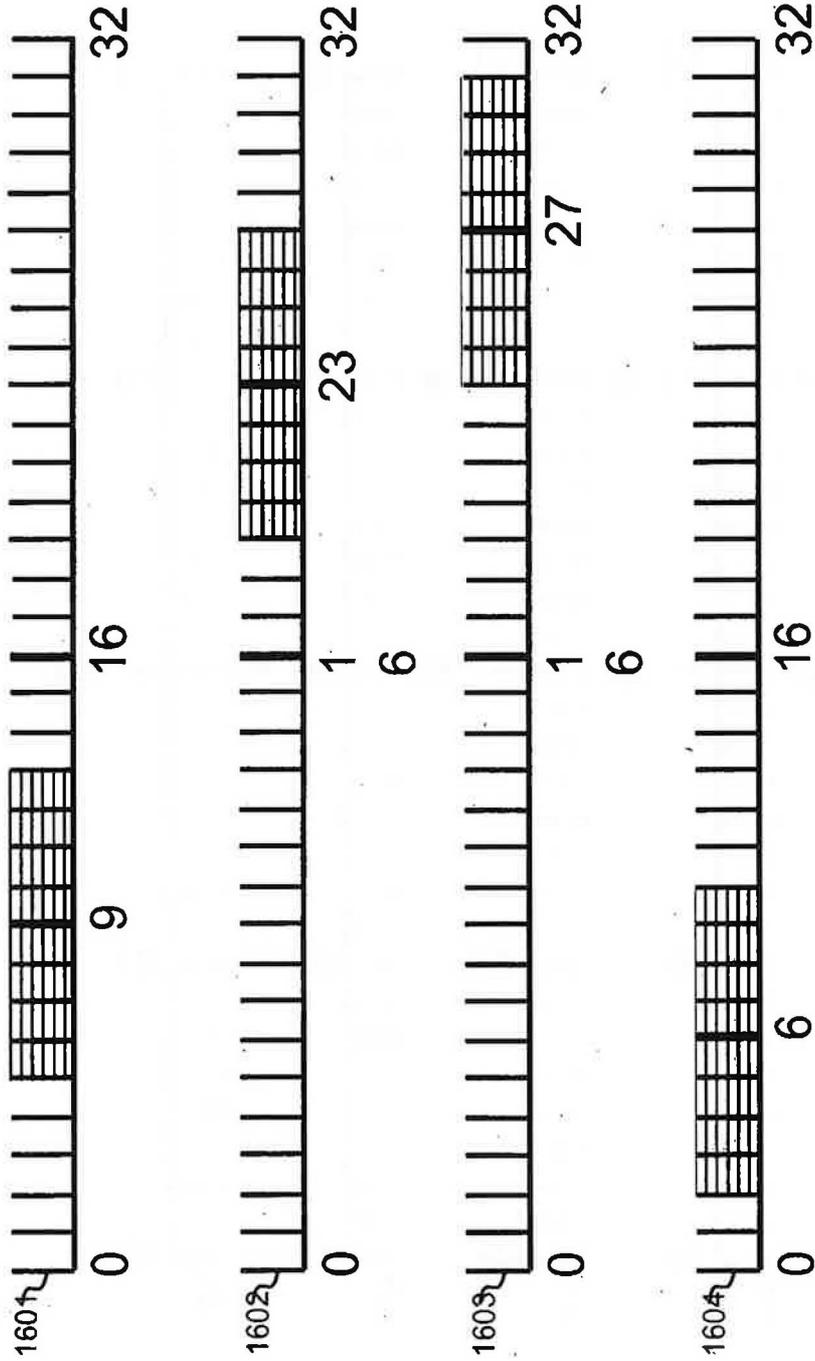
140

Figura 14



150

Figura 15



160

Figura 16

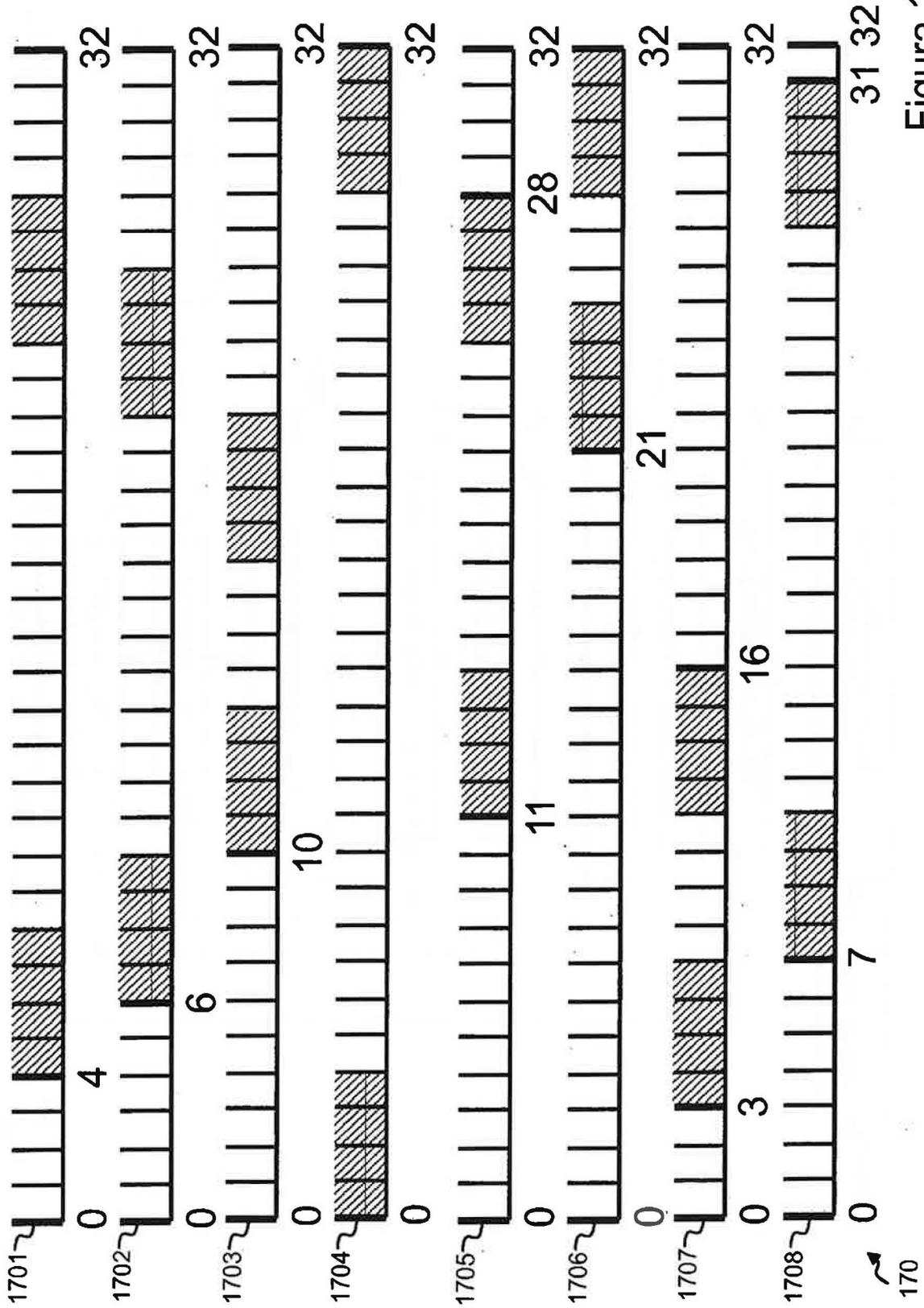


Figura 17

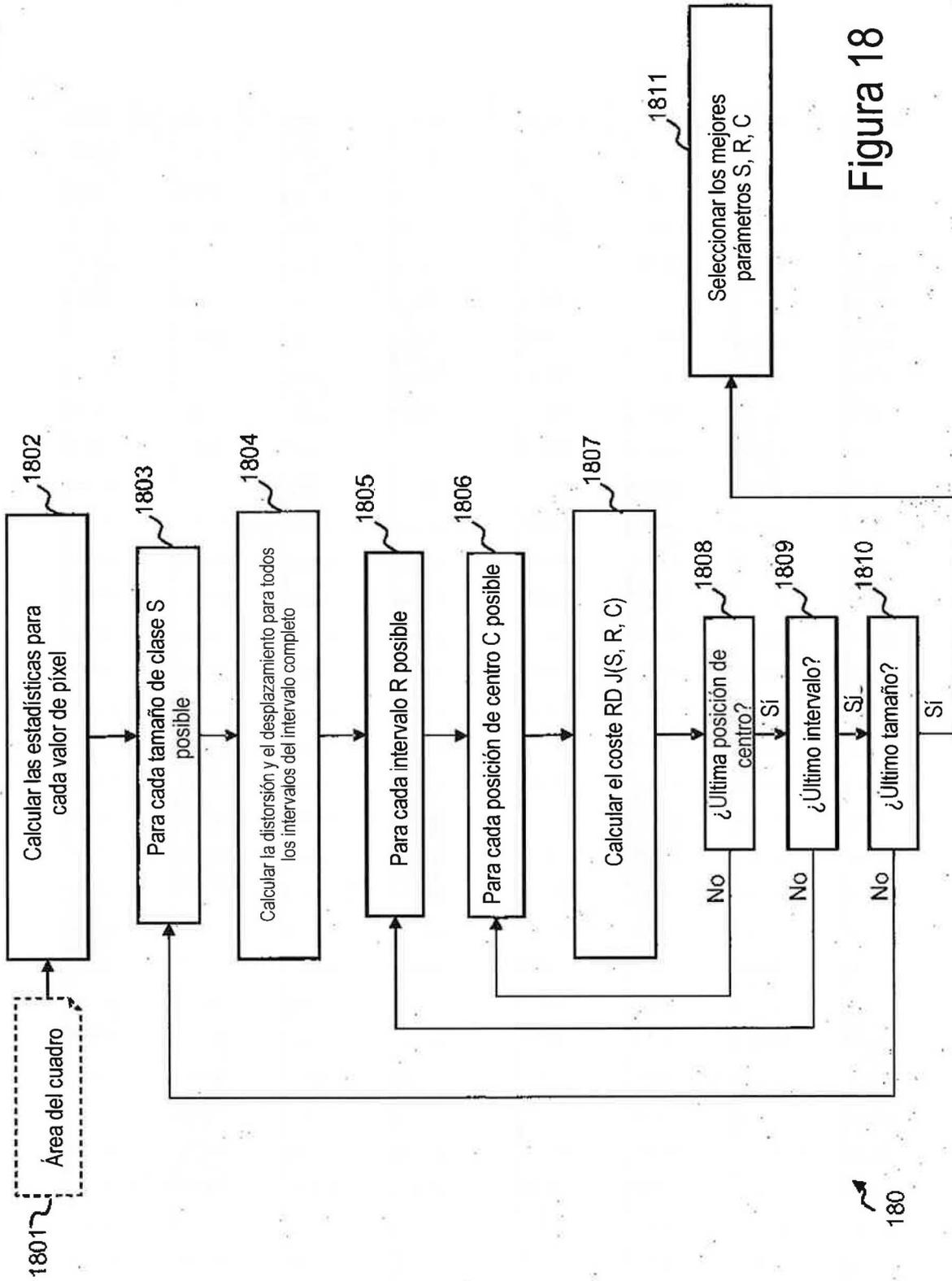


