

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 906 155**

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 19218277 (2)**

97 Fecha y número de publicación de la concesión europea: **22.12.2021** **EP 3657790**

54 Título: **Procedimiento y dispositivo para optimizar la codificación/descodificación de desplazamientos de compensación para un conjunto de muestras reconstruidas de una imagen**

30 Prioridad:

07.11.2011 GB 201119206

24.02.2012 GB 201203285

26.03.2012 GB 201205304

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

13.04.2022

73 Titular/es:

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

72 Inventor/es:

LAROCHE, GUILLAUME;
GISQUET, CHRISTOPHE;
FRANCOIS, EDOUARD y
ONNO, PATRICE

74 Agente/Representante:

DURAN-CORRETJER, S.L.P

ES 2 906 155 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 optimizar la codificación/descodificación de desplazamientos de compensación para un conjunto de muestras reconstruidas de una imagen

La presente invención se refiere a un procedimiento y un dispositivo para optimizar la codificación y/o la descodificación de desplazamientos de compensación para un conjunto de muestras reconstruidas de una imagen, con el fin de mejorar la eficiencia de la codificación de un codificador-descodificador de video y/o la complejidad de la descodificación. La invención se refiere además a un procedimiento y un dispositivo para codificar o descodificar una secuencia de imágenes digitales.

La invención se puede aplicar en el sector técnico del procesamiento de señales digitales, y en particular en el sector técnico de la compresión de video utilizando compensación del movimiento para reducir las redundancias espaciales y temporales en flujos de video.

Muchos formatos de compresión de video, tales como por ejemplo H.263, H.264, MPEG-1, MPEG-2, MPEG-4 y SVC utilizan la transformada de coseno discreta (DCT, discrete cosine transform) basada en bloques y la compensación del movimiento para eliminar redundancias espaciales y temporales. Estos son denominados a menudo formatos de video predictivos. Cada cuadro o imagen de la señal de video es dividida en segmentos que son codificados y pueden ser descodificados independientemente. Un segmento es habitualmente una parte rectangular del cuadro, o más generalmente, una parte de un cuadro o un cuadro completo. Además, cada segmento puede ser dividido en macrobloques (MBs), y cada macrobloque se divide además en bloques, habitualmente bloques de 64x64, 32x32, 16x16 u 8x8 píxeles.

En la codificación de video de alta eficiencia (HEVC, High Efficiency Video Coding) se pueden utilizar bloques desde 64x64 hasta 4x4. La división se organiza de acuerdo con una estructura de árbol cuádruple basada en la unidad máxima de codificación (LCU, largest coding unit). Una LCU corresponde, por ejemplo, a un bloque cuadrado de 64x64. Si una LCU tiene que ser dividida, un indicador de división indica que la LCU se divide en 4 bloques de 32x32. Del mismo modo, si alguno de estos 4 bloques tiene que ser dividido, el indicador de división se pone a verdadero, y el bloque de 32x32 se divide en 4 bloques de 16x16, etc. Cuando un 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 u 8x8 píxeles.

Existen dos familias de modos de codificación para la codificación de bloques 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 respecto del bloque original.

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

La predicción temporal consiste en encontrar un cuadro de referencia, ya sea un cuadro de referencia anterior o uno futuro de la secuencia de video, una parte de imagen o un área de referencia que sea la más próxima al bloque a codificar. Esta etapa se conoce habitualmente como estimación del movimiento. A continuación, se predice el bloque a codificar utilizando el área de referencia en una etapa denominada habitualmente 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.

Para reducir adicionalmente el coste de codificar la información del 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 la diferencia entre el vector de movimiento y un predictor del vector de movimiento, compuesto habitualmente de uno o varios vectores de movimiento de los bloques que rodean el bloque a codificar.

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

La codificación utilizando vectores de movimiento residuales ahorra algo de tasa de bits, pero requiere que el descodificador realice el mismo cálculo del predictor del vector de movimiento para descodificar el valor del vector de movimiento de un bloque a descodificar.

Ambos procesos de codificación y descodificación pueden implicar un proceso de descodificación de una imagen codificada. Este proceso se lleva a cabo habitualmente en el lado del codificador, con el objetivo de una estimación

del movimiento futuro que permite a un codificador y al correspondiente descodificador tener los mismos cuadros de referencia.

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, la primera reconstrucción se filtra mediante una o varias clases de procesos de filtrado posterior. Estos filtros posteriores se aplican sobre el cuadro reconstruido, en el lado del codificador y el del descodificador, para que se utilice el mismo cuadro de referencia en ambos lados. El objetivo de este filtrado posterior es eliminar artefactos de compresión y mejorar la calidad de la imagen. Por ejemplo, H.264/AVC utiliza un filtro anti-efecto bloque. Este filtro puede eliminar los artefactos de efecto bloque debidos a la cuantificación DCT de los residuos 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 de un proceso de filtrado de bucle de una implementación conocida de HEVC. En una etapa inicial 101, el codificador o el descodificador genera la reconstrucción del cuadro completo. A continuación, en la etapa 102 se aplica un filtro anti-efecto bloque sobre esta primera reconstrucción, para regenerar una reconstrucción sin bloques 103. El objetivo del filtro anti-efecto bloque es eliminar artefactos de bloque generados mediante la cuantificación de residuos y la compensación del movimiento de bloques o la predicción de bloques Intra. Estos artefactos son visualmente importantes a tasas de bit bajas. El filtro anti-efecto bloque actúa para suavizar los límites de los bloques en función de las características de dos bloques contiguos. Se tiene en cuenta el modo de codificación de cada bloque, los parámetros de cuantificación utilizados para la codificación de los residuos y las diferencias de los 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 artefactos de efecto bloque, y mejora asimismo la estimación del movimiento y la compensación del movimiento para los cuadros subsiguientes. De hecho, se eliminan las altas frecuencias de los artefactos de bloque, y por lo tanto no es necesario compensar estas frecuencias con la textura residual de los cuadros siguientes.

Después del filtro anti-efecto bloque, la reconstrucción sin bloques es filtrada mediante un filtro de bucle de desplazamiento adaptativo de muestras (SAO) en la etapa 104. El cuadro resultante 105 se filtra a continuación con un filtro de bucle adaptativo (ALF) en la etapa 106, para generar el cuadro reconstruido 107 que será visualizado y utilizado como cuadro de referencia para los siguientes cuadros *Inter*.

El objetivo del filtro de bucle SAO y del ALF es mejorar la reconstrucción de los cuadros enviando datos adicionales, a diferencia del filtro anti-efecto bloque en que no se transmite ninguna información.

El principio del filtro de bucle SAO es clasificar cada píxel en una clase y sumar 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 SAO proporciona dos clases de clasificación para un área del cuadro: desplazamiento de borde y 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). La dirección que se tiene que utilizar se proporciona mediante un parámetro SAO denominado *sao_type_idx* en las actuales especificaciones HEVC. Habitualmente, su valor varía de cero a cinco, tal como se muestra en la tabla 1 del Apéndice. El *sao_type_idx* se utiliza asimismo para indicar si se tiene o no que llevar a cabo filtrado SAO y el tipo de filtrado SAO.

A título ilustrativo, el desplazamiento que se tiene que añadir a un valor de píxel (o muestra) C se puede determinar, para una dirección determinada, según las reglas indicadas en la tabla 2 del Apéndice, en la que Cn_1 y Cn_2 indican el valor de los dos píxeles contiguos o muestras (en función de la dirección determinada). Por consiguiente, cuando el valor C es menor que los dos valores Cn_1 y Cn_2 , el desplazamiento que se tiene que añadir a C es $+O_1$, cuando el valor es menor que Cn_1 o Cn_2 e igual al otro valor (Cn_1 o Cn_2), el desplazamiento que se tiene que utilizar es $+O_2$, cuando es mayor que Cn_1 o Cn_2 e igual que el otro valor (Cn_1 o Cn_2), el desplazamiento se tiene que utilizar es $-O_3$, y cuando es mayor que Cn_1 y Cn_2 , el desplazamiento que se tiene que utilizar es $-O_4$. Cuando no se cumple ninguna de estas condiciones, no se añade ningún valor de desplazamiento al valor de píxel actual C .

Se debe observar que, en función del modo de desplazamiento de borde, solamente el valor absoluto de cada desplazamiento se codifica en el flujo de bits, determinándose el signo a aplicar en función de la categoría a la que pertenece el píxel actual. Por lo tanto, de acuerdo con la tabla 2 del Apéndice, se asocia un desplazamiento positivo con las categorías 1 y 2 mientras que se asocia un desplazamiento negativo con las categorías 3 y 4.

El segundo tipo de clasificación es una clasificación de desplazamiento de banda, que depende del valor de píxel. Una clase en un desplazamiento de banda 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 comprendido en un intervalo determinado de valores de píxel.

Para que sea más adaptativo al contenido del cuadro, el filtrado SAO se aplica sobre varias áreas del cuadro, lo que divide el cuadro actual en varias zonas espaciales. Actualmente, las áreas del cuadro corresponden a un número finito de la unidad máxima de codificación en HEVC. Por consiguiente, un área del cuadro puede o no ser filtrada mediante SAO, de tal modo que solamente algunas áreas del cuadro se filtran. Además, cuando SAO está activado, solamente se utiliza una clasificación SAO: desplazamiento de borde o desplazamiento de banda, en función de los parámetros relacionados, transmitidos para cada clasificación. Finalmente, para cada nodo hoja SAO, se transmite la clasificación SAO así como sus parámetros y los desplazamientos de todas las clases.

La imagen de datos de video a codificar se puede proporcionar como un conjunto de matrices bidimensionales (conocidas asimismo como canales de color) de valores de muestra, de las que cada entrada representa la intensidad de un componente de color, tal como una medida de la intensidad de la luma y la intensidad de la croma desde un color neutro de la escala de grises hacia azul o rojo (YUV), o como una medida de la intensidad del componente de luz roja, verde o azul (RGB). Un modelo YUV define un espacio de color en términos de un componente de luma (Y) y dos componentes de croma (UV). Generalmente, Y se refiere al componente de luma y U y V son los componentes de croma (color).

Habitualmente, el filtrado SAO se aplica independientemente sobre el componente de luma y ambos componentes de croma U y V.

Una implementación conocida del desplazamiento de banda SAO divide el intervalo de valores de píxel en 32 intervalos predefinidos del mismo tamaño, tal como se representa en la **figura 2**. El valor mínimo del intervalo de valores de píxel es siempre cero, y el valor máximo depende de la profundidad de bits de los valores de píxel según la siguiente relación $\text{Max} = 2^{\text{Profundidad de bits}} - 1$.

La división de todo el intervalo de valores de píxel en 32 intervalos permite la utilización de cinco bits para clasificar cada píxel, permitiendo una clasificación rápida. Por consiguiente, solamente se comprueban 5 bits para clasificar un píxel en una de las 32 clases o intervalos del intervalo completo. Esto se realiza generalmente comprobando los cinco bits más significativos de los valores codificados en 8 bits.

Por ejemplo, cuando la profundidad de bits es de 8 bits, el valor máximo posible de un píxel es de 255. Por lo tanto, el intervalo de valores de píxel está entre 0 y 255. Para esta profundidad de bits de 8 bits, cada clase incluye un intervalo de 8 valores de píxel.

El objetivo del filtrado de banda SAO es el filtrado de píxeles pertenecientes a un grupo de cuatro intervalos o clases consecutivas que está determinado por la primera clase o banda. Ésta se transmite en el flujo de bits, de tal modo que el descodificador puede determinar los cuatro intervalos o clases consecutivas de píxeles a filtrar. El parámetro que representa esta posición es denominado `sao_band_position` en las actuales especificaciones HEVC.

A título ilustrativo, en la figura 2 se representa como un área gris un grupo de cuatro intervalos o clases consecutivas 201 a 204 de píxeles a filtrar. Tal como se ha descrito anteriormente, este grupo se puede identificar mediante su posición (por ejemplo, `sao_band_position`). Según el ejemplo facilitado, una clase o intervalo 201 se refiere a píxeles que tienen valores comprendidos entre 64 y 71. Análogamente, las clases o intervalos 202 a 204 se refieren a píxeles que tienen valores comprendidos entre 72 y 79, 80 y 87, 88 y 96, respectivamente.

La **figura 3** es un diagrama de flujo que muestra las etapas de un procedimiento para seleccionar desplazamientos en un codificador para el área del cuadro actual 303 (normalmente, un bloque LCU correspondiente a un componente de la imagen procesada). El área del cuadro contiene N píxeles. En una etapa inicial 301, las variables Sum_j y SumNbPix_j se ajustan a un valor cero para cada uno de los cuatro intervalos. j indica el número de clase o intervalo actual. Sum_j indica la suma de la diferencia entre el valor de los píxeles en el intervalo j y el valor de sus correspondientes píxeles originales. SumNbPix_j indica el número de píxeles en el intervalo j .

En la etapa 302, la variable de cómputo i se ajusta al valor cero. A continuación, el siguiente píxel P_i del área del cuadro 303 se obtiene en la etapa 304, y el número de clase J correspondiente al píxel actual P_i se obtiene en la etapa 305. A continuación, se realiza una prueba en la etapa 306 para determinar si el número de clase J del píxel actual P_i corresponde o no al valor "N.A.", tal como se ha descrito anteriormente haciendo referencia a la tabla 2 del Apéndice. Si el número de clase J del píxel actual P_i corresponde al valor "N.A.", el valor de la variable de cómputo i se aumenta en una unidad para clasificar los píxeles subsiguientes en el área del cuadro 303. De lo contrario, si el número de clase J del píxel actual P_i no corresponde al valor "N.A.", la variable SumNbPix_j correspondiente al píxel actual P_i se aumenta en una unidad y la diferencia entre P_i y su valor original P_i^{org} se añade a Sum_j en la etapa 307.

En la siguiente etapa, la variable de cómputo i se aumenta en una unidad para aplicar la clasificación a los otros 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 Offset_j (desplazamiento) para cada clase en la etapa 310 con el fin de producir una tabla de desplazamientos 311 que presenta un desplazamiento para cada clase j , como resultado final del algoritmo de selección de desplazamientos. Este desplazamiento se calcula como el

promedio de la diferencia entre los valores de píxel de los píxeles de clase j y sus respectivos valores de píxel originales. El $Offset_j$ para la clase j viene dado por la siguiente ecuación:

$$Offset_j = \frac{Sum_j}{SumNbPix_j}$$

- 5 El desplazamiento calculado $Offset_j$ se puede considerar un desplazamiento óptimo en términos de distorsión. En lo que sigue, éste es denominado O_{optj} . A partir de este desplazamiento, es posible determinar un valor de desplazamiento mejorado O_{RDj} en función de un criterio de tasa-distorsión.

10 La **figura 4** es un diagrama de flujo que muestra las etapas de un procedimiento para determinar un desplazamiento mejorado según un criterio de tasa-distorsión. En una etapa inicial 401, el valor de tasa-distorsión J_j del número de clase o intervalo actual j se inicializa a un valor posible máximo predeterminado (MAX_VALUE).

15 A continuación, se lanza un bucle en la etapa 402 para hacer que el desplazamiento O_j varíe desde O_{optj} hasta cero. Si el valor O_{optj} es negativo, la variable O_j se aumenta en una unidad hasta llegar a cero, y si el valor O_{optj} es positivo, la variable O_j se disminuye en una unidad hasta llegar a cero.

En la etapa 403, se calcula el coste de tasa-distorsión relacionado con la variable O_j , denominado $J(O_j)$, por ejemplo según la fórmula siguiente:

$$20 \quad J(O_j) = SumNbPix_j \times O_j \times O_j - Sum_j \times O_j \times 2 + \lambda R(O_j)$$

λ es el parámetro de Lagrange y $R(O_j)$ es una función que proporciona el número de bits necesarios para la palabra de código asociada a O_j . La parte de la fórmula correspondiente a $SumNbPix_j \times O_j \times O_j - Sum_j \times O_j \times 2$ se refiere a la mejora en términos de distorsión proporcionada por el desplazamiento O_j .

25 En la etapa 404, los valores $J(O_j)$ y J_j son comparados entre sí. Si el valor $J(O_j)$ es menor que el valor J_j , entonces J_j se ajusta al valor de $J(O_j)$ y O_{RDj} se ajusta al valor de O_j .

30 En la etapa 405, se determina si se han procesado o no todos los posibles valores del desplazamiento O_j (es decir, si $O_j = 0$). Si el desplazamiento O_j es igual a cero, el bucle finaliza y se ha identificado un valor de desplazamiento mejorado (O_{RDj}) para la clase j . De lo contrario, el bucle continúa con el siguiente valor O_j .

35 Se debe observar que el algoritmo descrito haciendo referencia a la figura 3 puede ser utilizado para determinar una posición de una primera clase o intervalo (sao_band_position) en función del tipo de clasificación de desplazamiento de banda. A este respecto, el índice j representa un valor del intervalo [0, 32] (en lugar de [1, 4]). En otras palabras, el valor 4 es sustituido por el valor 32 en los módulos 301, 310 y 311 de la figura 3.

40 Más específicamente, se puede calcular la diferencia Sum_j entre el valor del píxel actual y su valor original $Porg_j$ para cada una de las 32 clases representadas en la figura 2, es decir para cada intervalo j (perteneciendo j al intervalo [0, 32]).

A continuación, se puede calcular un desplazamiento mejorado O_{RDj} en términos de la tasa-distorsión para las 32 clases, según un algoritmo similar al descrito haciendo referencia a la figura 4.

45 A continuación se puede determinar la posición de la primera clase o intervalo.

50 La **figura 5** es un diagrama de flujo que muestra las etapas de un procedimiento para la determinación de una posición de banda SAO para el desplazamiento de banda SAO de HEVC. Dado que estas etapas se llevan a cabo después de las descritas haciendo referencia a la figura 4, el valor de tasa-distorsión indicado como J_j se ha calculado ya para cada clase j .

En una etapa inicial 501, el valor J de tasa-distorsión se inicializa a un valor posible máximo predeterminado (MAX_VALUE). A continuación, se lanza un bucle en la etapa 502 para hacer que el índice i varíe de cero a 27, correspondiente a las 28 posibles posiciones de la primera clase del grupo de cuatro clases consecutivas.

55 En la etapa 503, la variable J'_i correspondiente al valor de tasa-distorsión de la banda actual, es decir la banda que comprende cuatro clases consecutivas desde la clase que tiene el índice i , se inicializa a cero. A continuación, se lanza un bucle en la etapa 504 para hacer que el índice j varíe de i a $i+3$, lo que corresponde a las 4 clases de la banda actual.

60 Después, en la etapa 505, el valor de la variable J'_i se incrementa mediante el valor de la tasa-distorsión de la clase que tiene el índice j (J_j). Esta etapa se repite para las cuatro clases de la banda actual, es decir hasta que el índice j llega a $i+3$ (etapa 506). En la etapa 507, se realiza una prueba para determinar si el valor de tasa-distorsión J'_i de la

banda actual es o no menor que el valor de tasa-distorsión J . Si el valor de la tasa-distorsión J'_i de la banda actual es menor que el valor de tasa-distorsión J , el valor de la tasa-distorsión se ajusta al valor de la tasa-distorsión J'_i de la banda actual, y el valor de posición de la banda denominado `sao_band_position` se ajusta al valor del índice i .

- 5 Estas etapas se repiten para las 28 posibles posiciones de la primera clase del grupo de cuatro clases consecutivas (etapa 508) para determinar la posición de la banda a utilizar.

10 La **figura 6** es un diagrama de flujo que muestra las etapas de un procedimiento para el filtrado de un área del cuadro, habitualmente un bloque LCU correspondiente a un componente de una imagen procesada, de acuerdo con un filtro de bucle SAO.

Dicho algoritmo se implementa generalmente en el descodificador para descodificar cuadros, y en el codificador para generar cuadros de referencia que se utilizan para estimación y compensación del movimiento de los cuadros siguientes.

15 En una etapa inicial 601, se obtienen parámetros de filtrado SAO, por ejemplo a partir de un flujo de bits recibido. Para un área del cuadro determinada, estos parámetros comprenden habitualmente cuatro desplazamientos que pueden ser almacenados en la tabla 603 y un parámetro del `sao_type_idx`. Dependiendo de éste último, dichos parámetros pueden comprender además una posición de banda SAO. A título ilustrativo, si un parámetro obtenido del tipo `sao_type_idx` es igual a cinco, se puede obtener (602) el correspondiente parámetro del tipo `sao_band_position`. Se pueden obtener otros parámetros como los que permiten determinar el valor de un parámetro SAO de un área del cuadro determinada, en función del valor del correspondiente parámetro de un área del cuadro anterior. Se debe tener en cuenta que un valor determinado de un parámetro SAO dado, tal como el valor cero para el parámetro `sao_type_idx`, puede indicar que no se tiene que aplicar un filtrado SAO.

25 En la etapa 604, la variable de cómputo i se ajusta al valor cero. A continuación, el primer píxel P_i del área del cuadro 605, que comprende N píxeles, es obtenido en la etapa 606 y clasificado en la etapa 607 de acuerdo con la clasificación de desplazamiento de borde o la clasificación de desplazamiento de banda, tal como se ha descrito anteriormente.

30 En la etapa 608 se realiza una prueba para determinar si el píxel P_i pertenece o no a la clase de píxeles a filtrar. Si el píxel P_i pertenece a la clase de píxeles a filtrar, se identifica su número de clase j relacionado y se obtiene su valor de desplazamiento relacionado $Offset_j$ en la etapa 610, a partir de la tabla de desplazamientos 603. A continuación, en la etapa 611, se añade el $Offset_j$ al valor de píxel P_i para producir un nuevo valor de píxel denominado P'_i (612). En la etapa 613, el píxel P'_i sustituye el píxel P_i en el área procesada 616 del cuadro. De lo contrario, si el píxel P_i no pertenece a la clase de píxeles a filtrar, el píxel P_i 609 permanece sin cambios en el área del cuadro, en la etapa 613.

40 A continuación, después de haber procesado el píxel P_i , la variable de cómputo i se aumenta en una unidad en la etapa 614, para aplicar el filtro de manera similar al siguiente píxel del área del cuadro actual 605.

En la etapa 615 se determina si han sido o no procesados todos los N píxeles del área del cuadro actual 605 ($i \geq N$). En caso afirmativo, el área procesada 616 del cuadro se reconstruye y se puede añadir al cuadro reconstruido SAO, tal como se ha descrito anteriormente haciendo referencia a la figura 1 (referencia 105).

45 Hay información adicional sobre propuestas para filtrado SAO en HEVC en el documento de Chinh-Ming-Fu y otros: "Sample adaptive offset for HEVC" ("desplazamiento adaptativo de muestras para HEVC"), MULTIMEDIA SIGNAL PROCESSING (MMSP), 2011, IEEE 13TH INTERNATIONAL WORKSHOP ON, IEEE, 17 de octubre de 2011 (17/10/2011), páginas 1 a 5, XP032027547.

50 Un inconveniente del proceso conocido para la selección y transmisión de compensaciones es que es necesario transmitir un gran número de parámetros del codificador al descodificador y almacenarlos en una memoria tampón SAO. Además, el proceso llevado a cabo por el descodificador para descodificar y procesar estos parámetros es complejo.

55 La presente invención ha sido concebida para solucionar uno o varios de los problemas mencionados.

60 Teniendo en cuenta las interdependencias entre componentes, la invención hace posible mejorar la eficiencia de codificación de SAO, reducir la memoria tampón necesaria para almacenar parámetros SAO y reducir la complejidad de la clasificación.

Según un primer aspecto de la presente invención, se da a conocer un procedimiento de descodificación, tal como se define mediante las reivindicaciones 1 a 7.

65 Según un segundo aspecto de la presente invención, se da a conocer un dispositivo de descodificación, tal como se define mediante las reivindicaciones 10 a 16.

Por lo menos, partes de los procedimientos según la invención se pueden implementar mediante ordenador. Por consiguiente, la presente invención puede adoptar la forma de una realización íntegramente de hardware, una realización íntegramente de software (incluyendo software inalterable, software residente, microcódigo, etc.) o una realización que combine aspectos de software y hardware, que en general se pueden denominar todas ellas conjuntamente en la presente memoria 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 un código de programa utilizable por ordenador incorporado en el medio.

Dado que la presente invención se puede implementar en software, la presente invención se puede realizar como código legible por ordenador para suministrarla a un aparato programable en cualquier medio portador adecuado. Un medio portador tangible puede comprender un medio de almacenamiento tal como un disco flexible, 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 RF.

Por lo tanto, según un quinto aspecto de la presente invención, se da a conocer un producto de programa informático para un aparato programable, tal como se define mediante la reivindicación 8.

De manera similar, según un sexto aspecto de la presente invención, se da a conocer un medio de almacenamiento legible por ordenador, tal como se define mediante la reivindicación 9.

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

la **figura 1** es un diagrama de flujo que muestra las etapas de un proceso de filtrado de bucle de la técnica anterior; la **figura 2** muestra gráficamente una clasificación de desplazamiento de banda adaptativo de muestras de un proceso HEVC de la técnica anterior;

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

la **figura 4** es un diagrama de flujo que muestra un proceso para seleccionar un desplazamiento SAO desde un punto de vista de tasa-distorsión;

la **figura 5** es un diagrama de flujo que muestra las etapas de un procedimiento para la determinación de una posición de banda SAO para el desplazamiento de banda SAO de HEVC.

la **figura 6** es un diagrama de flujo que muestra las etapas de un procedimiento para el filtrado de un área del cuadro, según un filtro de bucle SAO;

la **figura 7** es un diagrama de bloques que muestra esquemáticamente un sistema de comunicación de datos en el que se pueden implementar una o varias realizaciones de la invención;

la **figura 8** es un diagrama de bloques que muestra componentes de un dispositivo de procesamiento en el que se pueden implementar una o varias realizaciones de la invención;

la **figura 9** es un diagrama de flujo que muestra las etapas de un procedimiento de codificación;

la **figura 10** es un diagrama de flujo que muestra las etapas de un proceso de filtrado de bucle;

la **figura 11** es un diagrama de flujo que muestra las etapas de un procedimiento de decodificación, según realizaciones de la invención;

la **figura 12** es un diagrama de flujo que muestra las etapas de un procedimiento para obtener parámetros SAO a partir de un flujo de bits, de acuerdo con un modo de un conjunto de parámetros adaptativos;

la **figura 13** es un diagrama de flujo que muestra las etapas de un procedimiento para obtener parámetros SAO a partir de un flujo de bits, de acuerdo con un modo LCU;

la **figura 14** es un diagrama de flujo que muestra las etapas de un procedimiento para leer parámetros SAO en un flujo de bits;

las **figuras 15 y 16** son diagramas de flujo que muestran etapas de un procedimiento para determinar parámetros a utilizar en un filtro de bucle para el procesamiento de un área del cuadro actual;

la **figura 17** muestra un ejemplo según el cual los valores de los desplazamientos asociados con un componente corresponden a los valores especulares de los desplazamientos asociados con un segundo componente;

la **figura 18** muestra cómo se pueden obtener valores de desplazamientos asociados con un segundo componente, a partir de valores de desplazamientos asociados con un primer componente, cuando se aplica un filtro de bucle SAO utilizando una clasificación de desplazamiento de banda;

la **figura 19** muestra cómo se puede obtener un valor de un parámetro `sao_band_position` asociado con un segundo componente Z, a partir del valor del parámetro `sao_band_position` asociado con un primer componente X, cuando se aplica un filtro de bucle SAO utilizando una clasificación de desplazamiento de banda;

la **figura 20** es un diagrama de flujo que muestra las etapas de un procedimiento para determinar un desplazamiento óptimo a utilizar por dos o tres componentes, de acuerdo con un criterio de tasa-distorsión;

la **figura 21** es un diagrama de flujo que muestra las etapas de un procedimiento para determinar una posición de banda SAO compartida entre dos componentes;

la **figura 22** es un diagrama de flujo que muestra las etapas de un procedimiento para decodificar un flujo de bits; y

la **figura 23** muestra la utilización de clasificación SAO de un componente X para filtrar de manera similar otro componente Z.

La **figura 7** 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 701, que puede funcionar para transmitir paquetes de datos de un flujo de datos a un dispositivo receptor, en este caso el terminal cliente 702, por medio de una red de comunicación de datos 700. La red de comunicación de datos 700 puede ser una red de área extensa (WAN, Wide area Network) o una red de área local (LAN, Local Area Network). Una red de este tipo puede ser, por ejemplo, una red inalámbrica (Wifi / 802.11a ó b, ó g), una red Ethernet, una red de internet o una red mixta compuesta de varias redes diferentes. En una realización particular de la invención, el sistema de comunicación de datos puede ser un sistema de difusión de televisión digital en el que el servidor 701 envía el mismo contenido de datos a múltiples clientes.

El flujo de datos 704 proporcionado por el servidor 701 puede estar compuesto de datos multimedia que representan datos de video y de audio. En algunas realizaciones de la invención, los flujos de datos de audio y de video pueden ser capturados por el servidor 701 utilizando un micrófono y una cámara, respectivamente. En algunas realizaciones, los flujos de datos pueden ser almacenados en el servidor 701 o recibidos por el servidor 701 desde otro proveedor de datos, o generados en el servidor 701. El servidor 701 está dotado de un codificador para codificar flujos de video y de audio, en particular para proporcionar un flujo de bits comprimido para transmisión, que es una representación más compacta de los datos presentados como entrada al codificador.

Para obtener una mejor relación de la calidad de los datos transmitidos frente a la cantidad de datos transmitidos, la compresión de los datos de video puede ser, por ejemplo, acorde con el formato HEVC o con el formato H.264/AVC.

El cliente 702 recibe el flujo de bits transmitido y descodifica el flujo de bits reconstruido para reproducir imágenes de video en un dispositivo de visualización, y datos de audio mediante un altavoz.

Aunque en el ejemplo de la figura 7 se considera un escenario de transmisión continua, se apreciará que en algunas realizaciones de la invención la comunicación de datos entre el codificador y el 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, una imagen de video es transmitida con datos que representan desplazamientos de compensación para su aplicación a píxeles reconstruidos de la imagen con el fin de proporcionar píxeles filtrados en la imagen final.

La **figura 8** muestra esquemáticamente un dispositivo de procesamiento 800 configurado para implementar, por lo menos, una realización de la presente invención. El dispositivo de procesamiento 800 puede ser un dispositivo, tal como un microordenador, una estación de trabajo o un dispositivo portátil ligero. El dispositivo 800 comprende un bus de comunicación 813 conectado a:

- una unidad central de procesamiento 811, tal como un microprocesador, denominada CPU;
- una memoria de sólo lectura 807, denominada ROM, para almacenar programas informáticos para la implementación de la invención;
- una memoria de acceso aleatorio 812, denominada RAM, para almacenar el código ejecutable del procedimiento de las realizaciones de la invención, así como los registros adaptados para registrar 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 comunicación 802 conectada a una red de comunicación 803 sobre la que son transmitidos o recibidos datos a procesar.

Opcionalmente, el aparato 800 puede incluir asimismo los siguientes componentes:

- un medio de almacenamiento de datos 804, tal como un disco duro, para almacenar programas informáticos para la implementación de procedimientos de una o varias realizaciones de la invención, y datos utilizados o producidos durante la implementación de una o varias realizaciones de la invención;
- una unidad de disco 805 para un disco 806, estando adaptada la unidad de disco para leer datos del disco 806 o escribir datos en dicho disco;
- una pantalla 809 para visualizar datos y/o servir como interfaz gráfica con el usuario, por medio de un teclado 810 o cualquier otro medio de señalización.

El aparato 800 puede estar conectado a varios periféricos, tales como por ejemplo una cámara digital 820 o un micrófono 808, estando cada uno conectado a una tarjeta de entrada/salida (no mostrada) para suministrar datos multimedia al aparato 800.

El bus de comunicación proporciona comunicación e interoperabilidad entre los diversos elementos incluidos en el aparato 800 o conectados al mismo. La representación del bus no es limitativa, y en particular la unidad central de proceso puede funcionar para comunicar instrucciones a cualquier elemento del aparato 800, directamente o por

medio de otro elemento del aparato 800.

El disco 806 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 un medio de almacenamiento de información que pueda ser leído por un microordenador o por un microprocesador, integrado o no en el aparato, posiblemente extraíble y adaptado para almacenar uno o varios programas cuya ejecución permite implementar el procedimiento de codificar una secuencia de imágenes digitales y/o el procedimiento de decodificar un flujo de bits, según la invención.

El código ejecutable puede estar almacenado bien en la memoria de sólo lectura 807, el disco duro 804 o en un medio digital extraíble, tal como por ejemplo un disco 806 como el descrito anteriormente. Según una variante, el código ejecutable de los programas puede ser recibido por medio de la red de comunicación 803, por medio de la interfaz 802, para ser almacenado en uno de los medios de almacenamiento del aparato 800 antes de ser ejecutado, tal como el disco duro 804.

La unidad central de procesamiento 811 está adaptada para controlar y dirigir la ejecución de las instrucciones o partes del código de software del programa o programas según la invención, instrucciones que están almacenadas en uno de los medios de almacenamiento mencionados anteriormente. Al encendido, el programa o programas que están almacenados en una memoria no volátil, por ejemplo en el disco duro 804 o en la memoria de sólo lectura 807, son transferidos a la memoria de acceso aleatorio 812, que contiene a continuación el código ejecutable del programa o programas, así como 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. Sin embargo, alternativamente, la presente invención se puede implementar en hardware (por ejemplo, en forma de un circuito integrado de aplicación específica o ASIC).

La **figura 9** muestra un diagrama de bloques de un codificador 900. El codificador está representado por módulos conectados, estando adaptado cada módulo para implementar, por ejemplo en forma de instrucciones de programación a ejecutar mediante la CPU 811 del dispositivo 800, por lo menos, la correspondiente etapa de un procedimiento de codificación de una imagen de una secuencia de imágenes descrito en la presente memoria.

Una secuencia original de imágenes digitales i0 a in 901 es recibida como entrada por el codificador 900. Cada imagen digital está representada por un conjunto de muestras, conocidas como píxeles.

Un flujo de bits 910 es emitido por el codificador 900 después de la implementación del proceso de codificación. El flujo de bits 910 comprende una serie de unidades de codificación o segmentos, comprendiendo cada segmento la cabecera del segmento para transmitir valores de codificación de parámetros de codificación utilizados para codificar el segmento, y el cuerpo del segmento que comprende datos de video codificados.

Las imágenes digitales de entrada i0 a in 901 son divididas en bloques de píxeles mediante el módulo 902. Los bloques corresponden a partes de la imagen y pueden ser de tamaños variables (por ejemplo 4x4, 8x8, 16x16, 32x32, 64x64 píxeles). Se selecciona un modo de codificación para cada bloque de entrada. Están dispuestas 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 903 implementa un proceso de predicción Intra, en el que el bloque determinado para ser codificado se predice mediante un predictor calculado a partir de píxeles de las proximidades de dicho bloque a codificar. Si se selecciona codificación Intra, se codifica una indicación del predictor Intra seleccionado y la diferencia entre el bloque determinado y su predictor, para proporcionar un residuo.

Se implementa la predicción temporal mediante el módulo 904 de estimación del movimiento y el módulo 905 de compensación del movimiento. En primer lugar, se selecciona una imagen de referencia de entre un conjunto de imágenes de referencia 916, y mediante el módulo 904 de estimación del movimiento se selecciona 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 para ser codificado. A continuación, el módulo 905 de compensación del movimiento predice el bloque a codificar utilizando el área seleccionada. La diferencia entre el área de referencia seleccionada y el bloque determinado, denominada asimismo bloque residual, se calcula mediante el módulo 905 de compensación del movimiento. El área de referencia seleccionada se indica mediante un vector de movimiento.

Por lo tanto, en ambos casos (predicción espacial y temporal), se calcula un residuo restando la predicción del bloque original.

En la predicción INTRA implementada mediante el módulo 903, se codifica una dirección de predicción. En la predicción temporal, se codifica, por lo menos, un vector de movimiento.

Si se selecciona la predicción Inter, se codifica información relativa al vector de movimiento y al bloque residual. Para reducir adicionalmente la tasa de bits, suponiendo que el movimiento es homogéneo, el vector de movimiento se codifica mediante la diferencia con respecto a un predictor del vector de movimiento. Se obtienen predictores de vector de movimiento de un conjunto de predictores de información de movimiento, a partir del campo de vectores de movimiento 918 mediante un módulo 917 de predicción y codificación de vectores de movimiento.

El codificador 900 comprende además un módulo de selección 906 para la selección del modo de codificación aplicando un criterio de coste de codificación, tal como un criterio de tasa-distorsión. Para reducir adicionalmente las redundancias se aplica una transformada (tal como DCT) mediante el módulo de transformada 907 al bloque residual, los datos de la transformada obtenidos son cuantificados a continuación mediante el módulo de cuantificación 908 y son sometidos a codificación entrópica mediante el módulo de codificación entrópica 909. Finalmente, el bloque residual codificado, del bloque actual que está siendo codificado, es introducido en el flujo de bits 910.

El codificador 900 lleva a cabo asimismo la descodificación de la imagen codificada, para producir una imagen de referencia de la estimación del movimiento de las imágenes subsiguientes. Esto permite que el codificador y el descodificador que reciben el flujo de bits tengan los mismos cuadros de referencia. El módulo 911 de cuantificación inversa lleva a cabo una cuantificación inversa de los datos cuantificados, seguida por una transformada inversa mediante el módulo 912 de transformada inversa. El módulo 913 de predicción intra inversa utiliza la información de predicción para determinar qué predictor utilizar para un bloque determinado, y el módulo 914 de compensación del movimiento inversa suma en realidad el residuo obtenido por el módulo 912 al área de referencia obtenida a partir del conjunto de imágenes de referencia 916.

A continuación se aplica a un filtrado posterior mediante el módulo 915 para filtrar el cuadro de píxeles reconstruido. En las realizaciones de la invención se utiliza un filtro de bucle SAO en el que los desplazamientos de compensación se suman a los valores de píxel de los píxeles reconstruidos de la imagen reconstruida.

La **figura 10** es un diagrama de flujo que muestra las etapas de un proceso de filtrado de bucle. En una etapa inicial 1001, el codificador genera la reconstrucción del cuadro completo. A continuación, en la etapa 1002 se aplica un filtro anti-efecto bloque sobre esta primera reconstrucción, para regenerar una reconstrucción sin bloques 1003. El objetivo del filtro anti-efecto bloque es eliminar artefactos de bloque generados mediante la cuantificación de residuos y la compensación del movimiento de bloques o la predicción Intra de bloques. Estos artefactos son visualmente importantes a tasas de bit bajas. El filtro anti-efecto bloque actúa para suavizar los límites de los bloques en función de las características de dos bloques contiguos. Se tiene en cuenta el modo de codificación de cada bloque, los parámetros de cuantificación utilizados para la codificación de los 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 artefactos de efecto bloque, y mejora asimismo la estimación del movimiento y la compensación del movimiento para los cuadros subsiguientes. De hecho, se eliminan las altas frecuencias de los artefactos de bloque, y por lo tanto no es necesario compensar estas frecuencias con el residuo de textura de los cuadros siguientes.

Después del filtro anti-efecto bloque, la reconstrucción sin bloques es filtrada por un filtro de bucle de desplazamiento adaptativo de muestras (SAO) en la etapa 1004, en base a una clasificación de píxeles. El cuadro resultante 1005 se puede filtrar a continuación con un filtro de bucle adaptativo (ALF) en la etapa 1006, para generar el cuadro reconstruido 1007 que será visualizado y utilizado como cuadro de referencia para los siguientes cuadros Inter.

En la etapa 1004, cada píxel del área del cuadro es clasificado en una clase de la clasificación determinada, según su valor de píxel. La clase corresponde a un intervalo determinado de valores de píxel. Se suma el mismo valor de desplazamiento de compensación al valor de píxel de todos los píxeles que tienen un valor de píxel dentro del intervalo determinado de valores de píxel.

La codificación y/o la descodificación de los parámetros SAO a intercambiar entre un servidor y un cliente, y/o almacenados local o remotamente, para descodificar píxeles para el filtrado de desplazamiento adaptativo de muestras, será explicada con más detalle a continuación haciendo referencia a cualquiera de las figuras 12 a 23.

La **figura 11** muestra un diagrama de bloques de un descodificador 1100 que puede ser utilizado para recibir datos de un codificador según una realización de la invención. El descodificador está representado mediante módulos conectados, estando cada módulo adaptado para implementar, por ejemplo en forma de instrucciones de programación a ejecutar por la CPU 811 del dispositivo 800, la correspondiente etapa de un procedimiento implementado por el descodificador 1100.

El descodificador 1100 recibe un flujo de bits 1101 que comprende unidades de codificación, cada una de las cuales se compone de una cabecera que contiene información sobre los parámetros de codificación y un cuerpo que contiene los datos de video codificados. Tal como se ha explicado con respecto a la figura 9, el video codificado es

codificado por entropía, y los índices de los predictores del vector de movimiento se codifican, para un bloque determinado, sobre un número predeterminado de bits. Los datos de video codificados recibidos, son descodificados por entropía mediante el módulo 1102. A continuación, los datos de residuo son descuantificados mediante el módulo 1103 y a continuación se aplica una transformada inversa mediante el módulo 1104 para obtener valores de píxel.

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

En el caso del modo INTRA, se determina un predictor INTRA mediante un módulo de predicción inversa intra 1105 en base al modo de predicción intra especificado en el flujo de bits.

Si el modo es *INTER*, la información de predicción del movimiento es extraída del flujo de bits para encontrar el área de referencia utilizada por el codificador. La información de predicción del movimiento se compone del índice del cuadro de referencia y del 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 1110 de descodificación del vector de movimiento.

El módulo 1110 de descodificación del vector de movimiento aplica una descodificación de vectores de movimiento a cada bloque actual codificado mediante predicción de movimiento. Una vez se ha obtenido un índice del predictor del vector de movimiento para el bloque actual, el valor real del vector de movimiento asociado con el bloque actual puede ser descodificado y utilizado para aplicar la compensación del movimiento inversa mediante el módulo 1106. La parte de la imagen de referencia indicada mediante el vector de movimiento descodificado es extraída de una imagen de referencia 1108 para aplicar la compensación del movimiento inversa 1106. Los datos de campo 1111 del vector de movimiento son actualizados con el vector de movimiento descodificado para su utilización para la predicción inversa de los subsiguientes vectores de movimiento descodificados.

Finalmente, se obtiene un bloque descodificado. Se aplica un filtrado posterior mediante el módulo 1107 de filtrado posterior, de manera similar al módulo 915 de filtrado posterior aplicado en el codificador que se ha descrito haciendo referencia a la figura 9. Finalmente, el descodificador 1000 proporciona una señal de video descodificada 1109.

La **figura 12** es un diagrama de flujo que muestra las etapas de un procedimiento para obtener parámetros SAO a partir de un flujo de bits, de acuerdo con un modo de conjunto de parámetros adaptativos (APS). Por consiguiente, la figura 12 muestra cómo se almacenan los parámetros SAO en un flujo de bits cuando se utiliza el modo de conjunto de parámetros adaptativos para asociar un solo conjunto de parámetros SAO por cada área del cuadro. De acuerdo con este modo, un conjunto de parámetros SAO es válido para un área del cuadro rectangular cuyo tamaño es un múltiplo del tamaño de la unidad máxima de codificación (LCU). Con el fin de reducir la cantidad de bits para almacenar la totalidad de los parámetros SAO de un cuadro para este modo determinado, se utiliza una técnica de codificación por longitud de serie así como un indicador de predicción a partir del área del cuadro anterior.

Según el modo APS implementado en las actuales especificaciones de HEVC, hay indicadores que indican si está o no activado el filtrado de bucle SAO para los componentes de una imagen determinada. Esto significa que, por ejemplo, cuando se considera una imagen codificada según el esquema YUV, los parámetros SAO para el componente *U* o el componente de *V* pueden estar activados o desactivados.

Dichos indicadores son leídos en una etapa inicial 1201. A continuación, en la etapa 1202, se obtiene el tamaño del área del cuadro a la que se tiene que aplicar un conjunto de parámetros SAO. Tal como se ha descrito anteriormente, el área del cuadro en la que se tiene que aplicar el filtrado de bucle SAO cuando se utiliza el modo APS corresponde a un área que es igual al tamaño de la LCU o que corresponde a un tamaño múltiplo del tamaño de la LCU.