Figura 18

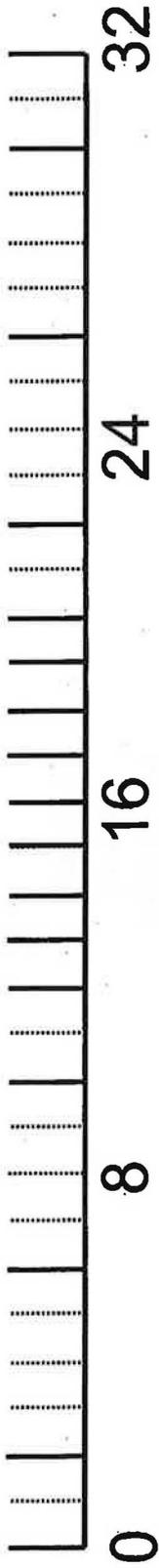


Figura 19a

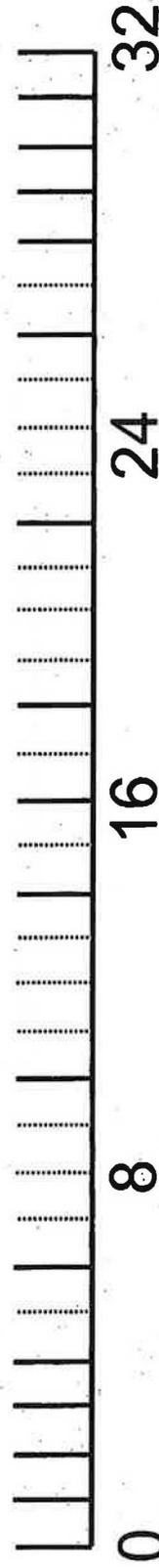


Figura 19b

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