A continuación se lanza un bucle en la etapa 1203 para tratar sucesivamente cada componente de color de la secuencia, denominado *X*. Este bucle permite leer el indicador de "unicidad" para cada componente, lo cual se lleva a cabo en la etapa 1204. A continuación, se lleva a cabo una prueba en la etapa 1205 para determinar si el indicador de "unicidad" es o no verdadero. Si el indicador de "unicidad" leído es verdadero, existe solamente un único conjunto de parámetros SAO, en relación con el actual componente *X* procesado, para toda la imagen. En ese caso, los parámetros SAO son leídos en el flujo de bits en la etapa 1206. Dicha etapa se describe en detalle haciendo referencia a la figura 14. De lo contrario, si el indicador de "unicidad" leído no es verdadero, existe más de un parámetro SAO para toda la imagen en relación con el actual componente *X* procesado. En este caso, los parámetros SAO son leídos en las etapas siguientes, tal como se describe haciendo referencia a las etapas 1208 a 1218.

A continuación, se lleva a cabo una prueba en la etapa 1207 para determinar si el actual componente *X* procesado es o no el último componente a procesar. Si el actual componente *X* procesado es el último componente a procesar,

el algoritmo se ramifica a la etapa 1203 y se repiten las etapas anteriores para el componente o componentes restantes.

5 Se debe observar que si el indicador de "unicidad" es verdadero para un componente, las etapas 1208 a 1218 no se llevan a cabo para dicho componente.

A partir de la etapa 1208, se determinan parámetros SAO para cada área del cuadro y para cada componente cuando existe más de un conjunto de parámetros SAO por componente.

10 En la etapa 1208 se lanza un bucle para tratar sucesivamente cada área del cuadro de la imagen determinada a procesar, de acuerdo con el tamaño del área del cuadro determinado en la etapa 1202. En el ejemplo dado de la figura 12, las áreas del cuadro son procesadas en orden de exploración por trama, de manera que el proceso se inicia con la primera fila de las áreas del cuadro de la imagen y continúa hasta la última fila de la imagen. Las áreas del cuadro se procesan desde el lado izquierdo al lado derecho de la imagen.

15 A continuación, se lanza otro bucle en la etapa 1209 para tratar sucesivamente cada componente denominado X. En este caso se debe señalar que si el indicador de "activación" del componente actual, leído en la etapa 1201, es falso, las etapas siguientes se ignoran en la etapa 1209 para el componente en cuestión, y se selecciona el siguiente componente a procesar. En la etapa 1210 se lee un "indicador de repetir fila". A continuación, en la etapa 1211 se realiza una prueba para determinar si el indicador es o no verdadero. Si es verdadero, los parámetros SAO de la fila anterior de un área del cuadro se utilizan para la fila actual del área del cuadro. Esto permite ahorrar muchos bits en la representación de parámetros SAO en el flujo de bits, al limitarse a señalar que la fila actual utiliza los mismos parámetros que la fila anterior.

25 De lo contrario, si el "indicador de repetir fila" leído es falso, el área del cuadro actual es codificada en el flujo de bits y el algoritmo se ramifica a la etapa 1213, en la que se realiza una prueba para determinar si se utiliza o no una técnica predictiva (utilizando codificación por longitud de serie o el valor de los parámetros SAO del área del cuadro anterior) para codificar y representar los parámetros SAO asociados con un área del cuadro anterior en la fila actual. En este caso, en la etapa 1214 los parámetros SAO del área del cuadro actual se deducen, en la fila actual, de un área del cuadro anterior. Más específicamente, se utilizan dos elementos de sintaxis: *sao_run_diff* y *sao_merge_up_flag*, que indican respectivamente si los parámetros SAO se codifican según un procedimiento de longitud de series o se deducen de los parámetros SAO del área del cuadro anterior. De lo contrario, si la respuesta es negativa en la etapa 1213, los parámetros SAO para esta área determinada del cuadro se codifican explícitamente en el flujo de bits. Por consiguiente, estos parámetros SAO son leídos del flujo de bits, en la etapa 35 1215. Esta etapa es similar a la 1206 y se describe con más detalle haciendo referencia a la figura 14.

Después de la etapa 1212, cuando se determinan los parámetros SAO para una fila completa, se lleva a cabo una prueba en la etapa 1216 para determinar si el componente actual es o no el último componente a procesar. Si no es el último componente a procesar, el algoritmo se ramifica a la etapa 1209 para procesar el siguiente o siguientes componentes. Análogamente, cuando se determinan los parámetros SAO para el área del cuadro actual, ya sea llevando a cabo la etapa 1214 o la etapa 1215, el algoritmo se ramifica a la etapa 1216 para determinar si el componente actual es o no el último componente a procesar. De nuevo, si no es el último componente a procesar, el algoritmo se ramifica a la etapa 1209 para procesar el siguiente o siguientes componentes.

45 De lo contrario, si se han procesado todos los componentes, se lleva a cabo una prueba en la etapa 1217 para determinar si ha sido o no procesada la última área del cuadro. Si no ha sido procesada la última área del cuadro, el proceso se ramifica a la etapa 1208 y se repiten las etapas anteriores. De lo contrario, el proceso finaliza.

50 En esta fase, un conjunto de parámetros SAO, con los diferentes parámetros presentados haciendo referencia a la figura 14, está asociado con cada área del cuadro y para cada componente.

La **figura 13** es un diagrama de flujo que muestra las etapas de un procedimiento para obtener parámetros SAO a partir de un flujo de bits de acuerdo con un modo LCU. Por lo tanto, la figura 13 muestra una alternativa al modo APS para codificar parámetros SAO. Más específicamente, la figura 13 muestra cómo se pueden almacenar parámetros SAO en un flujo de bits cuando se utiliza la representación del modo de unidad máxima de codificación (LCU) para asociar un solo conjunto de parámetros SAO por cada LCU. Para evitar codificar un conjunto de parámetros SAO por cada LCU, lo cual es costoso, se utiliza un esquema predictivo. En el ejemplo facilitado, este modo predictivo consiste en comprobar si la LCU situada a la izquierda de la LCU actual utiliza los mismos parámetros SAO. En caso negativo, se realiza una segunda prueba con la LCU situada sobre la LCU actual. Esta técnica predictiva permite una reducción de la cantidad de datos para representar parámetros SAO para el modo LCU. Está basada en la utilización de indicadores especializados, denominados indicadores Merge (de fusión).

65 En la etapa 1301 se lanza un bucle para tratar sucesivamente cada componente denominado X. A título ilustrativo, las imágenes siguen estando codificadas según el esquema YUV. A continuación, en la etapa 1303, un indicador denominado *sao_merge_left_flag_X* (en el que X significa Y, U o V) se lee del flujo de bits 1302 para ser procesado y descodificado. Si su valor es verdadero, los parámetros SAO de la LCU situada a la izquierda de la LCU actual son

seleccionados en la etapa 1304 para ser utilizados para procesar la LCU actual. Más específicamente, esto permite la determinación, en la etapa 1308, del tipo de filtro de bucle SAO a aplicar en la LCU actual.

De lo contrario, si el indicador leído denominado *sao_merge_left_flag_X* es falso, se lee del flujo de bits 1302 y se descodifica un indicador denominado *sao_merge_up_flag_X* (en el que *X* significa *Y*, *U* o *V*). Si su valor es verdadero, en la etapa 1306 se seleccionan los parámetros SAO de la LCU situada sobre la LCU actual para ser utilizados para procesar la LCU actual. De nuevo, esto permite la determinación, en la etapa 1308, del tipo de filtro de bucle SAO a aplicar en la LCU actual.

De lo contrario, si el indicador leído denominado *sao_merge_up_flag_X* es falso, los parámetros SAO a utilizar para el procesamiento de la LCU actual son leídos y descodificados desde el flujo de bits 1302 en la etapa 1307, describiéndose ésta última haciendo referencia a la figura 14.

En esta fase, se obtienen los parámetros SAO para procesar la LCU actual y en la etapa 1308 se determina el tipo de filtro de bucle SAO a aplicar a la LCU actual.

A continuación, en la etapa 1309, se realiza una prueba para determinar si han sido o no procesados todos los componentes *Y*, *U* y *V* para la LCU actual. Si no todos los componentes han sido procesados, se selecciona el siguiente componente, el algoritmo se ramifica a la etapa 1301 y se repiten las etapas anteriores.

A continuación, se puede procesar la siguiente LCU de manera similar.

La **figura 14** es un diagrama de flujo que muestra las etapas de un procedimiento para leer parámetros SAO en un flujo de bits. Por consiguiente, la figura 14 muestra la manera en la que los parámetros SAO se almacenan en un flujo de bits codificado.

En una etapa inicial 1402, un indicador denominado *sao_type_idx_X* (donde *X* significa *Y*, *U* o *V*) es leído del flujo de bits 1401 y descodificado. La palabra de código que representa este elemento de sintaxis puede utilizar un código de longitud fija o cualquier procedimiento de codificación aritmética.

Tal como se ha descrito anteriormente haciendo referencia a la tabla 1 del Apéndice, este elemento de sintaxis hace posible determinar el tipo de filtro de bucle SAO a aplicar al área del cuadro para procesar el componente *X*. De acuerdo con el ejemplo facilitado, el indicador *sao_type_idx_X* (o genéricamente, *sao_type_idx*), puede adoptar uno de cinco valores que dependen del filtro de bucle SAO a aplicar. Tal como se indica en la tabla 1 del Apéndice, 0 no corresponde a ningún filtro de bucle SAO, 1 a 4 corresponden a las 4 categorías de filtro de bucle SAO de desplazamiento de borde, tal como se ha descrito haciendo referencia a la tabla 2 del Apéndice, y el valor 5 corresponde al filtro de bucle SAO de desplazamiento de banda.

En la etapa 1402, después de haber leído y descodificado el *indicador sao_type_idx_X*, se realiza una prueba para determinar si el valor de este índice es o no estrictamente positivo. Si el valor de este índice es igual a cero, lo que significa que no se debe aplicar ningún filtro de bucle SAO al área del cuadro actual, la determinación de los parámetros SAO finaliza en la etapa 1407. De lo contrario, si el valor del indicador *sao_type_idx_X* es estrictamente positivo, existen parámetros SAO en el flujo de bits 1401 para el área del cuadro actual.

A continuación, en la etapa 1403 se realiza una prueba para determinar si el tipo de filtro de bucle SAO a aplicar es o no del tipo de desplazamiento de banda (*sao_type_idx_X* == 5). Si el filtro de bucle SAO a aplicar es del tipo de desplazamiento de banda, la posición de la banda SAO se lee en la etapa 1404 en el flujo de bits 1401 y se descodifica.

Después de haber leído la posición de la banda SAO en la etapa 1404, se lanza un bucle en la etapa 1405 para leer y descodificar cuatro *Offset_j* del flujo de bits 1401 (*j* varía de 1 a 4) en la etapa 1406. Estos cuatro desplazamientos corresponden a los cuatro desplazamientos relacionados con los cuatro intervalos del desplazamiento de banda SAO que se han descrito haciendo referencia a la figura 2.

Análogamente, si el valor del indicador *sao_type_idx_X* es positivo y diferente de 5, se lanza un bucle en la etapa 1405 para leer y descodificar cuatro *Offset_j* del flujo de bits 1401 (*j* varía de 1 a 4). Estos cuatro desplazamientos corresponden a los cuatro desplazamientos de los cuatro tipos de desplazamiento de borde SAO que se han descrito haciendo referencia a la tabla 2 del Apéndice.

Cuando se han leído y descodificado los cuatro desplazamientos en la etapa 1406, la lectura de los parámetros SAO finaliza en la etapa 1407.

Según un primer grupo de ejemplos y una realización de la presente invención descrita a continuación, algunos o todos los parámetros SAO son compartidos entre componentes. A este respecto, se definen procesos para obtener parámetros SAO de un componente para otro. Dichas dependencias entre componentes pueden ser definidas para las dos clases de clasificaciones SAO (desplazamientos de borde y de banda).

Tal como se ha descrito anteriormente haciendo referencia a la figura 13, en la actual implementación de HEVC para el modo LCU se utilizan dos indicadores denominados *sao_merge_left_flag_X* y *sao_merge_up_flag_X*, con el fin de obtener parámetros SAO en función de parámetros SAO procesados anteriormente. El indicador denominado *sao_merge_left_flag_X* sirve para especificar que los parámetros SAO que se tienen que utilizar para procesar el componente *X* de la LCU actual son los utilizados para procesar el mismo componente de la LCU situada a la izquierda de la LCU actual. Análogamente, el indicador denominado *sao_merge_up_flag_X* sirve para especificar que los parámetros SAO que se tienen que utilizar para procesar el componente *X* de la LCU actual son los utilizados para procesar el mismo componente de la LCU situada sobre la LCU actual.

Según un primer ejemplo concreto, estos indicadores son comunes para dos o tres componentes de la LCU actual. En otras palabras, los indicadores de los tipos *sao_merge_left_flag* y *sao_merge_up_flag* ya no se utilizan para especificar que los parámetros SAO que se tienen que utilizar para procesar un componente de la LCU actual son los utilizados para procesar el mismo componente de otra LCU, sino para especificar que los parámetros SAO que se tienen que utilizar para procesar dos o tres componentes de la LCU actual son los utilizados para procesar los mismos dos o tres componentes de otra LCU. A título ilustrativo, se puede considerar que los indicadores denominados *sao_merge_left_flag_UV* y *sao_merge_up_flag_UV* especifican que los parámetros SAO a utilizar para procesar los componentes *U* y *V* de la LCU actual son los utilizados para procesar los componentes *U* y *V* de la LCU situada a la izquierda o encima de la LCU actual, respectivamente.

Según un segundo ejemplo concreto, los indicadores se utilizan para especificar que los parámetros SAO a utilizar para procesar dos o tres componentes de la LCU actual son los utilizados para procesar un componente de otra LCU. Como consecuencia, un indicador asociado con un componente de la LCU actual puede indicar cuál de la LCU izquierda o superior tiene que ser utilizada como referencia y, en dicha LCU, cuál de los componentes tiene que ser utilizado para obtener parámetros SAO. Por lo tanto, los indicadores pueden ser los siguientes: *sao_merge_left_flag_Y*, *sao_merge_left_flag_U*, *sao_merge_left_flag_V*, *sao_merge_up_flag_Y*, *sao_merge_up_flag_U* y *sao_merge_up_flag_V*. A título ilustrativo, los parámetros SAO de un componente de la LCU actual que se refieren al indicador *sao_merge_left_flag_V* son los parámetros SAO asociados con el componente *V* de la LCU situada a la izquierda de la LCU actual.

En este segundo ejemplo, los parámetros SAO asociados con cada componente de una LCU son preferentemente iguales (para cada componente de esta LCU) cuando se refieren a los de otra LCU. A título ilustrativo, los parámetros SAO de todos los componentes de la LCU actual en la que un componente se refiere al indicador *sao_merge_left_flag_V* son los parámetros SAO asociados con el componente *V* de la LCU situada a la izquierda de la LCU actual. Por consiguiente, los parámetros SAO del componente de luma (*Y*) de la izquierda se pueden utilizar para filtrar el componente de croma (*U*) de la LCU actual.

La **figura 15** es un diagrama de flujo que muestra las etapas de un procedimiento para determinar los parámetros a utilizar en un filtro de bucle para procesar un área del cuadro actual. El ejemplo dado en la figura 15 se puede ver como una combinación del primer y el segundo ejemplos concretos descritos anteriormente.

En la etapa 1502, el indicador denominado *sao_merge_left_flag_Y* se obtiene del flujo de bits 1501 y se realiza una prueba para determinar si el valor de este indicador es o no igual a uno (es decir, si es verdadero). Si el indicador denominado *sao_merge_left_flag_Y* es verdadero, se determina en la etapa 1503 que los parámetros SAO a utilizar para filtrar los componentes *Y*, *U* y *V* de la LCU actual son iguales a los determinados anteriormente para filtrar el componente *Y* en la LCU situada a la izquierda de la LCU actual.

Por consiguiente, los valores de los indicadores *sao_type_idx_U* y *sao_type_idx_V* para la LCU actual, que definen el tipo de filtro de bucle a aplicar en los componentes *U* y *V* de la LCU actual, se consideran iguales a los del indicador *sao_type_idx_Y* de la LCU situada a la izquierda de la LCU actual. Análogamente, los otros parámetros SAO utilizados para filtrar los componentes *U* y *V* de la LCU actual, por ejemplo valores de desplazamiento y, si es necesario, posiciones de banda (*sao_band_position_U* y *sao_band_position_V*), se obtienen a partir de los del componente *Y* de la LCU situada a la izquierda de la LCU actual.

Por otra parte, si el indicador denominado *sao_merge_left_flag_Y* es falso, el indicador denominado *sao_merge_up_flag_Y* se obtiene del flujo de bits 1501 y se realiza una prueba para determinar si el valor de este indicador es o no igual a uno (es decir, si es verdadero) en la etapa 1504. Si el indicador denominado *sao_merge_up_flag_Y* es verdadero, se determina en la etapa 1505 que los parámetros SAO a utilizar para filtrar los componentes *Y*, *U* y *V* de la LCU actual son iguales a los determinados anteriormente para filtrar el componente *Y* en la LCU situada encima de la LCU actual.

Por consiguiente, los valores de los indicadores *sao_type_idx_U* y *sao_type_idx_V* para la LCU actual, que definen el tipo de filtro de bucle a aplicar en los componentes *U* y *V* de la LCU actual, se consideran iguales a los del indicador *sao_type_idx_Y* de la LCU situada encima de la LCU actual. Análogamente, los otros parámetros SAO utilizados para filtrar los componentes *U* y *V* de la LCU actual, por ejemplo valores de desplazamiento y, si es necesario, posiciones de banda (*sao_band_position_U* y *sao_band_position_V*) se obtienen de los del componente

Y de la LCU situada encima de la LCU actual.

De lo contrario, si el indicador denominado *sao_merge_up_flag_Y* es falso, los parámetros SAO para el componente Y de la LCU actual se extraen del flujo de bits 1501 en la etapa 1511, tal como se ha descrito anteriormente haciendo referencia a la figura 14. En ese caso, el valor de los indicadores *sao_type_idx_Y*, *sao_banc_position_Y* (si se requiere) y los cuatro *offsetj_Y* se extraen del flujo de bits 1501.

A continuación, si el indicador denominado *sao_merge_up_flag_Y* es falso, en la etapa 1506 el indicador denominado *sao_merge_left_flag_UV* se obtiene del flujo de bits 1501 y se realiza una prueba para determinar si el valor de este indicador es o no igual a uno (es decir, si es verdadero). Si el indicador denominado *sao_merge_left_flag_UV* es verdadero, se determina en la etapa 1507 que los parámetros SAO que se tienen que utilizar para filtrar los componentes U y V de la LCU actual son iguales a los determinados anteriormente para filtrar los componentes U y V, respectivamente, en la LCU situada a la izquierda de la LCU actual.

De lo contrario, si el indicador denominado *sao_merge_left_flag_UV* es falso, el indicador denominado *sao_merge_up_flag_UV* se obtiene del flujo de bits 1501 y se realiza una prueba para determinar si el valor de este indicador es o no igual a uno (es decir, si es verdadero) en la etapa 1508. Si el indicador denominado *sao_merge_up_flag_UV* es verdadero, se determina en la etapa 1509 que los parámetros SAO a utilizar para filtrar los componentes U y V de la LCU actual son iguales a los determinados anteriormente para filtrar los componentes U y V, respectivamente, en la LCU situada encima de la LCU actual.

De lo contrario, si el indicador denominado *sao_merge_up_flag_UV* es falso, los parámetros SAO para los componentes U y V de la LCU actual se extraen del flujo de bits 1501 en la etapa 1510, tal como se ha descrito anteriormente haciendo referencia a la figura 14. En ese caso, el valor de los indicadores *sao_type_idx_U*, *sao_type_idx_V*, *sao_band_position_U* (si es necesario), *sao_band_position_V* (si es necesario), los cuatro *offsetj_U* y los cuatro *offsetj_V* se extraen del flujo de bits 1501. En esta fase (referencia 1512), están disponibles todos los parámetros SAO para todos los componentes de la LCU actual.

A continuación, en la etapa 1513 se aplica el filtro de bucle SAO en todos los componentes de la LCU actual y el proceso de decodificación avanza a la siguiente etapa de decodificación **1514**.

También según un tercer ejemplo concreto, se pueden utilizar los indicadores comunes a todos los componentes de la LCU actual, por ejemplo los indicadores *sao_merge_left_flag_YUV* y *sao_merge_up_flag_YUV*. En este caso, si ambos indicadores son falsos, se pueden utilizar indicadores dirigidos a cada componente, denominados en general *sao_merge_left_flag_X* y *sao_merge_up_flag_X* (en los que X significa Y, U o V).

La figura 16 es un diagrama de flujo que muestra las etapas de otro procedimiento para determinar los parámetros a utilizar en un filtro de bucle para procesar un área del cuadro actual. El ejemplo proporcionado en la figura 16 está dirigido más específicamente al tercer ejemplo concreto descrito anteriormente.

En la etapa 1602, se obtiene del flujo de bits 1601 el indicador denominado *sao_merge_left_flag_YUV* y se realiza una prueba para determinar si el valor de este indicador es o no igual a uno (es decir, si es verdadero). Si el indicador denominado *sao_merge_left_flag_YUV* es verdadero, se determina en la etapa 1603 que los parámetros SAO que se tienen que utilizar para filtrar los componentes Y, U y V de la LCU actual son iguales a los determinados anteriormente para filtrar los componentes U y V, respectivamente, en la LCU situada a la izquierda de la LCU actual.

Por consiguiente, los valores de los indicadores *sao_type_idx_Y*, *sao_type_idx_U* y *sao_type_idx_V* para la LCU actual, que definen el tipo de filtro de bucle a aplicar en los componentes Y, U y V de la LCU actual, se consideran iguales a los de los indicadores *correspondientes* de la LCU situada a la izquierda de la LCU actual. Análogamente, los otros parámetros SAO utilizados para filtrar los componentes Y, U y V de la LCU actual se obtienen a partir de los del componente correspondiente de la LCU situada a la izquierda de la LCU actual.