Figura 20A
(técnica anterior)

sao_offset_vlc(rx, ry, cldx) {	Descriptor
sao_type_idx[cldx][rx][ry]	u(3)
if(sao_type_idx[cldx][rx][ry] != 0) {	
if(sao_type_idx[cldx][rx][ry] = 5) {	
sao_band_position[cldx][rx][ry]	u(5)
}	
for(i = 0; i < 4; i++)	
sao_offset[cldx][rx][ry][i]	u(2)
}	
}	

Figura 20B

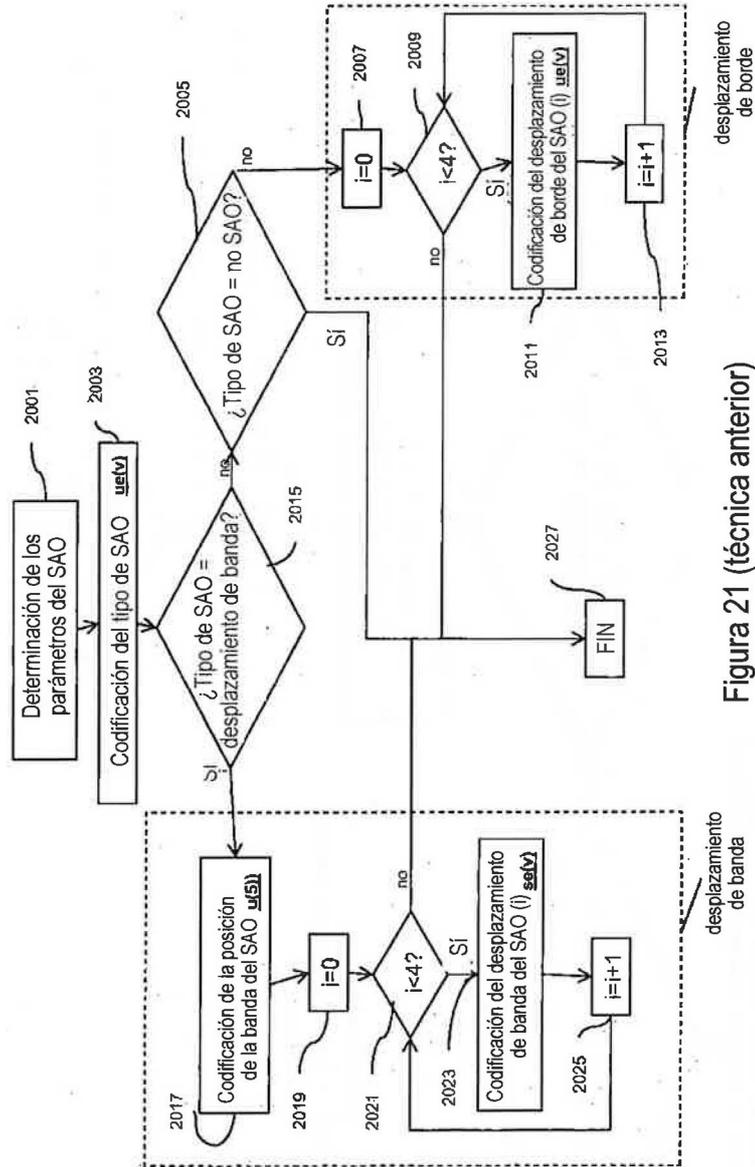


Figura 21 (técnica anterior)

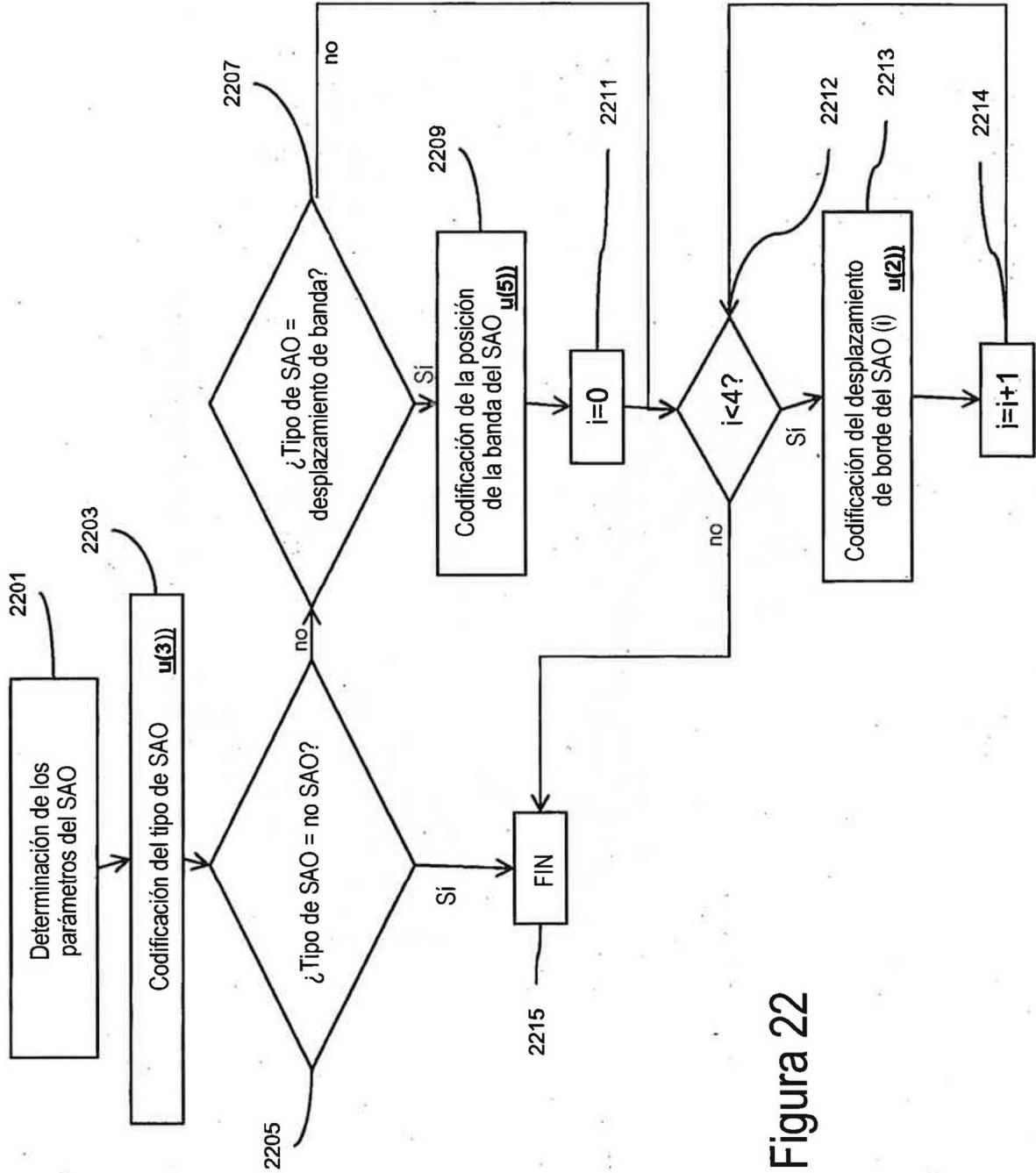


Figura 22

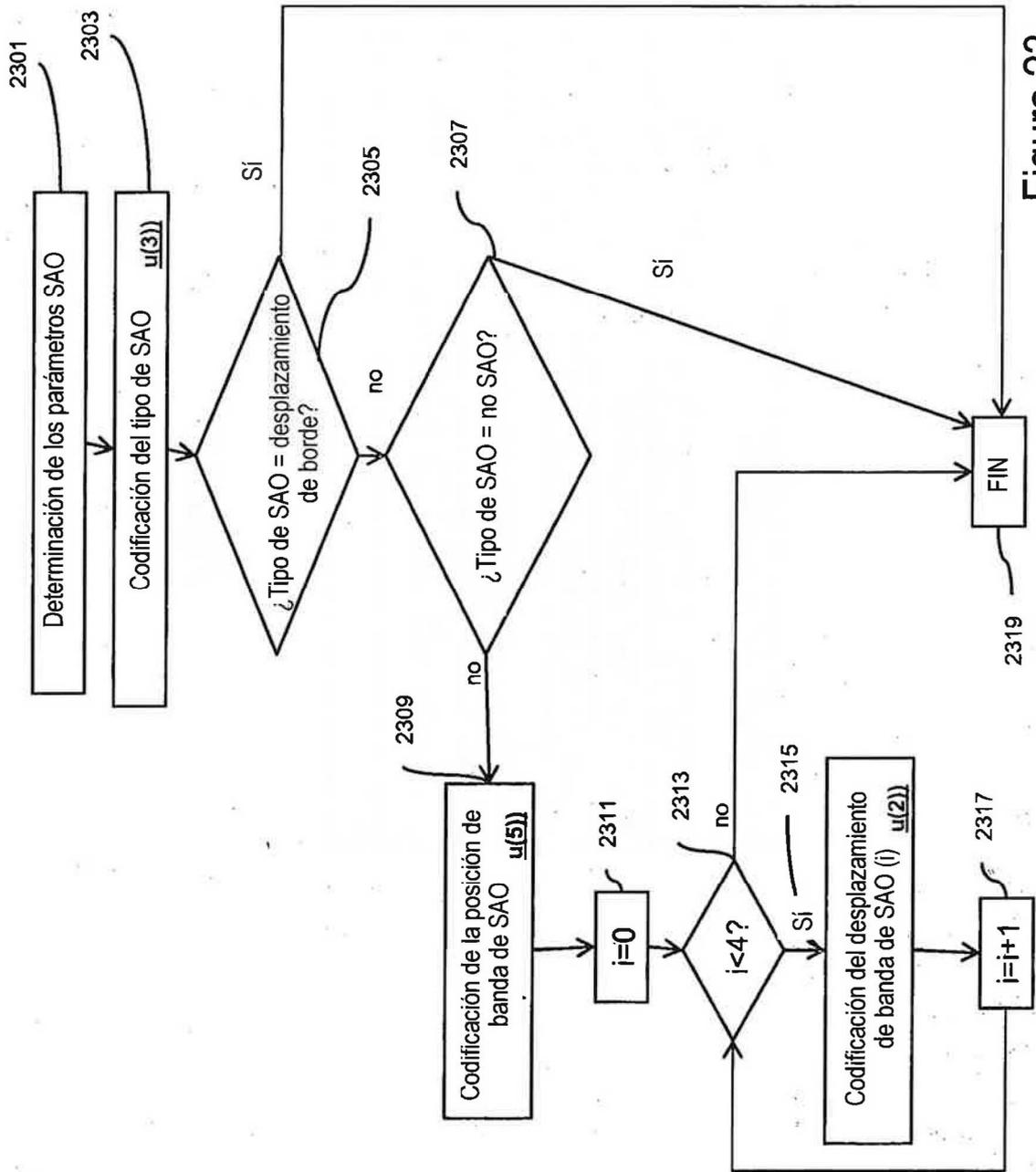


Figura 23

sao_offset_vlc(rx, ry, cldx) {	Descrip
sao_lcu_flag[cldx][rx][ry]	tor
if(sao_lcu_flag[cldx][rx][ry]){	u(1)
sao_band_position_and_EO[cldx][rx][ry]	u(5)
}	
for(i = 0; i < 4; i++)	
sao_offset[cldx][rx][ry][i]	u(2)
}	
}	