De lo contrario, si el indicador denominado *sao_merge_left_flag_YUV* es falso, el indicador denominado *sao_merge_up_flag_YUV* se obtiene del flujo de bits 1601 y se realiza una prueba para determinar si el valor de este indicador es o no igual a uno (es decir, si es verdadero) en la etapa 1604. Si el indicador denominado *sao_merge_up_flag_YUV* es verdadero, se determina en la etapa 1605 que los parámetros SAO a utilizar para filtrar los componentes Y, U y V de la LCU actual son iguales a los determinados anteriormente para filtrar los componentes Y, U y V, respectivamente, en la LCU situada encima de la LCU actual.

De lo contrario, si el indicador denominado *sao_merge_up_flag_YUV* es falso, se decodifican los indicadores asociados con cada componente de la LCU actual para determinar si los parámetros SAO tienen o no que ser obtenidos como una función de parámetros SAO procesados anteriores.

A este respecto, en la etapa 1606 se lanza un bucle para procesar secuencialmente cada componente de la LCU actual. Para cada componente X (siendo X igual a Y, U o V), el indicador denominado *sao_merge_left_flag_X* se

extrae del flujo de bits 1601 y en la etapa 1607 se realiza una prueba para determinar si el valor de este indicador es o no igual a uno (es decir, si es verdadero). Si el indicador denominado *sao_merge_left_flag_X* es verdadero, se determina en la etapa 1608 que los parámetros SAO a utilizar para filtrar el componente *X* de la LCU actual son iguales a los determinados anteriormente para filtrar el mismo componente en la LCU situada a la izquierda de la LCU actual.

De lo contrario, si el indicador denominado *sao_merge_left_flag_X* es falso, el indicador denominado *sao_merge_up_flag_X* se obtiene del flujo de bits 1601 y se realiza una prueba para determinar si el valor de este indicador es o no igual a uno (es decir, si es verdadero) en la etapa 1609. Si el indicador denominado *sao_merge_up_flag_X* es verdadero, se determina en la etapa 1610 que los parámetros SAO a utilizar para filtrar el componente *X* de la LCU actual son iguales a los determinados anteriormente para filtrar el mismo componente en la LCU situada encima de la LCU actual.

De lo contrario, si el indicador denominado *sao_merge_up_flag_X* es falso, los parámetros SAO para el componente *X* para la LCU actual se extraen del flujo de bits 1601 en la etapa 1611.

A continuación, después de haber obtenido los parámetros SAO para el componente *X* de la LCU actual, en la etapa 1612 se realiza una prueba para determinar si el componente *X* es o no el componente *V*. Si el componente *X* no es el componente *V*, el proceso se ramifica a la etapa 1606 para procesar el siguiente componente. De lo contrario, si el componente *X* es el componente *V*, en la etapa 1613 se descodifican los parámetros SAO y están disponibles para todos los componentes de la LCU actual.

A continuación, en la etapa 1614 los componentes de la LCU actual se filtran en un filtro de bucle SAO utilizando los parámetros SAO obtenidos, y en la etapa 1615 el descodificador avanza a la siguiente etapa de descodificación.

Se debe observar que los procedimientos descritos haciendo referencia a las figuras 15 y 16 se basan en la sintaxis LCU de la descrita haciendo referencia a la figura 13. Sin embargo, los procedimientos para determinar los parámetros a utilizar en un filtro de bucle se pueden basar asimismo en la sintaxis SAO para APS descrita haciendo referencia a la figura 12. En este caso concreto, los parámetros compartidos entre componentes son los parámetros SAO, así como los indicadores denominados unicidad y repetir fila, y los elementos de sintaxis denominados *sao_run_diff* y *sao_merge_up_flag*.

Tal como se ha descrito anteriormente, en particular haciendo referencia a la tabla 1 del Apéndice, el elemento de sintaxis o indicador denominado *sao_type_idx* especifica que el componente actual de la LCU actual no es procesado cuando su valor es igual a cero. De lo contrario, éste especifica el tipo de SAO. Si el valor del indicador *sao_type_idx* es igual a 1, 2, 3 o 4, se utiliza la clasificación de desplazamiento de borde para filtrar la LCU actual, y si el valor del indicador *sao_type_idx* es igual a 5, se utiliza el desplazamiento de banda de SAO para filtrar la LCU actual.

Según realizaciones de la invención tales como las descritas anteriormente, los indicadores denominados *sao_type_idx* pueden ser compartidos entre dos o tres componentes.

En otro ejemplo concreto, los componentes pueden compartir solamente algunos valores del indicador denominado *sao_type_idx*. En otras palabras, un segundo componente comparte el valor del indicador denominado *sao_type_idx* de un primer componente solamente si este valor pertenece a un conjunto predeterminado de valores. Por lo tanto, según este ejemplo, el valor del indicador denominado *sao_type_idx* es compartido condicionalmente.

A título ilustrativo, se puede suponer que el valor del indicador denominado *sao_type_idx* asociado con el componente *Y* está compartido, para una LCU determinada, con el componente *U*. Sin embargo, se puede hacer que si este valor indica una clasificación SAO de tipo desplazamiento de borde (es decir, *sao_type_idx* = 1, 2, 3 o 4), la dirección del desplazamiento de borde se obtenga sin embargo de una parte del flujo de bits asociada con el componente *U* en lugar de utilizar la dirección de la clasificación de desplazamiento de borde proporcionada por el componente *Y*. Si el valor del indicador denominado *sao_type_idx*, asociado con el componente *Y*, no es igual a un desplazamiento de borde (*sao_type_idx* = 0 o 5), no se obtiene ninguna información de la dirección del flujo de bits. En otras palabras, si el tipo de SAO para el componente *Y* es cero o cinco, este parámetro SAO (relacionado con el tipo) será utilizado por los componentes *U* y *V*. Sin embargo, si el tipo de SAO es del tipo de desplazamiento de borde (es decir, un valor de uno a cuatro) para el componente *Y*, se utilizarán tipos de SAO especializados para los componentes *U* y *V*.

Tal como se ha descrito anteriormente, los parámetros SAO (asociados con el filtro de bucle SAO que utiliza una clasificación de píxeles del tipo de desplazamiento de borde) comprenden cuatro desplazamientos que están relacionados con las cuatro clases, de las cinco posibles. La primera clase (clase "0") no corresponde a ningún filtrado SAO. Se debe recordar que la dirección de la clasificación de desplazamiento de borde se señala en el tipo de SAO.

Según los ejemplos descritos más adelante, estos cuatro desplazamientos pueden ser compartidos entre dos o tres

componentes cuando se selecciona un filtro de desplazamiento de borde para la LCU actual, independientemente de la dirección de clasificación de desplazamiento de borde.

Según un primer ejemplo concreto, se utilizan los mismos desplazamientos para dos o tres componentes. Por consiguiente, solamente es necesario obtener cuatro desplazamientos de un flujo de bits, en lugar de ocho o doce si dos o tres componentes comparten respectivamente los mismos cuatro desplazamientos, para el filtrado de píxeles con tres componentes.

Naturalmente, dichos desplazamientos comunes tienen que ser calculados en el lado del codificador para permitir el procesamiento de dos o tres componentes con los mismos desplazamientos. Tal como se describe a continuación, esta elección de la codificación es la misma que la utilizada para el filtrado de desplazamiento de banda.

En otro ejemplo concreto, los desplazamientos Ozj asociados con el componente Z se obtienen a partir de desplazamientos Oxj de otro componente X , de acuerdo con un factor de ponderación βj , como sigue:

$$Ozj = \beta j Oxj$$

El factor de ponderación βj puede ser el mismo independientemente de cuál sea el valor de j ($\beta = \beta j$), o no. El valor de β o los valores de βj pueden ser predeterminados y constantes, o pueden ser transmitidos en cabeceras concretas, tales como la cabecera de secuencia o una cabecera de segmento, o incluso transmitir LCU a LCU (o, más generalmente, área del cuadro a área del cuadro).

Cuando los valores varían de una LCU a otra, se pueden calcular dinámicamente en los lados del codificador y del decodificador.

Los valores de los factores de ponderación pueden ser calculados según la fórmula siguiente:

$$\beta_j = \frac{\sum_{i=0}^N \max(|c_i^Z - c_{in1}^Z|, |c_i^Z - c_{in2}^Z|)}{\sum_{i=0}^N \max(|c_i^X - c_{in1}^X|, |c_i^X - c_{in2}^X|)}$$

en que c_i^Z es el valor del componente Z del píxel i de clase j , y en que c_{in1}^Z y c_{in2}^Z son los valores del componente Z de los píxeles contiguos al píxel i de clase j tal como se ha descrito anteriormente, en particular haciendo referencia a la tabla 2 del Apéndice.

Por consiguiente, el valor del factor de ponderación βj es la relación entre la diferencia media (máxima) del componente Z de los píxeles clasificados y sus píxeles contiguos, y la diferencia media (máxima) del componente X de los píxeles clasificados y sus píxeles contiguos. Se pueden utilizar otras relaciones o fórmulas para calcular los valores del factor de ponderación βj (o de β).

En otro ejemplo concreto más, el valor de los desplazamientos codificados en un flujo de bits puede depender de los valores del factor de ponderación βj (o β), como quiera que se determinen. En dicho ejemplo, si el valor del factor de ponderación β o βj es menor que uno, en el flujo de bits se codifica el valor del desplazamiento Ozj en lugar del valor del desplazamiento Oxj .

Por supuesto, en la fórmula $Ozj = \beta j Oxj$, $Ozj < Oxj$ si $\beta j < 1$. Como consecuencia, el valor del desplazamiento Ozj requiere menos bits para ser codificado en el flujo de bits que el valor de Oxj . Dado que el decodificador conoce o puede determinar los valores del factor de ponderación βj , puede decodificar el valor de Ozj a partir del flujo de bits y determinar el valor de Oxj según la fórmula siguiente:

$$Ozj = (1 / \beta j) \times Oxj$$

También en un ejemplo concreto, los valores absolutos de los desplazamientos Ozj asociados con un segundo componente Z son iguales a los valores absolutos de los desplazamientos $Ox(5-j)$ asociados con un primer componente X . Se debe observar en este caso que, según este ejemplo concreto, solamente se utiliza el valor absoluto dado que el signo del desplazamiento para la clase de desplazamiento de borde está predeterminado para reducir artefactos.

La **figura 17** muestra dicho ejemplo concreto, según el cual los valores de los desplazamientos asociados con un componente corresponden a los valores especulares de los desplazamientos asociados con un segundo componente. Tal como se muestra, los desplazamientos para las 4 clases de componente Z se obtienen de los desplazamientos del componente X , considerando el orden inverso de los desplazamientos. Se debe recordar que los signos aplicados a los desplazamientos se deducen de la clasificación.

En otro ejemplo concreto más, los valores especulares de los desplazamientos asociados con un primer componente utilizados como desplazamientos asociados con un segundo componente se pueden ponderar con factores de ponderación β_j . Según este ejemplo concreto, los valores absolutos de los desplazamientos Oz_j asociados con un segundo componente Z son iguales a los valores absolutos ponderados de los desplazamientos $Ox(5-j)$ asociados con un primer componente X , de acuerdo con la fórmula siguiente:

$$Oz_j = \beta_j Ox(5-j)$$

Tal como se ha descrito anteriormente, el filtro de desplazamiento de banda SAO requiere dos tipos de parámetros. El primer tipo está dirigido a una posición (`sao_band_position`) y el segundo tipo está dirigido a los cuatro desplazamientos relacionados con las cuatro clases de la banda que se define a partir de la posición determinada. Según los ejemplos descritos anteriormente, los dos tipos de parámetros asociados con componentes de un área del cuadro se pueden obtener independientemente de los correspondientes parámetros asociados con los componentes de otra área del cuadro ya procesada. Sin embargo, los ejemplos que se refieren a la posición (`sao_band_position`) y los que se refieren a los desplazamientos se pueden combinar fácilmente. Además, tal como se ha descrito anteriormente haciendo referencia a los ejemplos de desplazamiento de borde, estos ejemplos o combinaciones de ejemplos que se refieren a posiciones y desplazamientos para el filtrado de desplazamiento de banda se aplican en función de los valores de indicadores Merge (de fusión) o de la sintaxis SAO para APS que se ha descrito anteriormente.

Se debe observar que dado que los parámetros SAO asociados con un componente pueden ser utilizados para procesar dos o tres componentes, la siguiente descripción se centra en la utilización de parámetros SAO asociados con un componente para procesar dos componentes, a título ilustrativo.

La **figura 18** muestra cómo se pueden obtener valores de desplazamientos asociados con un segundo componente Z , a partir de valores de desplazamientos asociados con un primer componente X cuando se aplica un filtro de bucle SAO utilizando una clasificación de desplazamiento de banda.

La referencia 1801 muestra la utilización de parámetros SAO, en particular de desplazamientos asociados con componentes para aplicar un filtro de bucle SAO utilizando una clasificación de desplazamiento de banda, según las actuales especificaciones de HEVC. Los desplazamientos asociados con cada componente están codificados en el flujo de bits utilizado para transmitir las imágenes codificadas. A título ilustrativo, el flujo de bits comprende los desplazamientos $Ox1$ a $Ox4$, denominados genéricamente Oxj , asociados con un primer componente X y los desplazamientos $Oz1$ a $Oz4$, denominados genéricamente Ozj , asociados con un primer componente Z . Tal como se muestra mediante la referencia 1801, los desplazamientos Oxj corresponden a una primera banda que comprende cuatro clases consecutivas denominadas $BO X$ y los desplazamientos Ozj corresponden a una segunda banda, que comprende también cuatro clases consecutivas, denominadas $BO Z$.

La referencia 1802 muestra un primer ejemplo concreto en el que los desplazamientos Ozj son iguales a los desplazamientos Oxj . Por lo tanto, los desplazamientos Ozj no están codificados en el flujo de bits. Estos se pueden recuperar utilizando los correspondientes indicadores Merge (de fusión) que se han descrito anteriormente haciendo referencia al filtrado de desplazamiento de borde. Análogamente, se puede utilizar un factor de ponderación β o β_j , de tal modo que los desplazamientos Ozj se pueden determinar en función de los desplazamientos Oxj ($Ozj = \beta_j Oxj$). De nuevo, el valor o valores de los factores de ponderación β o β_j pueden ser predeterminados o representados a diferentes niveles en el flujo de bits (a nivel de secuencia, a nivel de cuadro, a nivel de segmento o a nivel de LCU). En este contexto, el valor o valores del factor de ponderación β o β_j puede ser negativo (en caso de desplazamiento de borde, el signo de los desplazamientos se determina en función de la clase). En particular, el factor de ponderación puede ser igual a menos uno ($\beta = -1$).

En otro ejemplo concreto más, el valor o valores de los factores de ponderación β o β_j se puede determinar dinámicamente. A título ilustrativo, el valor o valores de los factores de ponderación β o β_j se puede determinar en función de la posición de la banda (`sao_band_position`). Por ejemplo, si la clase j de la banda Z y la clase j de la banda X no están en los mismos subintervalos desde el punto de vista del centro del intervalo, (representado en línea gruesa en la figura 18) $\beta = -1$, de lo contrario $\beta = 1$. Se considera en este caso que todo el intervalo está dividido en dos subintervalos, siendo el primero el intervalo de los valores de píxel desde 0 hasta 127, y siendo el segundo el intervalo de los valores de píxel desde 128 hasta 256 (cuando se utiliza una profundidad de 8 bits para el valor de píxel).

Según otro ejemplo, los valores del factor de ponderación β_j se determinan en función de la relación entre la distancia de la clase actual del componente Z (o el conjunto de clases para el valor del factor de ponderación β) hasta el centro del intervalo, y la distancia entre la clase actual del componente X (o el conjunto de clases para el valor del factor de ponderación β) y el centro del intervalo, lo que se puede expresar como sigue:

$$\beta_j = \frac{sao_band_position_Z_j - 16}{sao_band_position_X_j - 16}$$

en que $sao_band_position_Zj - 16$ es la distancia entre el componente Z actual y el centro del intervalo, y $sao_band_position_Xj - 16$ es la distancia entre el componente X actual y el centro del intervalo. En otro ejemplo, se podría considerar la relación inversa. En relación con el desplazamiento de borde, se pueden utilizar otras relaciones y fórmulas para determinar el valor o valores del factor de ponderación β o β_j . Además, tal como se ha descrito anteriormente, se pueden introducir desplazamientos Oxj o desplazamientos Ozj en el flujo de bits en función del valor o valores de los factores de ponderación β o β_j (como quiera que se determinen) con el fin de reducir el número de bits necesarios para codificar desplazamientos Oxj o desplazamientos Ozj .

La referencia 1803 muestra cómo los valores especulares de los desplazamientos asociados con un primer componente pueden ser utilizados como valores de desplazamientos asociados con un segundo componente para el filtrado de desplazamiento de banda. Tal como se ha descrito anteriormente haciendo referencia al filtrado de desplazamiento de borde, las relaciones entre los desplazamientos Ozj y Oxj se pueden expresar mediante la fórmula siguiente:

$$Ozj = Ox(5-j)$$

De nuevo, este ejemplo concreto se puede combinar con la utilización de un factor de ponderación, según la fórmula $Ozj = \beta_j Ox(5-j)$. En particular, β_j es igual a -1.

Una derivación específica de los desplazamientos para el filtrado de desplazamiento de banda es el aumento o la disminución del tamaño de las clases que permite la modificación de la clasificación del desplazamiento de banda de los píxeles. La referencia 1604 muestra un ejemplo de dicha adaptación de la clasificación. Según el ejemplo mostrado, el número de píxeles por clase está dividido por dos para el componente Z en comparación con la clasificación de banda para el componente X . Por lo tanto, la clasificación de píxeles para el componente Z , según el filtrado de desplazamiento de banda, requiere la comprobación de seis bits. En otras palabras, es necesario comprobar los seis bits más significativos para clasificar un píxel. Por lo tanto, se puede considerar que el intervalo completo de valores de píxel está dividido en 64 intervalos para el componente Z .

Esta relación puede ser fija para la implementación o puede ser señalizada a nivel de secuencia, cuadro, segmento o LCU. Esto se puede determinar dinámicamente teniendo en cuenta la relación entre el tamaño del intervalo de píxeles para el componente X y el tamaño del intervalo de los píxeles para el componente Z a nivel de LCU (o de secuencia, cuadro o segmento). Por consiguiente, el tamaño de las clases para el componente Z se puede determinar como sigue:

$$size(class_Z) = \frac{max(P_{zi}) - min(P_{zi})}{max(P_{xi}) - min(P_{xi})} size(class_X)$$

en todos los casos $i = 0$ a N , en que P_{zi} son los píxeles del componente Z y P_{xi} son los píxeles del componente X . Para simplificar la clasificación de bandas, el número de valores de píxel por clase es igual a una potencia de dos (permitiendo la comprobación de bits para los bits más significativos)

Este ejemplo concreto puede ser combinado con todos los demás ejemplos particulares descritos para la obtención de desplazamientos del desplazamiento de banda, tal como se muestra mediante la referencia 1805, en la que este ejemplo se combina con la utilización de desplazamientos de un valor especular asociados con el componente X .

En otro ejemplo concreto que es aplicable al filtrado SAO tanto de borde como de banda, los desplazamientos para un componente se pueden obtener a partir de dos o tres componentes.

Todos los ejemplos descritos aplicables al filtrado SAO tanto de borde como de banda, que se refieren a la utilización de desplazamientos de un componente para determinar (u obtener) los desplazamientos de uno o dos componentes diferentes, presentan ventajas, en particular en relación con el tamaño de la memoria tampón requerida cuando dicha obtención se utiliza sistemáticamente (para todas las LCU), dado que no es necesario almacenar los desplazamientos del segundo componente y/o del tercer componente.

En otro ejemplo concreto (para filtrado SAO tanto de borde como de banda), se puede utilizar la obtención de desplazamientos entre componentes para predecir los desplazamientos de un segundo componente, en función de los desplazamientos de un primer componente. (La descripción se dedica a compartir desplazamientos entre componentes). En este ejemplo, se han codificado en el flujo de bits los residuos de los desplazamientos del componente Z en lugar de codificar directamente el valor de los desplazamientos del componente Z . En el lado del descodificador, los desplazamientos del componente Z se obtienen sumando los desplazamientos residuales a los desplazamientos del componente X .

En otro ejemplo concreto aplicable al filtrado de desplazamientos SAO tanto de borde como de banda, la obtención

de los desplazamientos entre componentes se puede utilizar como un contexto para la codificación aritmética por contexto de los desplazamientos asociados con un segundo componente, en función de los desplazamientos asociados con un primer componente.

5 En otro ejemplo concreto más, aplicable al filtrado SAO tanto de borde como de banda, la obtención de desplazamientos entre dos componentes puede ser señalizada en el flujo de bits y puede competir con el modo convencional mediante el cual los desplazamientos están codificados en el flujo de bits para todos los componentes.

10 Tal como se ha descrito anteriormente, una posición de banda (*sao_band_position*) es la posición de la primera clase de una banda en los intervalos completos de los valores de píxel. Considerando una comprobación estándar de cinco bits, el valor del parámetro *sao_band_position* pertenece al intervalo [0; 32]. **Este** parámetro puede ser compartido entre dos o tres componentes. A título ilustrativo, la descripción se centra en la obtención de este parámetro en relación con un componente *Z* a partir de su valor asociado con un componente *X*.

15 La **figura 19** muestra cómo se puede obtener un valor de un parámetro *sao_band_position* asociado con un segundo componente *Z*, a partir del valor del parámetro *sao_band_position* asociado con el primer componente *X* cuando se aplica un filtro de bucle SAO utilizando una clasificación de desplazamiento de banda.

20 La referencia 1901 muestra un ejemplo de dos valores de *sao_band_position* que son transmitidos en el flujo de bits según las actuales especificaciones de HEVC.

En un ejemplo concreto mostrado mediante la referencia 1902, se utiliza un valor de *sao_band_position* común tanto para el componente *Z* como para el componente *X*.

25 En ejemplos concretos, se compara la posición central de una banda con el centro de todo el intervalo de píxeles. El valor resultante es igual a *sao_band_position* - 14 (correspondiendo 14 al centro del intervalo (16) menos el centro de la banda (-2)).

30 Por consiguiente, en un ejemplo concreto mostrado mediante la referencia 1903, la *sao_band_position_Z* asociada con el componente *Z* se define como que es simétrica a *sao_band_position_X* asociada con el componente *X* con respecto al centro de todo el intervalo de píxeles. Por lo tanto, se considera que (*sao_band_position_Z* - 14) es igual a -(*sao_band_position_X* - 14). Dado que *sao_band_position_X* es igual a seis en este ejemplo, *sao_band_position_Z* es por lo tanto igual a veintidós (-(6-14)+14= 22).

35 También en un ejemplo concreto mostrado mediante la referencia 1904, se utiliza un factor de ponderación β para determinar el valor de *sao_band_position_Z* asociado con el componente *Z* en función del valor de *sao_band_position_X* asociado con el componente *X* y de la posición del centro de todo el intervalo de píxeles. Esto se basa, igual que en el ejemplo anterior, en la posición relativa entre el centro de todo el intervalo de píxeles y el valor de *sao_band_position* de cada componente *X*, y *Z*. Según este ejemplo, la relación a utilizar puede ser la siguiente

$$sao_band_position_Z - 14 = \beta (sao_band_position_X - 14)$$

es decir

$$sao_band_position_Z = \beta (sao_band_position_X - 14) + 14$$

En el ejemplo facilitado en la figura 19, $\beta = -1/2$. En este caso, *sao_band_position_X* es igual a 6, y por consiguiente *sao_band_position_Z* es igual a dieciocho (-1/2(6-14)+14=18).

El valor del factor de ponderación β puede estar predeterminado para la implementación o ser transmitido a cualquier nivel (de secuencia, cuadro, segmento o LCU).

También en otro ejemplo concreto aplicable al filtrado SAO tanto de borde como de banda, la obtención del parámetro *sao_band_position* entre dos componentes puede ser señalizada en el flujo de bits y puede competir con el modo convencional en el que el valor del parámetro *sao_band_position* está codificado en el flujo de bits para todos los componentes.

La selección de la codificación de parámetros SAO cuando se comparten parámetros SAO entre dos o tres componentes de color, se refiere preferentemente a la eficiencia de la codificación. En otras palabras, al seleccionar parámetros SAO compartidos se tiene en cuenta ventajosamente la eficiencia de la codificación de estos parámetros y los valores asociados. Se debe recordar en este caso que la selección del mejor desplazamiento en términos del coste de la tasa-distorsión es la misma para el filtrado de desplazamiento tanto de borde como de banda.

Las selecciones de codificación de desplazamientos cuando estos desplazamientos son compartidos entre dos componentes Z y X, cualquiera que sea tipo de SAO (borde o banda) utilizado, se describen a continuación antes de la selección del centro de la banda.

Tal como se ha descrito haciendo referencia a la figura 3 que muestra el proceso de codificación para determinar un desplazamiento óptimo en términos de distorsión para un componente, los valores de $SumXj$ y $SumNbPixXj$ relacionados con el componente X y los valores de $SumZj$ y $SumNbPixZj$ relacionados con el componente Z son conocidos para todas las clases j. $SumXj$ y $SumZj$ son las sumas de las diferencias entre los valores de píxel en la clase j y sus valores originales para los componentes X y Z, respectivamente. $SumNbPixXj$ y $SumNbPixZj$ son el número de píxeles en la clase j para los componentes X y Z, respectivamente. Se supone que los valores de estos cuatro parámetros han sido calculados ya para todas las clases.

En el ejemplo concreto según el cual el objetivo es conseguir que $Ozj = Oxj$, el desplazamiento óptimo $Ooptj$ se puede determinar mediante la fórmula siguiente:

$$Ooptj = Ozj = Oxj = (SumXj + SumZj) / (SumNbPixXj + SumNbPixZj)$$

A continuación, tal como se ha descrito anteriormente para procesar independientemente cada componente, el codificador determina el mejor desplazamiento Oxj en términos del criterio de tasa-distorsión.

Esto comprende, en el ejemplo concreto según el cual $Ozj = Oxj$, minimizar el coste de tasa-distorsión como sigue:

$$J(Oxj, Ozj) = SumNbPixXj \times (Oxj \times Oxj) - SumXj \times Oxj \times 2 + SumNbPixZ(j) \times (Oxj \times Oxj) - SumZ(j) \times Oxj \times 2 + \lambda R(Oxj)$$

lo que se puede simplificar según la expresión siguiente:

$$J(Oxj, Ozj) = (SumNbPixXj + SumNbPixZj) \times (Oxj \times Oxj) - (SumXj + SumZj) \times Oxj \times 2 + \lambda R(Oxj)$$

En el ejemplo concreto según el cual los valores especulares están dirigidos a determinar desplazamientos ($Ozj = -Ox(5-j)$) en el caso de desplazamiento de borde, el desplazamiento óptimo se puede definir mediante la fórmula siguiente:

$$Ooptj = Oz(5-j) = Oxj = (SumXj - SumZ(5-j)) / (SumNbPixXj + SumNbPixZ(5-j))$$

A continuación, el codificador determina el mejor desplazamiento Oxj en términos del criterio de tasa-distorsión: Esto comprende, de acuerdo con el ejemplo concreto según el cual $Oz(5-j) = -Oxj$, minimizar el coste de tasa-distorsión como sigue:

$$J(Oxj, Ozj) = SumNbPixXj \times (Oxj \times Oxj) - SumXj \times Oxj \times 2 + SumNbPixZ(5-j) \times (Oz(5-j) \times Oz(5-j)) - SumZ(5-j) \times Oz(5-j) \times 2 + \lambda R(Oxj)$$

o

$$J(Oxj, Ozj) = SumNbPixXj \times (Oxj \times Oxj) - SumXj \times Oxj \times 2 + SumNbPixZ(5-j) \times (Oxj \times Oxj) + SumZ(5-j) \times Oxj \times 2 + \lambda R(Oxj)$$

En el ejemplo concreto según el cual $Ozj = -Oxj$, el desplazamiento óptimo $Ooptj$ se determina mediante la fórmula siguiente:

$$Ooptj = (SumXj - SumZj) / (SumNbPixXj + SumNbPixZj)$$

A continuación, el codificador determina el mejor desplazamiento Oxj en términos del criterio de tasa-distorsión: Esto comprende, de acuerdo con el ejemplo concreto según el cual $Ozj = -Oxj$, minimizar el coste de tasa-distorsión tal

como sigue:

$$J(Oxj, Ozi) = SumNbPixXj \times (Oxj \times Oxj) - SumXj \times Oxj \times 2 + SumNbPixZ(j) \times (Ox(j) \times Ox(j)) + SumZ(j) \times Ox(j) \times 2 + \lambda R(Oxj),$$

5 que se puede simplificar mediante la expresión siguiente:

$$J(Oxj, Ozi) = (SumNbPixXj + SumNbPixZj) \times (Oxj \times Oxj) - (SumXj - SumZj) \times Oxj \times 2 + \lambda R(Oxj)$$

10 En el ejemplo concreto según el cual $Ozi = \beta_j Oxj$, el codificador determina independientemente un valor $OoptXj$ óptimo para el componente X y un valor $OoptZj$ óptimo para el componente Z. A continuación, el codificador determina el mejor desplazamiento Oxj en términos del criterio de tasa-distorsión: Esto comprende, de acuerdo con el ejemplo concreto $Ozi = \beta_j Oxj$, minimizar el coste de tasa-distorsión tal como se ha descrito haciendo referencia a la figura 4.

15 La **figura 20** es un diagrama de flujo que muestra las etapas de un procedimiento para determinar un desplazamiento óptimo a utilizar por dos o tres componentes, de acuerdo con un criterio de tasa-distorsión. El proceso mostrado en la figura 20 está basado en el que se ha descrito haciendo referencia a la figura 4. Las etapas 2001, 2004, 2005 y 2006 son similares a las etapas 401, 404, 405 y 406, respectivamente.

20 La etapa 2002 se diferencia de la etapa 402 en que el bucle lanzado en la etapa 2002 hace que el desplazamiento Oxj varíe de $sign(OoptXj) \times \max(f \times OoptXj, OoptZj)$ a cero, en que f es igual a $1/\beta_j$ si $\beta_j < 1$, y a β_j en caso contrario.

En el ejemplo generalizado en el que $Ozi = \beta_j Oxj$, el criterio de tasa-distorsión se calcula según la fórmula siguiente:

$$J(Oxj, Ozi) = SumNbPixXj \times (Oxj \times Oxj) - SumXj \times Oxj \times 2 + SumNbPixZ(j) \times (\beta_j \times Oxj \times \beta_j \times Oxj) - SumZ(j) \times \beta_j \times Oxj \times 2 + \lambda R(Oxj)$$

esto se puede escribir asimismo como sigue:

$$J(Oxj, Ozi) = (SumNbPixXj + SumNbPixZ(j) \times (\beta_j) \times (\beta_j)) \times (Oxj \times Oxj) - (SumXj + SumZ(j) \times \beta_j) \times Oxj \times 2 + \lambda R(Oxj)$$

En la etapa 2003, se considera a título ilustrativo el caso particular en el que $\beta_j = -1$.

La selección de un valor de `sao_band_position` compartido entre dos componentes X y Z puede ser determinada según la siguiente relación:

$$sao_band_position_Z = \beta (sao_band_position_X - 14) + 14$$

40 La **figura 21** es un diagrama de flujo que muestra las etapas de un procedimiento para determinar una posición de banda SAO compartida entre dos componentes. Se basa en el proceso descrito haciendo referencia a la figura 5. El proceso mostrado en la figura 21 está dirigido a determinar el valor de `sao_band_position_Z` asociado con el componente Z a partir del valor de `sao_band_position_X` asociado con el componente X.

45 La diferencia principal entre el proceso mostrado en la figura 21 y el mostrado en la figura 5 se refiere al módulo 2105, en el que se suman los valores de tasa-distorsión de las clases asociadas con el componente Z y de las clases asociadas con el componente X, según la fórmula siguiente:

$$J'i = J'i + JZj + JXjx$$

50 en la que JZj es el valor de tasa-distorsión para la clase j del componente Z y $JXjx$ es el valor de tasa-distorsión para la clase j del componente X. Además, para seleccionar conjuntamente el `sao_band_position_Z` con el `sao_band_position_X`, el índice jx se obtiene según la fórmula siguiente:

$$sao_band_position_Z = \beta (sao_band_position_X - 14) + 14$$

55 Por consiguiente, jx es igual a $\beta (j - 14) + 14$.

De acuerdo con un segundo grupo de ejemplos descritos a continuación que evitan la clasificación de componentes con el fin de reducir la complejidad de la decodificación, los píxeles en una clase SAO son determinados en función de un único componente. Las posiciones de los píxeles se utilizan a continuación para determinar el componente o componentes que se tienen que filtrar. Por consiguiente, si un componente de píxel debe ser filtrado, según este componente, se filtra otro componente o los otros dos componentes de este píxel.

La **figura 22** es un diagrama de flujo que muestra las etapas de un procedimiento para decodificar un flujo de bits, según un ejemplo concreto. El proceso mostrado en esta figura está basado en el mostrado en la figura 6. Está dirigido a utilizar la clasificación del componente *X* para filtrar el componente *Z*.

Una etapa inicial 2201 comprende determinar parámetros SAO según el proceso descrito haciendo referencia a las figuras 12, 13 y 14. Sin embargo, según el algoritmo mostrado en la figura 22, en la etapa 2202 se determina solamente el valor del parámetro *sao_type_idx* y, si es igual a cinco, el valor del parámetro *sao_band_position* para el componente *X* a partir del flujo de bits (no representado). Por supuesto, cuando estos parámetros están relacionados con la clasificación, no precisan clasificar el componente *Z* que está filtrado según la clasificación del componente *X*.

A continuación, en la etapa 2203 se obtienen del flujo de bits los cuatro desplazamientos asociados con el componente *X*. De manera similar, en la etapa 2204 se obtienen del flujo de bits los cuatro desplazamientos asociados con el componente *Z*. Se debe observar que estos desplazamientos pueden ser determinados en función de otros desplazamientos asociados con otros componentes y/o con otra LCU, tal como se ha descrito anteriormente.

En la etapa 2205, la variable de cómputo *i* se ajusta al valor cero. A continuación, en la etapa 2207 se obtiene el primer píxel P_{xi} del área del cuadro del componente *X* (referencia 2206), que comprende *N* píxeles. De manera similar, en la etapa 2209 se obtiene el primer píxel P_{zi} del área del cuadro del componente *Z* (referencia 2208). A título ilustrativo, se admite en este caso que el área del cuadro del componente *Z* contiene asimismo *N* píxeles (los componentes *X* y *Z* tienen la misma cantidad de píxeles por área del cuadro).

A continuación, en la etapa 2210, el componente *X* del primer píxel P_{xi} se clasifica según la clasificación de desplazamiento de borde o la clasificación de desplazamiento de banda gracias al valor del parámetro *sao_type_idx_X* (y, si es necesario, al valor del parámetro *sao_band_position_X*), tal como se ha descrito anteriormente haciendo referencia a la tabla 2 del Apéndice y a la figura 2, respectivamente.

En la etapa 2211, se realiza una prueba para determinar si es necesario o no filtrar el componente *X* del píxel P_{xi} (es decir, si P_{xi} pertenece a una clase SAO). Si P_{xi} pertenece a una clase SAO, se identifica el número de clase *j* relacionado y se obtiene el correspondiente valor de desplazamiento en la etapa 2212 a partir de la tabla de desplazamientos asociada con el componente *X*. A continuación, en la etapa 2213, el componente *X* del píxel P_{xi} es filtrado añadiendo a su valor el desplazamiento obtenido. El componente *X* filtrado del píxel P_{xi} es denominado P'_{xi} (referencia 2214). Simultánea o secuencialmente, el valor de desplazamiento asociado con el componente *Z* y el correspondiente número de clase *j* se obtienen en la etapa 2215 a partir de la tabla de desplazamientos asociada con el componente *Z*. A continuación, en la etapa 2216, el componente *Z* del píxel P_{zi} es filtrado añadiendo a su valor el desplazamiento obtenido. El componente *Z* filtrado del píxel P_{zi} es denominado P'_{zi} (referencia 2217).

A continuación, en la etapa 2218, los componentes filtrados P'_{xi} y P'_{zj} se ponen en el componente *X* y el componente *Z* del área del cuadro filtrada, denominados 2219 y 2220, respectivamente. En la etapa 2218, los píxeles filtrados P'_{xi} y P'_{zj} sustituyen respectivamente los píxeles P_{xi} y P_{zj} de los cuadros procesados denominados 2219 y 2220.

Si P_{xi} no pertenece a una clase SAO, los componentes P_{xi} y P_{zj} en el componente *X* y el componente *Z* de las áreas del cuadro filtradas permanecen sin cambios en las etapas 2221, 2222 y 2218.

A continuación, en la etapa 2223, la variable de cómputo *i* se aumenta en una unidad para aplicar el proceso de filtrado de manera similar a los píxeles siguientes del área del cuadro actual.

En la etapa 2224 se determina si han sido o no procesados todos los *N* píxeles del área del cuadro actual ($i \geq N$). En caso afirmativo, los componentes filtrados *X* y *Z* del área del cuadro se reconstruyen y se pueden añadir al cuadro reconstruido SAO.

La figura 22 muestra la utilización de la clasificación SAO de un componente *X* para filtrar de manera similar otro componente *Z* cuando ambos componentes tienen el mismo número de píxeles para el área del cuadro actual.

Sin embargo, los componentes de un área del cuadro no siempre tienen el mismo número de píxeles. Dependiendo del número relativo de píxeles para el componente *X*, denominado *N_x*, en relación con el número de píxeles para el componente *Z*, denominado *N_z*, se pueden considerar dos casos. Dado que la diferencia entre los números de

píxeles de dos componentes diferentes en un área del cuadro es habitualmente un múltiplo de dos, generalmente no es necesario tener en cuenta relaciones no enteras entre números de componentes de píxeles.

Considerando el caso según el cual el número de píxeles del componente Z es mayor que el número de píxeles del componente X ($N_x < N_z$), el algoritmo descrito haciendo referencia a la figura 22 se puede adaptar fácilmente considerando que P_{zi} es un grupo de N_z/N_x píxeles. Por consiguiente, cuando se añade un desplazamiento $Offset_{Zi}$ a un grupo de píxeles P_{zi} , este desplazamiento se aplica todos los píxeles del grupo de píxeles P_{zi} . Además, en un ejemplo concreto, se puede considerar que el desplazamiento $Offset_{Zi}$ es un grupo de desplazamientos del mismo tamaño que el grupo de píxeles P_{zi} .

Considerando el caso según el cual el número de píxeles del componente X es mayor que el número de píxeles del componente Z ($N_x > N_z$), se pueden considerar varios ejemplos. En este caso, las etapas 2216 y 2218 descritas haciendo referencia a la figura 22 tienen que tener en cuenta la relación entre el número de píxeles del componente X y el número de píxeles del componente Z . En la figura 22, P_{zi} se cambia por $P_{z(i/N_x)}$.

En un primer ejemplo concreto, en desplazamiento $Offset_{Zi}$ se añade al grupo $P_{z(i/N_x)}$ solamente si P_{xi} ha sido filtrado, por lo menos, una vez.

Según otro ejemplo concreto, la etapa 2218 de la figura 22 tiene en cuenta el número de veces que P_{xi} ha sido filtrado. Si es mayor que $2 \times N_x/N_z$, entonces $Offset_{Zi}$ se añade a $P_{z(i/N_x)}$.

También según un ejemplo concreto, la etapa 2218 recibe N_x/N_z veces $P_{z(i/N_x)}$ o $P_{z(i/N_x)}$, y calcula el promedio para producir el valor de píxel establecido en el área del cuadro filtrada para el componente Z (2220).

La **figura 23** muestra la utilización de la clasificación SAO de un componente X para filtrar de manera similar otro componente Z .

A título ilustrativo, los píxeles en gris del bloque 2301 representan píxeles filtrados para un filtro de bucle SAO del tipo i de la clase j actual para el componente X , según la actual implementación de HEVC. Análogamente, los píxeles negros del bloque 2302 representan píxeles filtrados para un filtro de bucle SAO del tipo i de la clase j actual para el componente Z , según las especificaciones de HEVC actuales.

Por lo tanto, tal como se muestra en la figura 23, los píxeles en gris en el componente X y los píxeles en negro en el componente Z no siempre están en la misma posición considerando las actuales especificaciones de HEVC.

También a título ilustrativo, los píxeles en gris del bloque del 2303 representan píxeles filtrados para un filtro de bucle SAO del tipo i de la clase j actual para el componente X , según la actual implementación de HEVC. Sin embargo, los píxeles en negro del bloque 2304 representan píxeles filtrados para un filtro de bucle SAO para el componente Z , según la clasificación del componente X (bloque 2303). Por consiguiente, los píxeles en gris y los píxeles en negro están situados en las mismas posiciones. Esto tiene como resultado la utilización de solamente una clasificación para ambos componentes.

Cabe señalar que los ejemplos del primer y el segundo grupos de ejemplos descritos anteriormente se pueden combinar para optimizar la eficiencia de la codificación y la complejidad de la decodificación.

Aunque la presente invención ha sido descrita en lo que antecede haciendo referencia a realizaciones específicas, la presente invención no se limita a las realizaciones específicas, y para un experto en la materia resultarán evidentes modificaciones que caen dentro del alcance de la presente invención. En particular, aunque la descripción anterior se refiere a componentes YUV, a título ilustrativo, se pueden utilizar otros esquemas tales como RGB (componentes rojo, verde y azul).

Por ejemplo, aunque las realizaciones anteriores se han descrito en relación con píxeles de una imagen y sus correspondientes valores de píxel, se apreciará que, dentro del contexto de la invención, se puede considerar un grupo de píxeles junto con el correspondiente valor de píxel del grupo. De este modo, una muestra puede corresponder a uno o varios píxeles de una imagen.

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

APÉNDICETabla 1

<u>sao_type_idx</u>	<u>Tipo de SAO</u>	<u>Significado del tipo de SAO</u>
0	ninguno	No se aplica ningún filtrado SAO en el área del cuadro
1	<i>borde</i>	1D 0 grados
2	<i>borde</i>	1D 90 grados
3	<i>borde</i>	1D 135 grados
4	<i>borde</i>	1D 45 grados
5	<i>banda</i>	Desplazamiento de banda con posición de banda

Tabla 2

<u>Categoría (J)</u>	<u>Condiciones</u>	<u>Desplazamiento</u>
1	$C < Cn_1$ y $C < Cn_2$	+O ₁
2	$(C < Cn_1$ y $C == Cn_2)$ o $(C < Cn_2$ y $C == Cn_1)$	+O ₂
3	$(C > Cn_1$ y $C == Cn_2)$ o $(C > Cn_2$ y $C == Cn_1)$	-O ₃
4	$C > Cn_1$ y $C > Cn_2$	-O ₄
N.D.	<i>Ninguno de los anteriores</i>	N.D.