Figura 24

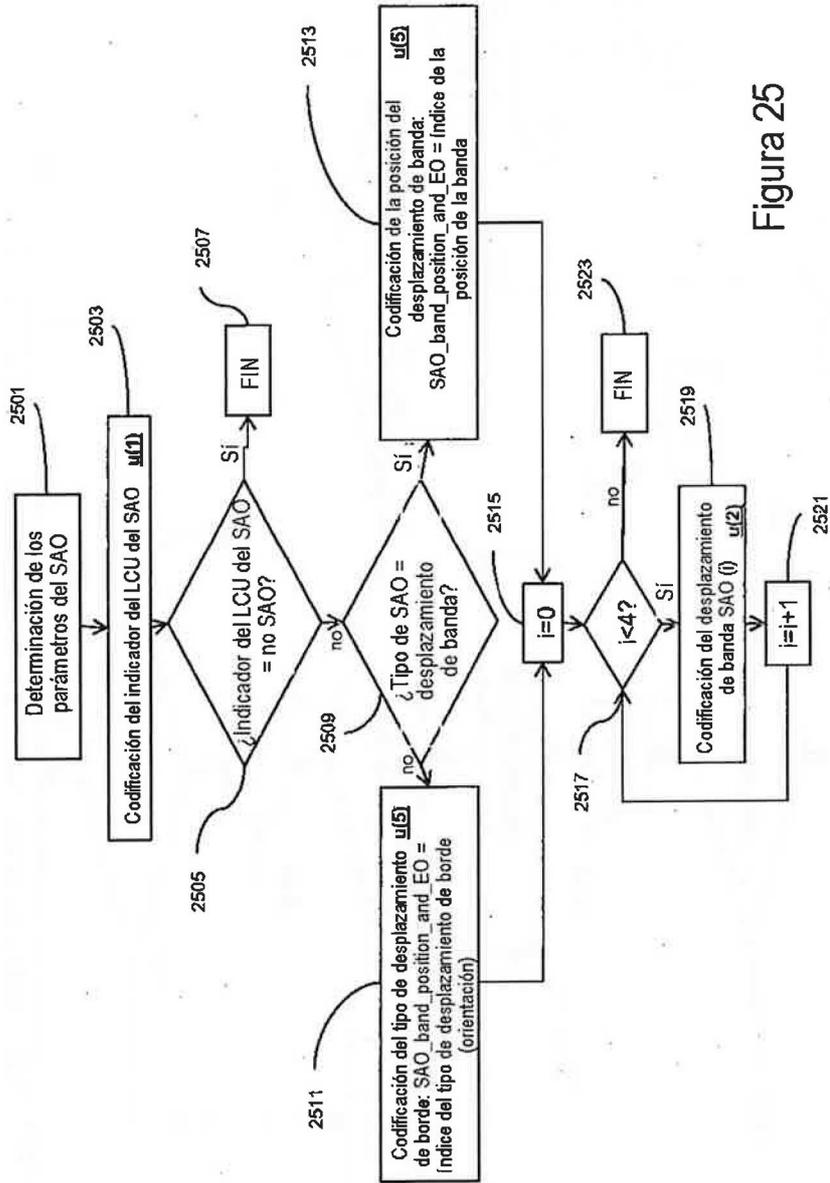


Figura 25

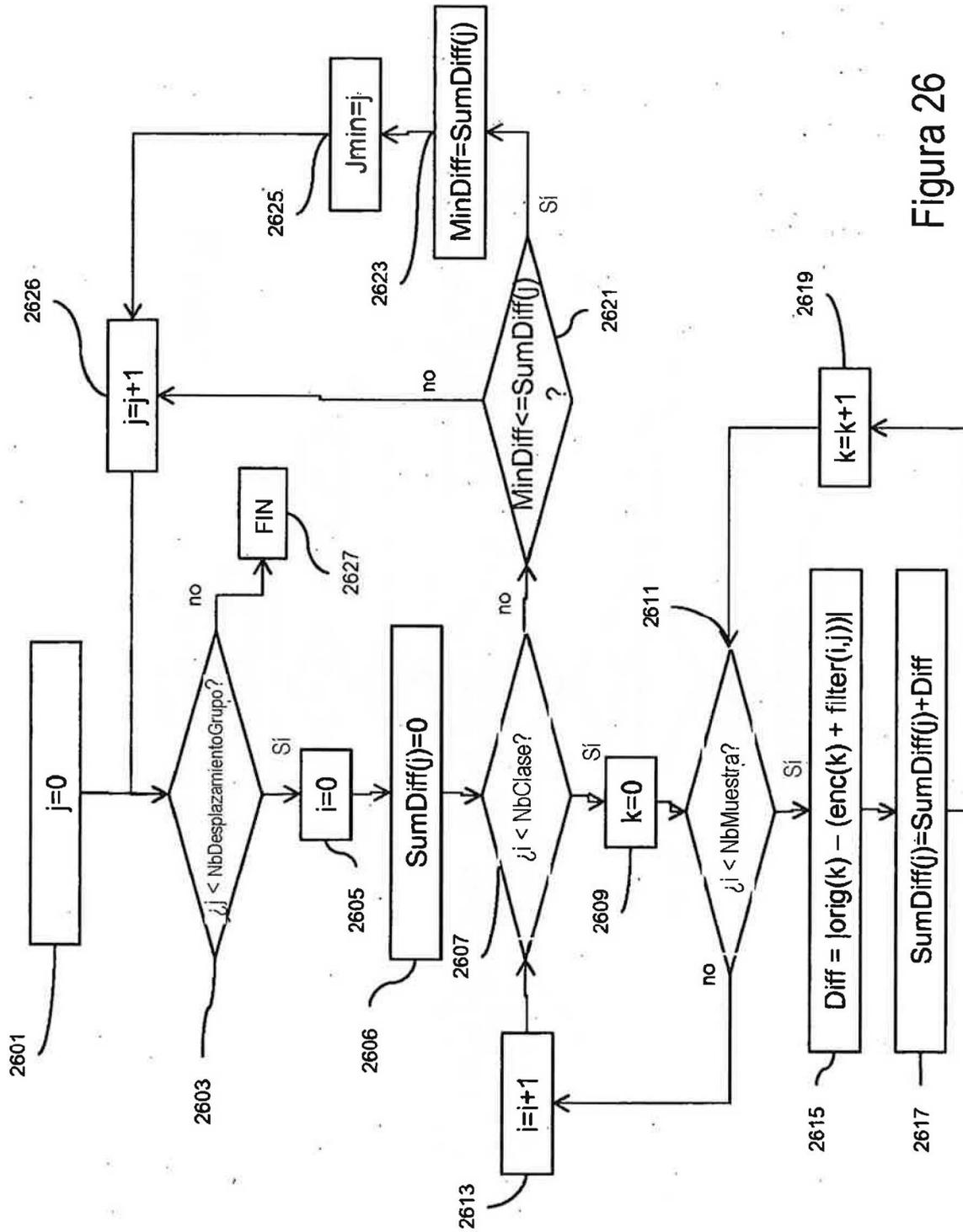


Figura 26

REFERENCIAS CITADAS EN LA DESCRIPCIÓN

5 *Esta lista de referencias citada por el solicitante es únicamente para mayor comodidad del lector. No forman parte del documento de la Patente Europea. Incluso teniendo en cuenta que la compilación de las referencias se ha efectuado con gran cuidado, los errores u omisiones no pueden descartarse; la EPO se exime de toda responsabilidad al respecto.*

Documentos de patentes citados en la descripción

- Sample adaptive offset for HEVC. **MING FU et al.**
Multimedia Signal Processing (MMSP), 2011 IEEE
13th International Workshop On. IEEE, 17 October
2011, 1-5