REIVINDICACIONES

1. Procedimiento de decodificación de, por lo menos, una parte de una imagen que comprende una serie de muestras, comprendiendo cada muestra un componente de luma y un primer y un segundo componentes de croma, comprendiendo la imagen codificada, por lo menos, un parámetro de filtrado, comprendiendo el procedimiento
5 decodificar (1103-1106) el primer componente de croma y el segundo componente de croma de, por lo menos, una muestra codificada para proporcionar un primer componente de croma y un segundo componente de croma de, por lo menos, una muestra reconstruida;
10 decodificar un conjunto de parámetros de filtrado para llevar a cabo un filtrado de desplazamiento adaptativo de muestras sobre la parte de imagen, comprendiendo el conjunto descodificado un parámetro de tipo desplazamiento adaptativo de muestras, que indica si se utiliza un filtrado de bucle de desplazamiento adaptativo de muestras del tipo borde, del tipo banda o ninguno para la, por lo menos, una muestra reconstruida, y comprendiendo asimismo,
15 por lo menos, un parámetro adicional de filtrado; y
llevar a cabo un filtrado de bucle de desplazamiento adaptativo de muestras sobre la, por lo menos, una muestra reconstruida en un filtro de bucle (1107);
20 **caracterizado por que** la parte de imagen es una unidad máxima de codificación de la imagen y el conjunto descodificado está asociado con la unidad máxima de codificación, y por que el filtrado comprende utilizar dicho parámetro de tipo desplazamiento adaptativo de muestras como un parámetro de filtrado común para filtrar tanto el primer componente de croma como el segundo componente de croma de la, por lo menos, una muestra reconstruida, y utilizar dicho parámetro adicional de filtrado como un parámetro de filtrado especializado, para filtrar
25 uno individual del primer y del segundo componentes de croma.
2. Procedimiento, según la reivindicación 1, en el que el primer y el segundo componentes de croma son componentes de croma U y V, respectivamente.
3. Procedimiento, según la reivindicación 1, en el que la dirección de filtrado de bucle de desplazamiento adaptativo de muestras de tipo borde es otro parámetro de filtrado común.
4. Procedimiento, según la reivindicación 1, 2 o 3, en el que los desplazamientos son parámetros de filtrado especializados.
35
5. Procedimiento, según cualquier reivindicación anterior, que comprende llevar a cabo un filtrado de desplazamiento adaptativo de muestras sobre la primera y la segunda partes de imagen correspondientes, respectivamente, a la primera y segunda unidades máximas de codificación y, cuando se descodifica un indicador para la segunda unidad máxima de codificación, utilizar el conjunto descodificado de parámetros de filtrado asociados con la primera unidad máxima de codificación para llevar a cabo asimismo un filtrado de desplazamiento adaptativo de muestras sobre la segunda unidad máxima de codificación.
40
6. Procedimiento, según la reivindicación 5, en el que la primera unidad máxima de codificación está inmediatamente a la izquierda de la segunda unidad máxima de codificación, y dicho indicador es un indicador izquierdo.
45
7. Procedimiento, según la reivindicación 5, en el que la primera unidad máxima de codificación está inmediatamente por encima de la segunda unidad máxima de codificación y dicho indicador es un indicador superior.
8. Producto de programa informático para un aparato programable, comprendiendo el producto de programa informático una secuencia de instrucciones para implementar cada etapa de un procedimiento, según cualquiera de las reivindicaciones 1 a 7, cuando es cargado en el aparato programable y ejecutado por este.
50
9. Medio de almacenamiento legible por ordenador, que almacena instrucciones de un programa informático para implementar cada etapa de un procedimiento, según cualquiera de las reivindicaciones 1 a 7.
55
10. Dispositivo de decodificación para decodificar, por lo menos, una parte de una imagen que comprende una serie de muestras, comprendiendo cada muestra un componente de luma y, por lo menos, un primer y un segundo componentes de croma, comprendiendo la imagen codificada, por lo menos, un parámetro de filtrado, comprendiendo el dispositivo de decodificación:
60 medios para decodificar (1103-1106) el primer componente de croma y el segundo componente de croma de, por lo menos, una muestra codificada para proporcionar un primer componente de croma y un segundo componente de croma de, por lo menos, una muestra reconstruida;
65 medios para decodificar un conjunto de parámetros de filtrado para llevar a cabo un filtrado de desplazamiento adaptativo de muestras sobre la parte de imagen, comprendiendo el conjunto descodificado un parámetro de tipo

desplazamiento adaptativo de muestras, que indica si se utiliza un filtrado de bucle de desplazamiento adaptativo de muestras de tipo borde, de tipo banda o ninguno, para la, por lo menos, una muestra reconstruida, y comprendiendo asimismo, por lo menos, un parámetro adicional de filtrado; y

- 5 medios de filtrado para llevar a cabo un filtrado de desplazamiento adaptativo de muestras (1004) sobre la, por lo menos, una muestra reconstruida en un filtro de bucle (1107);

10 **caracterizado por que** la parte de imagen es una unidad máxima de codificación de la imagen y el conjunto descodificado está asociado con la unidad máxima de codificación, y **por que** los medios de filtrado pueden funcionar para utilizar dicho parámetro de tipo desplazamiento adaptativo de muestras como un parámetro de filtrado común para filtrar tanto el primer componente de croma como el segundo componente de croma de la, por lo menos, una muestra reconstruida, y pueden funcionar además para utilizar dicho parámetro adicional de filtrado como un parámetro de filtrado especializado, para filtrar uno individual del primer y el segundo componentes de croma.

- 15 11. Dispositivo, según la reivindicación 10, en el que el primer y el segundo componentes de croma son componentes de croma U y V, respectivamente.

20 12. Dispositivo, según la reivindicación 10 o 11, en el que la dirección de filtrado de bucle de desplazamiento adaptativo de muestras del tipo borde es otro parámetro de filtrado común.

13. Dispositivo, según la reivindicación 10, 11 o 12, en el que los desplazamientos son parámetros de filtrado especializados.

25 14. Dispositivo, según cualquiera de las reivindicaciones 10 a 13, en el que los medios de filtrado pueden funcionar para llevar a cabo filtrados de desplazamiento adaptativo de muestras sobre la primera y la segunda partes de imagen correspondientes, respectivamente, a una primera y una segunda unidades máximas de codificación y, cuando se descodifica un indicador para la segunda parte de imagen, para utilizar el conjunto descodificado de parámetros de filtrado asociados con la primera unidad máxima de codificación para llevar a cabo asimismo un filtrado de desplazamiento adaptativo de muestras sobre la segunda unidad máxima de codificación.

- 30 15. Dispositivo, según la reivindicación 14, en el que la primera unidad máxima de codificación está inmediatamente a la izquierda de la segunda unidad máxima de codificación, y dicho indicador es un indicador izquierdo.

35 16. Dispositivo, según la reivindicación 14, en el que la primera unidad máxima de codificación está inmediatamente por encima de la segunda unidad máxima de codificación y dicho indicador es un indicador superior.

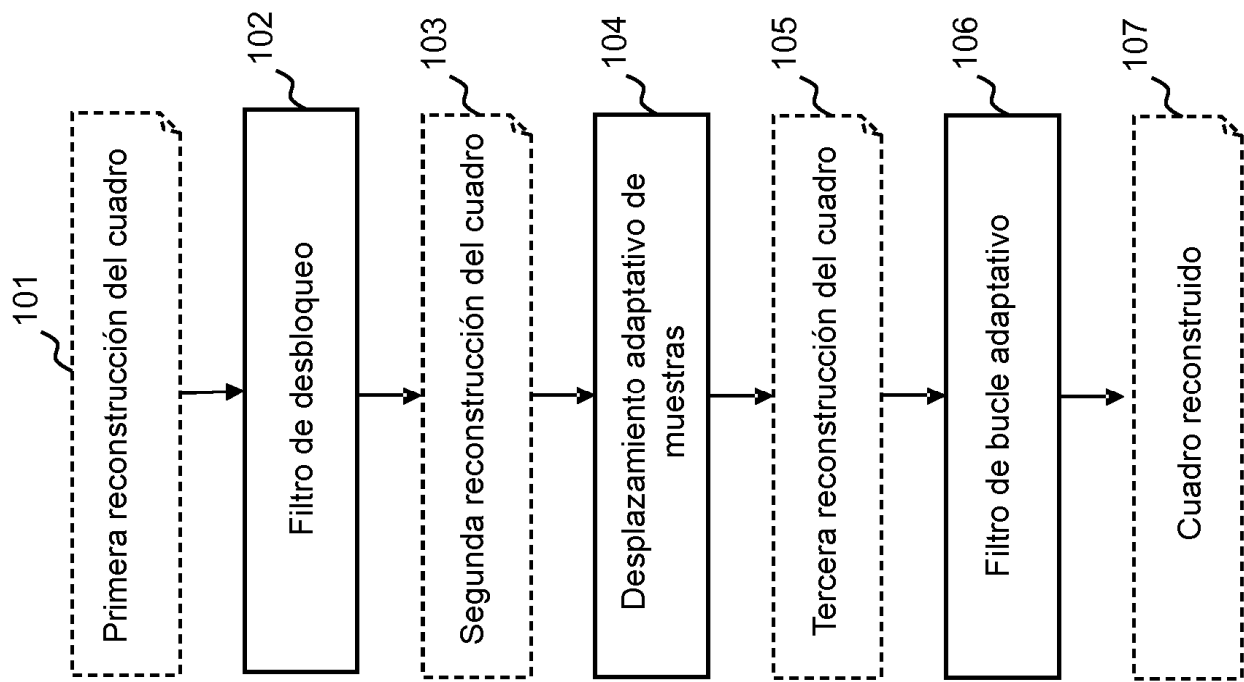
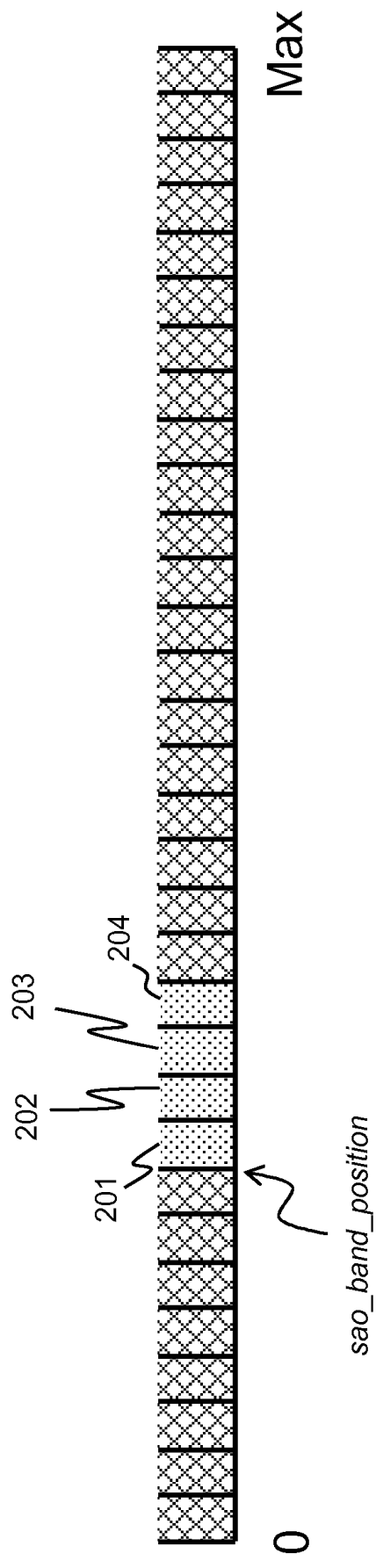


Figura 1

Figura 2



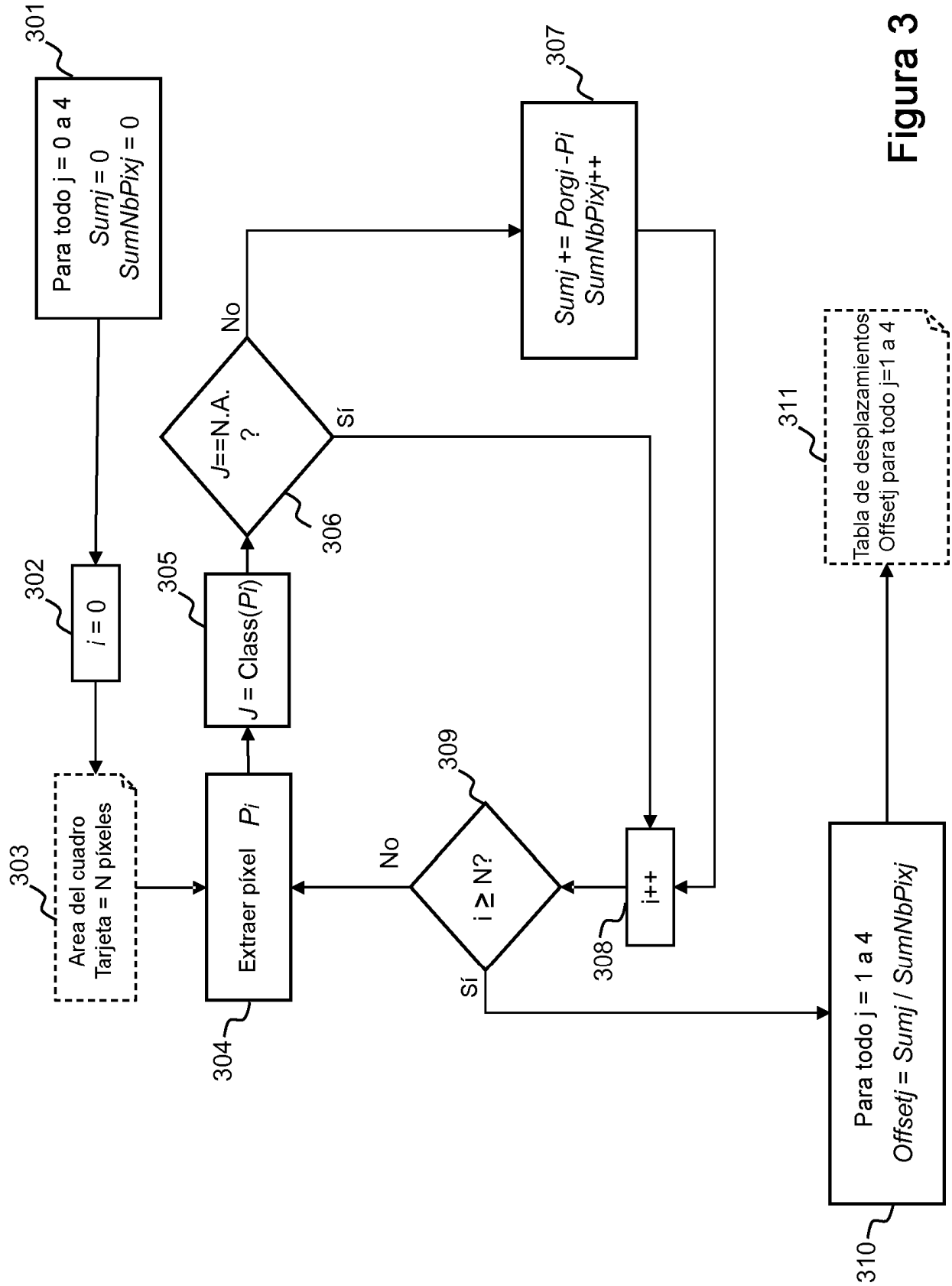


Figura 3

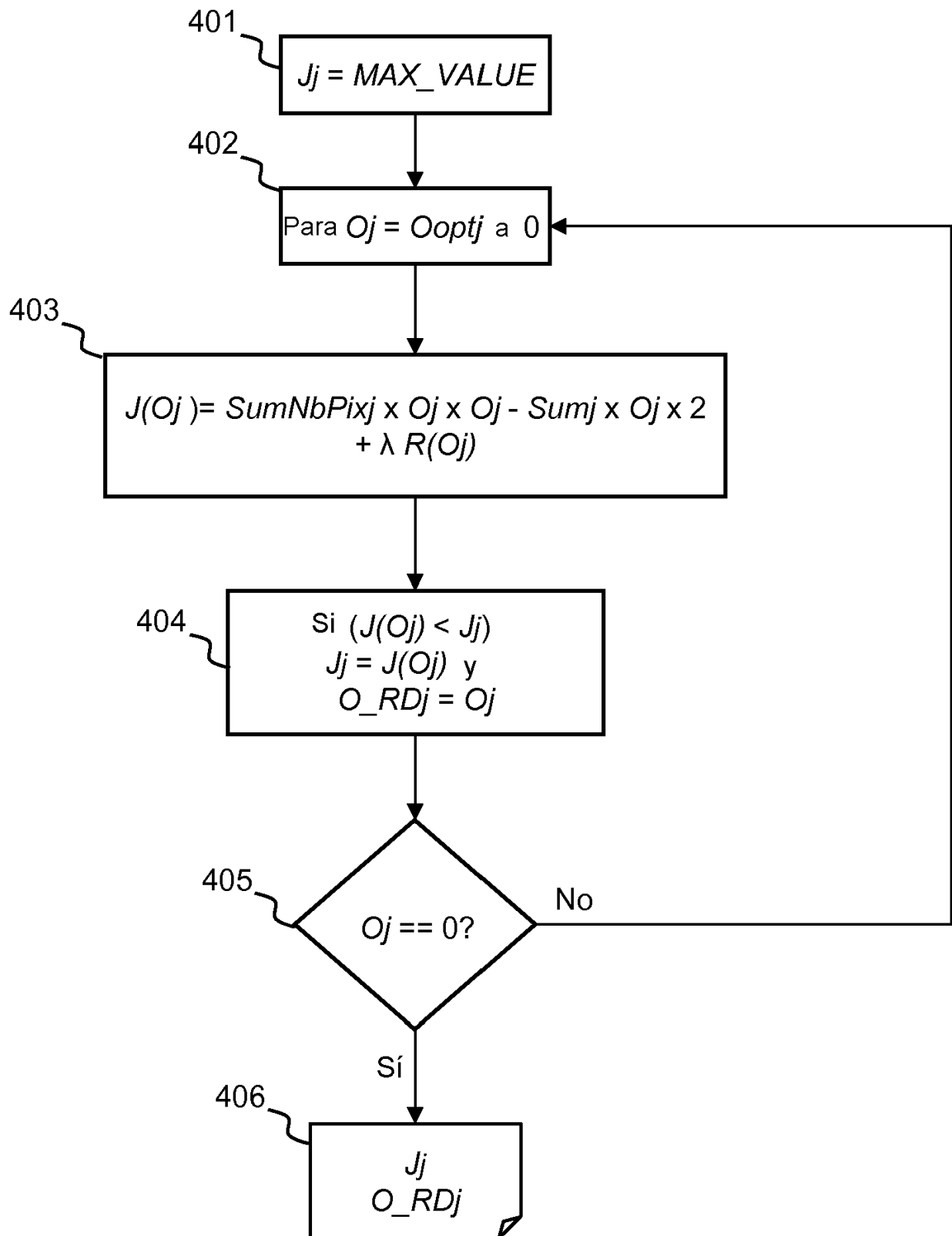


Figura 4

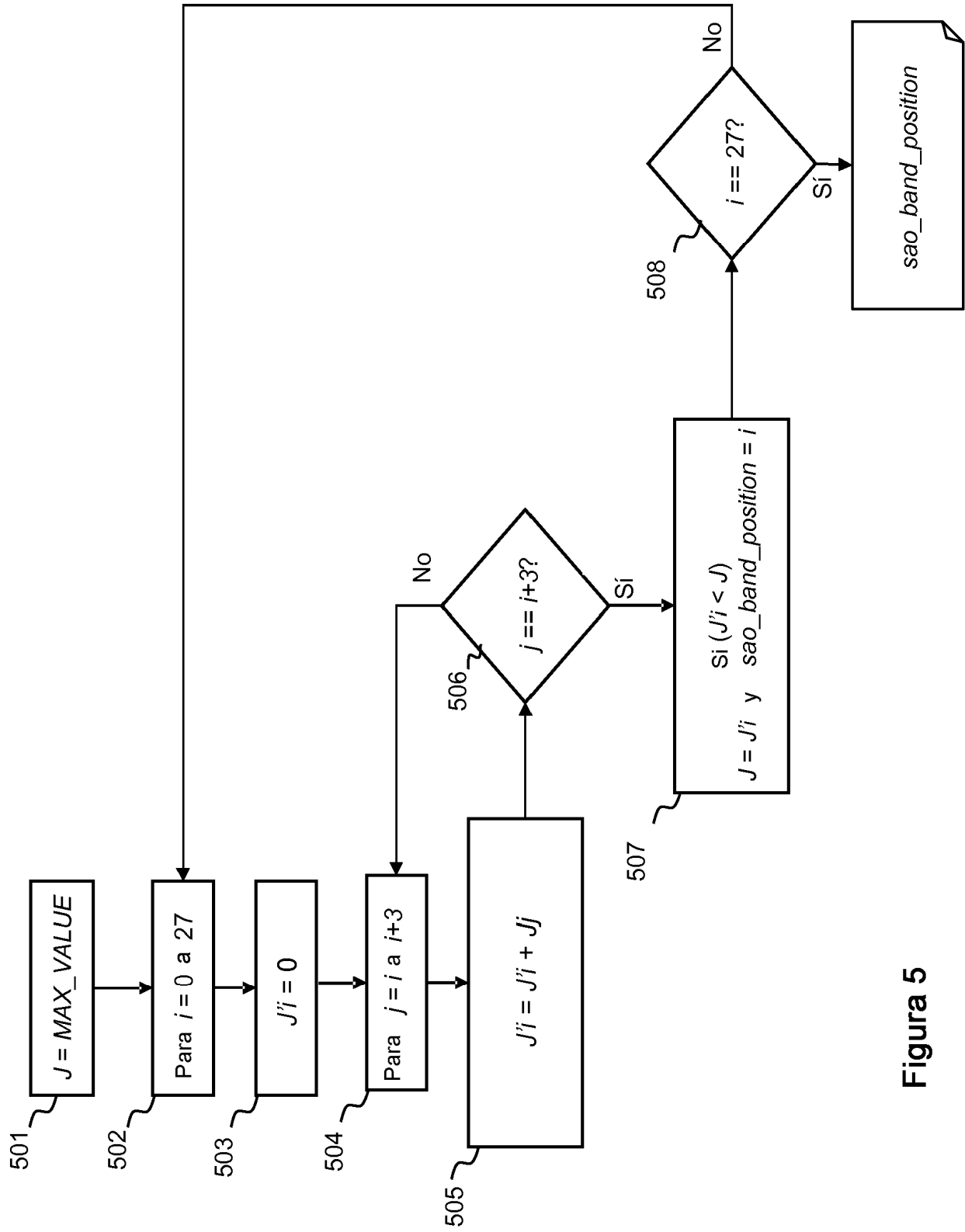


Figura 5

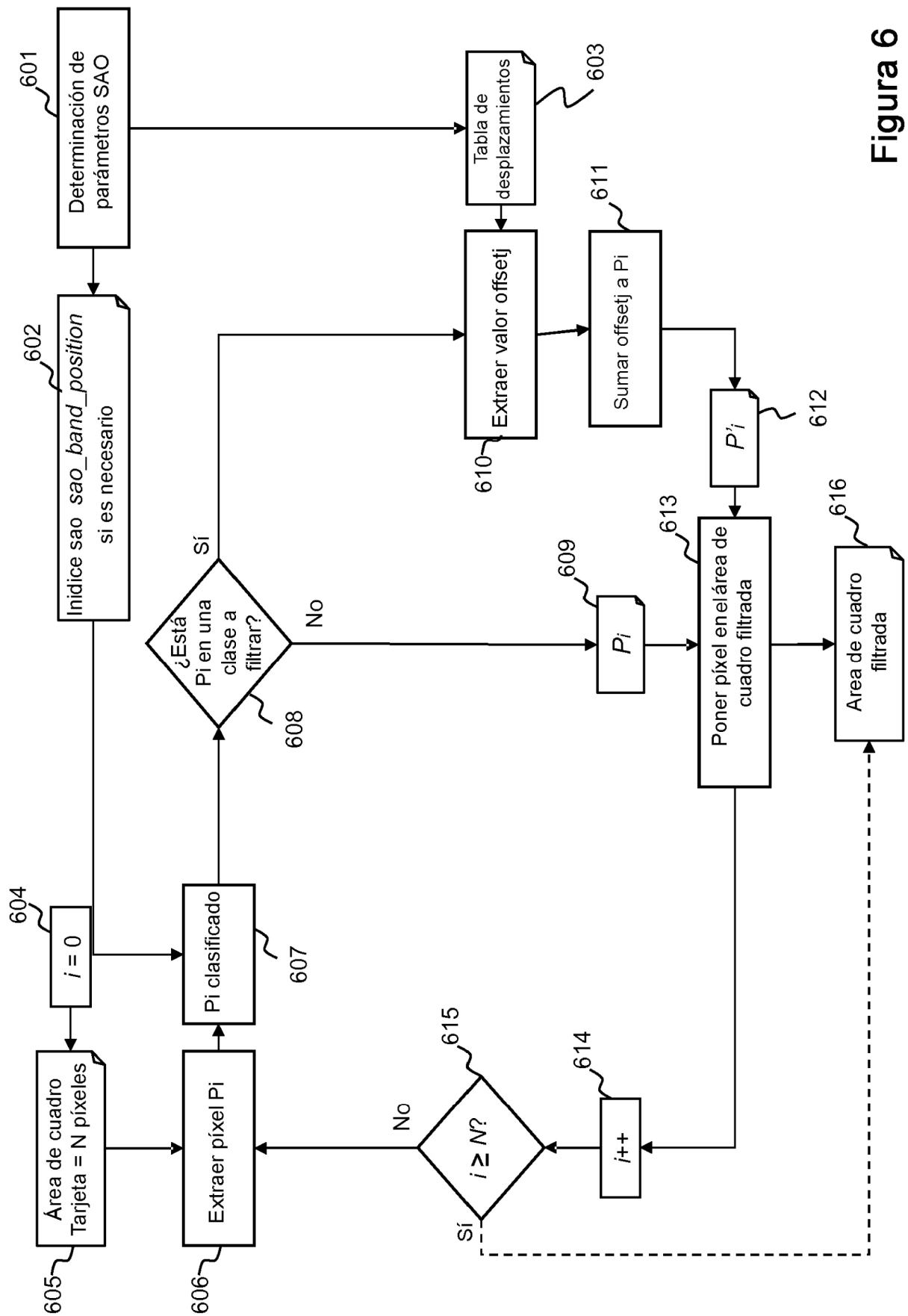


Figura 6

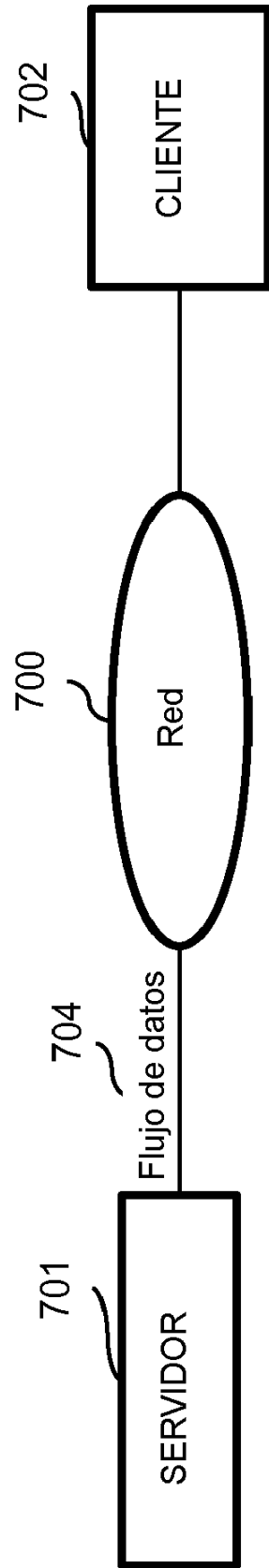


Figura 7

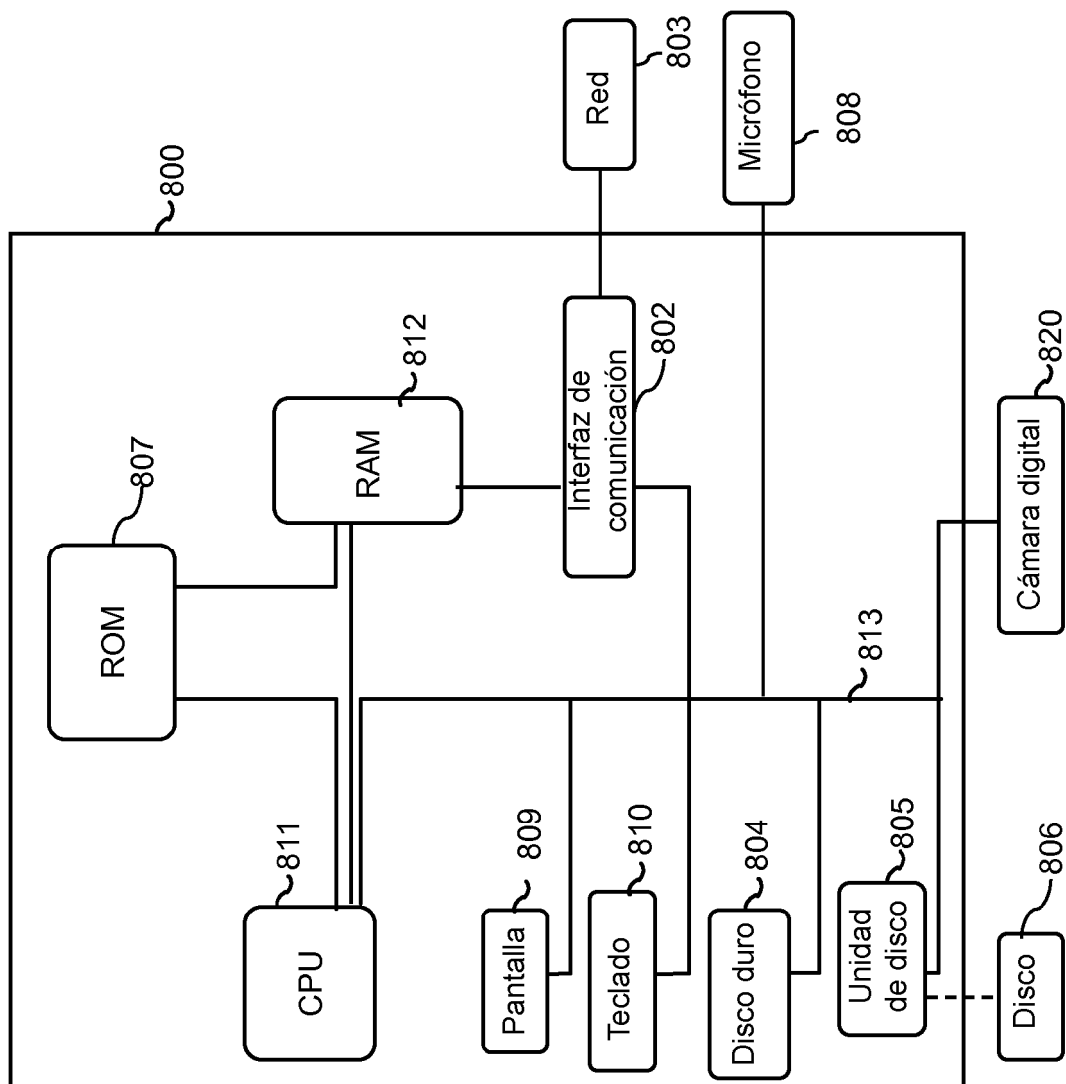


Figura 8

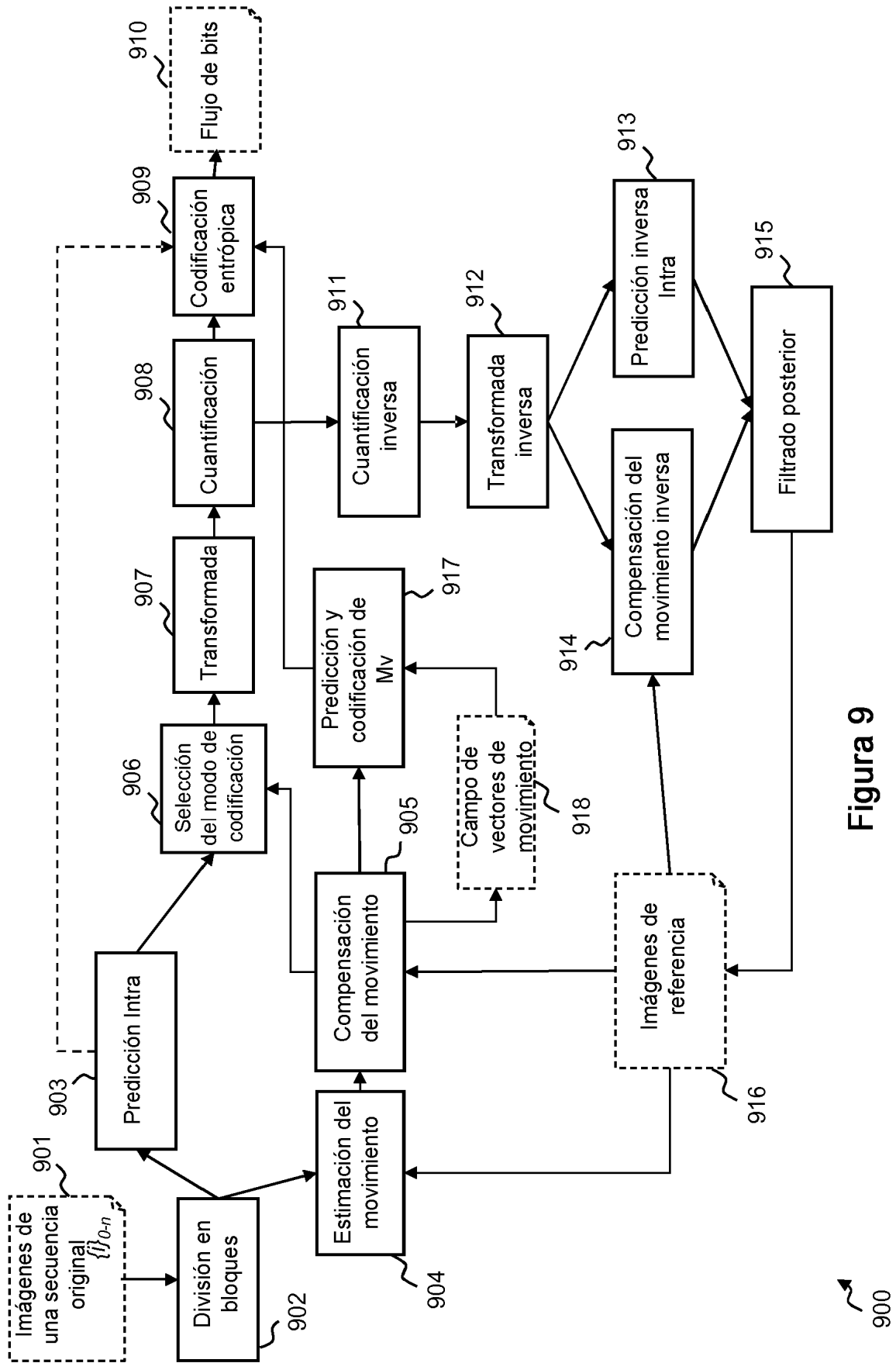
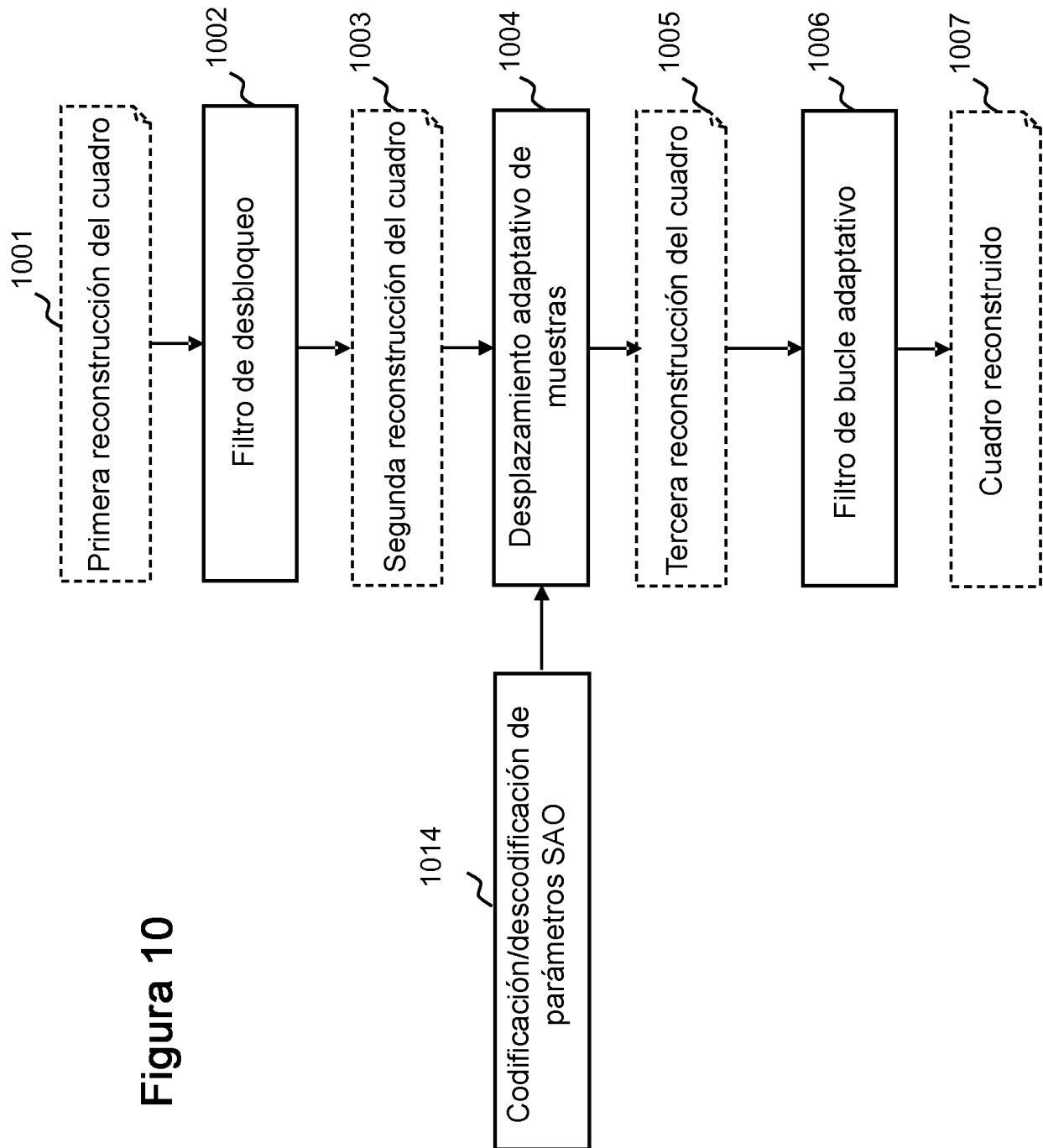


Figura 9



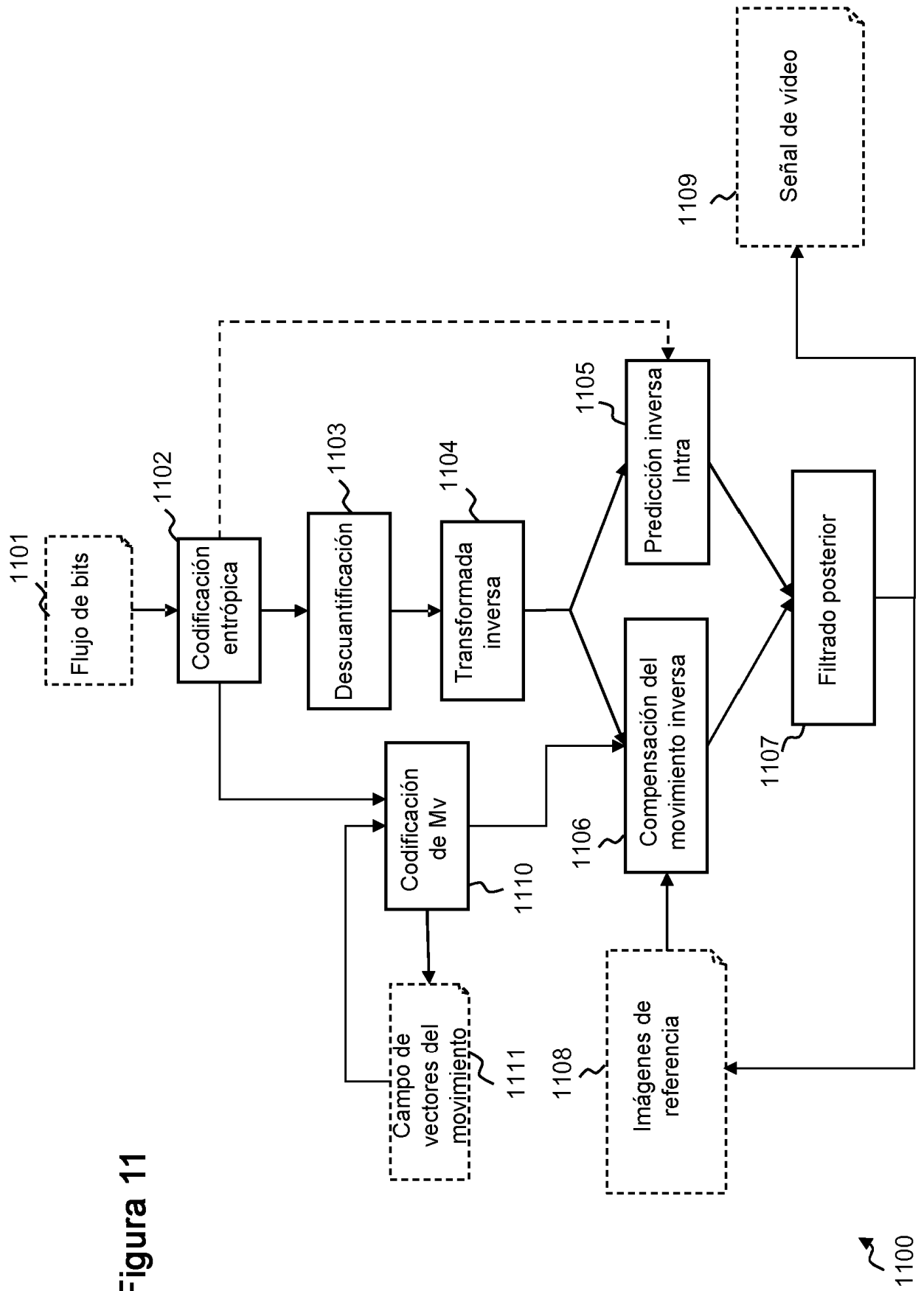


Figura 11

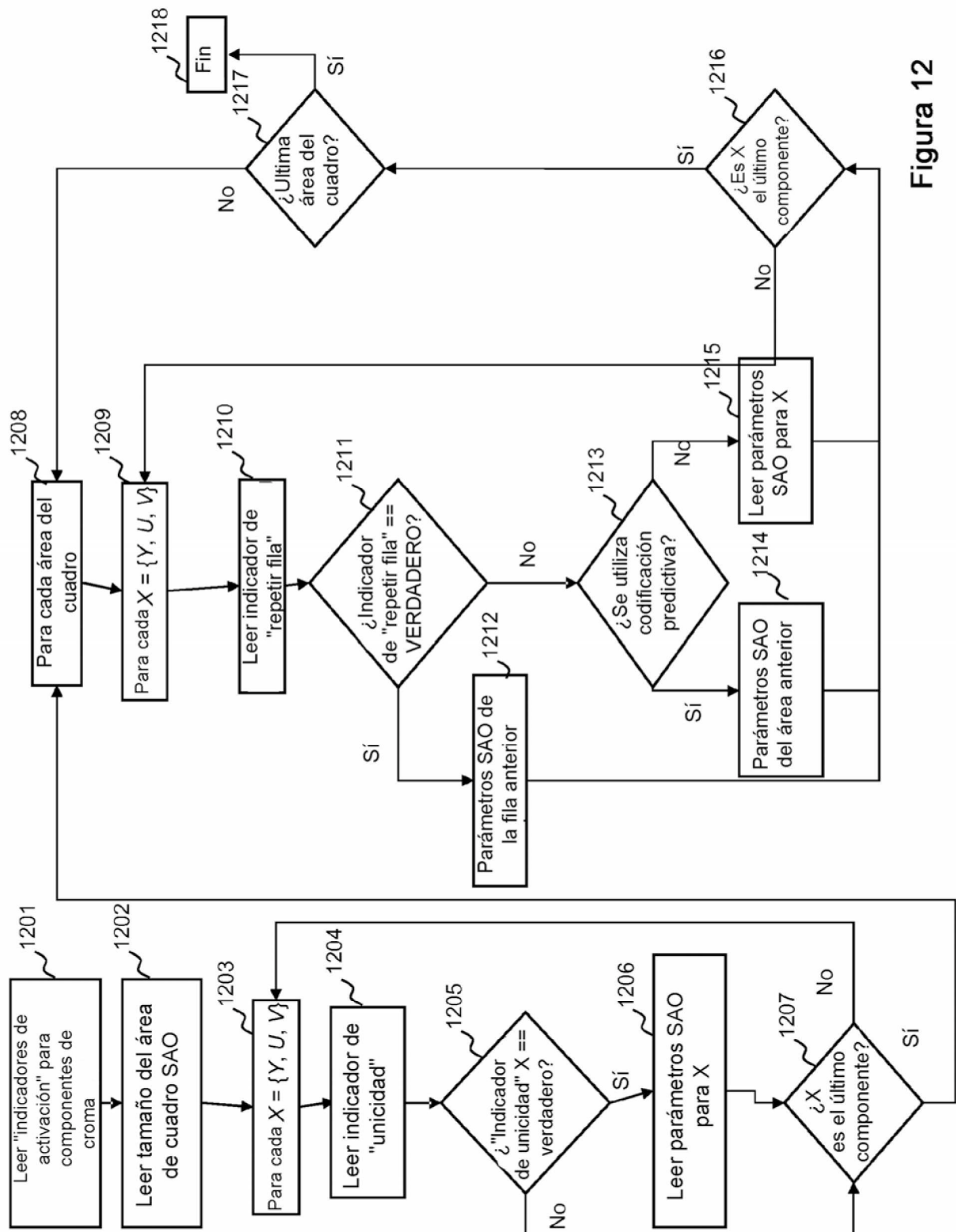


Figura 12

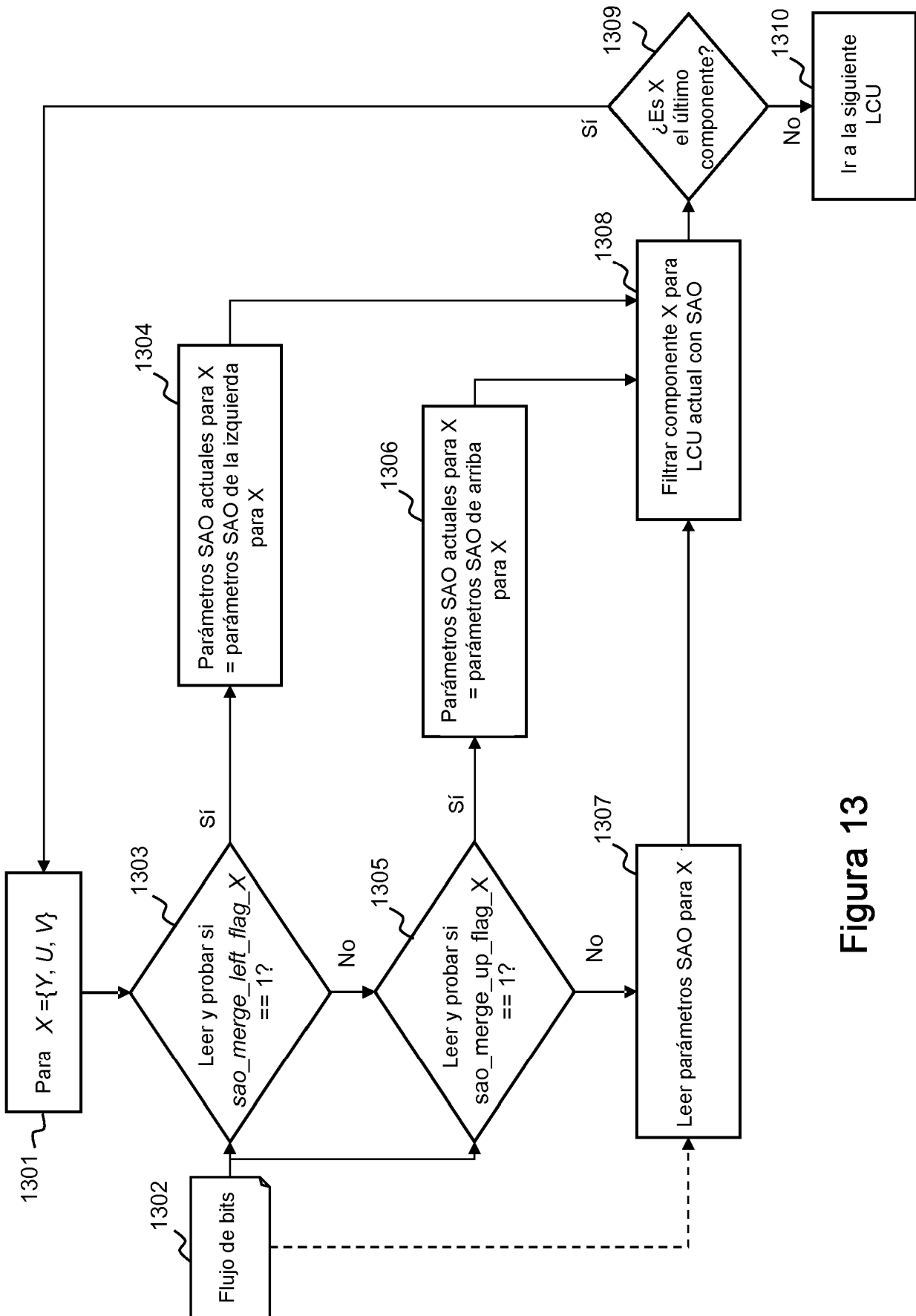


Figura 13

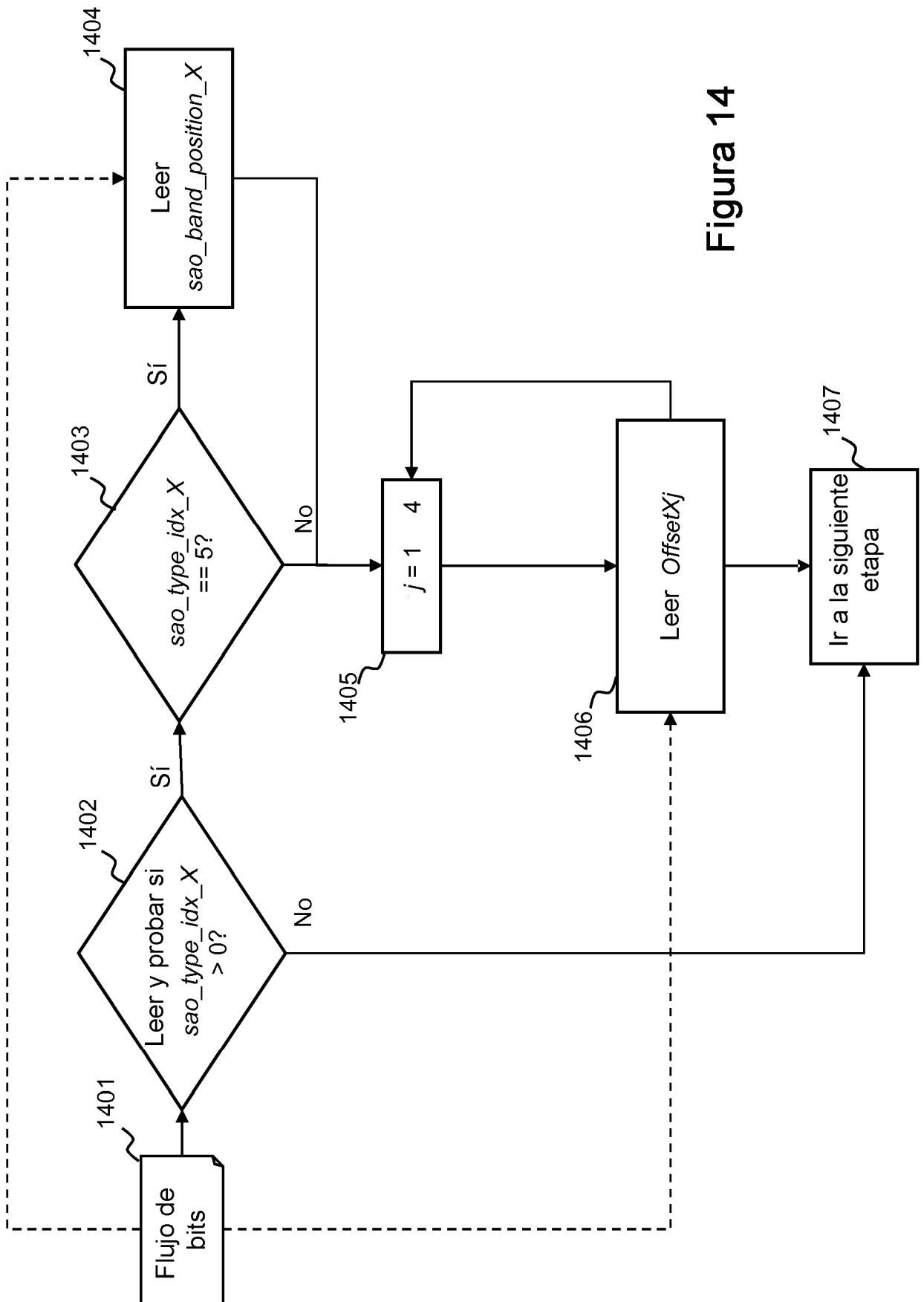
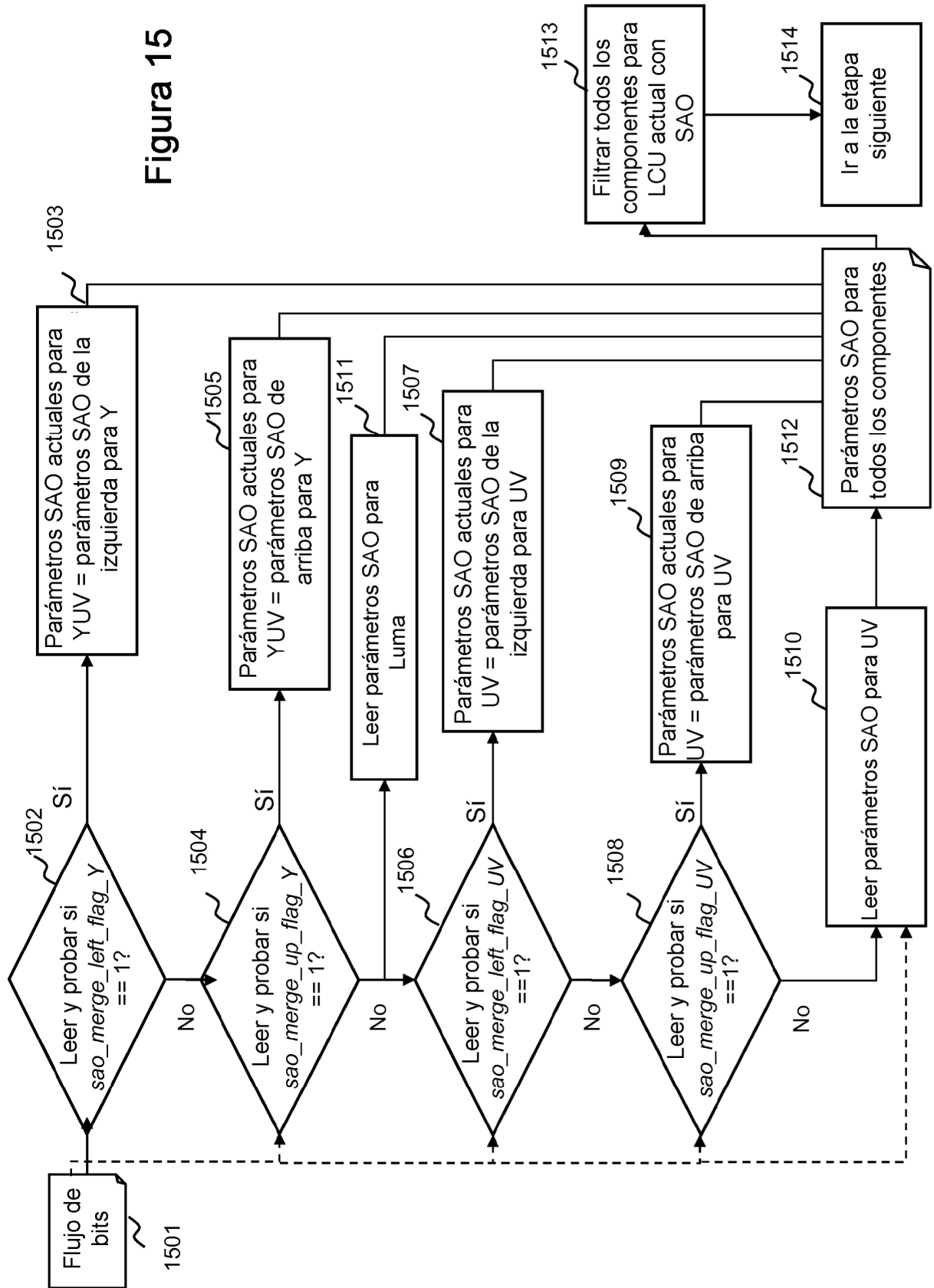
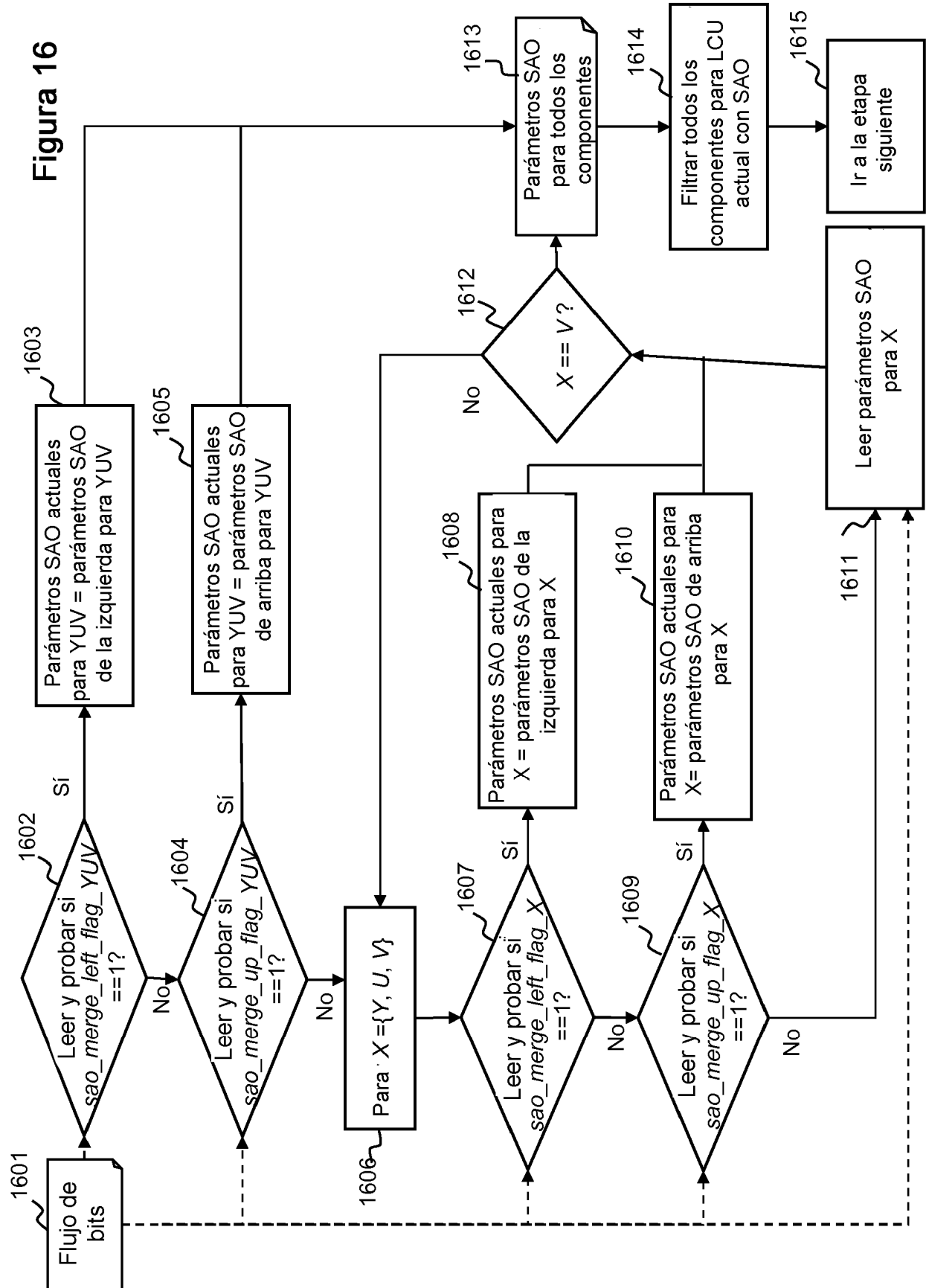


Figura 14

Figura 15





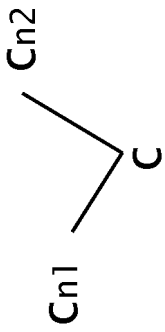
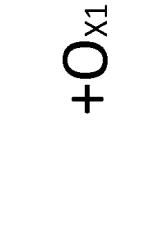
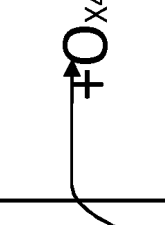
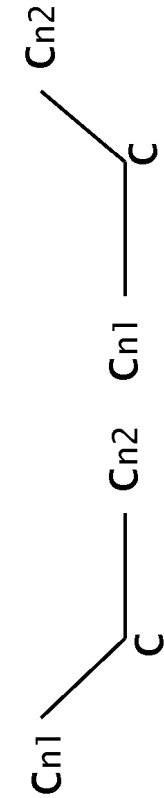
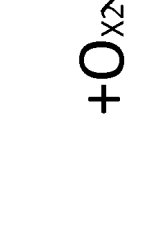
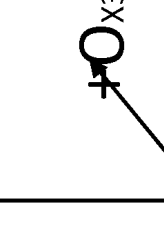
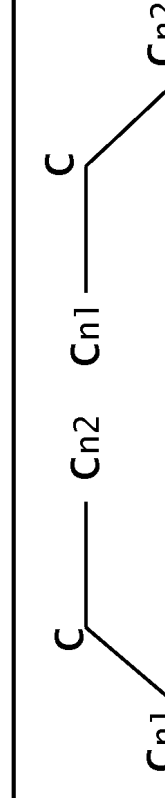
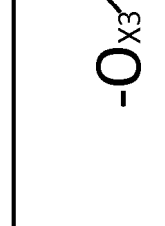
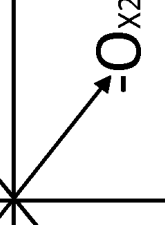
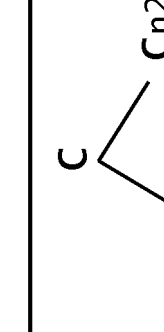
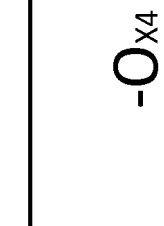
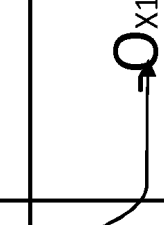
Categoría	Tipo de señal	Desplazamiento X	Desplazamiento Z
1			
2			
3			
4			

Figura 17

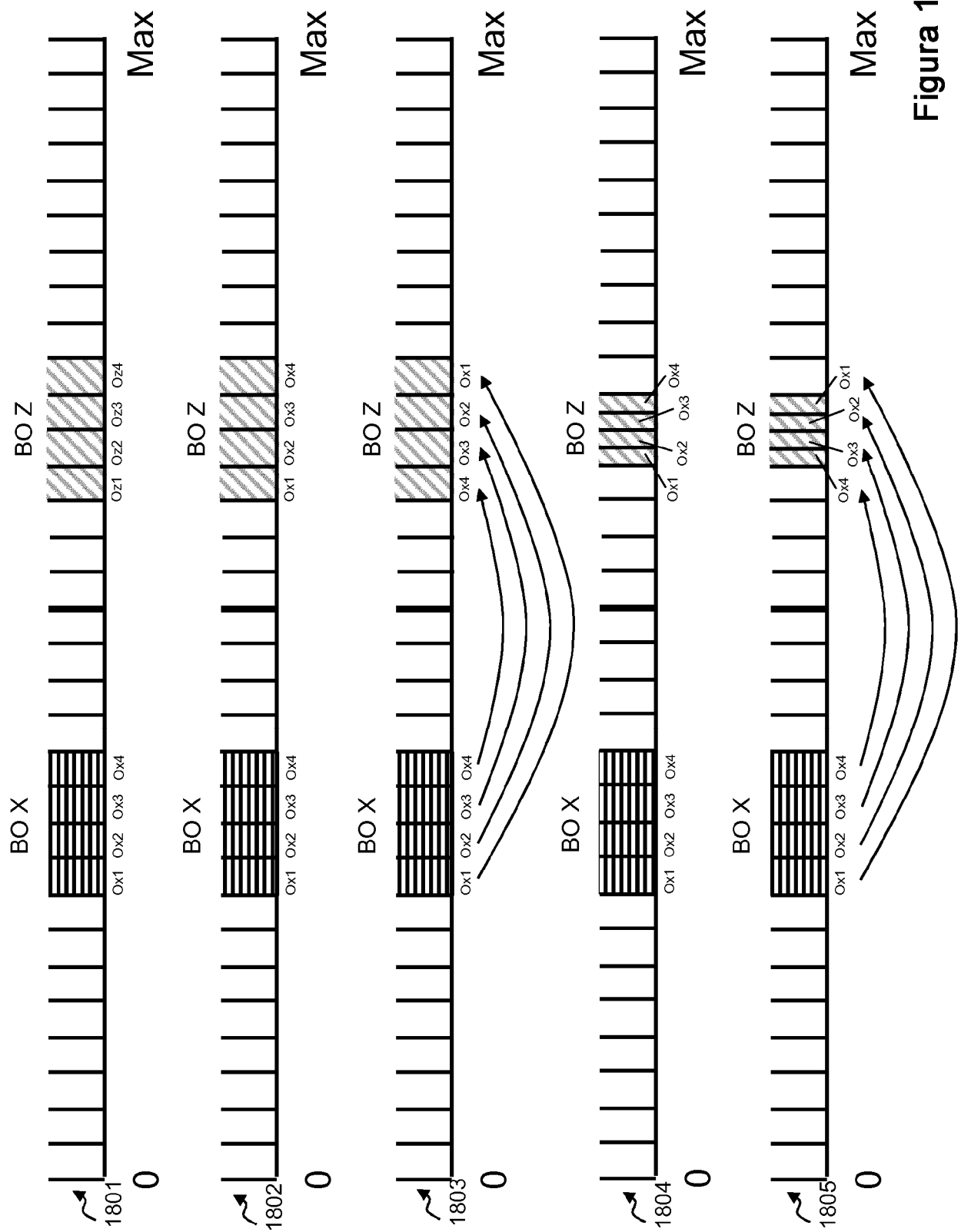


Figura 18

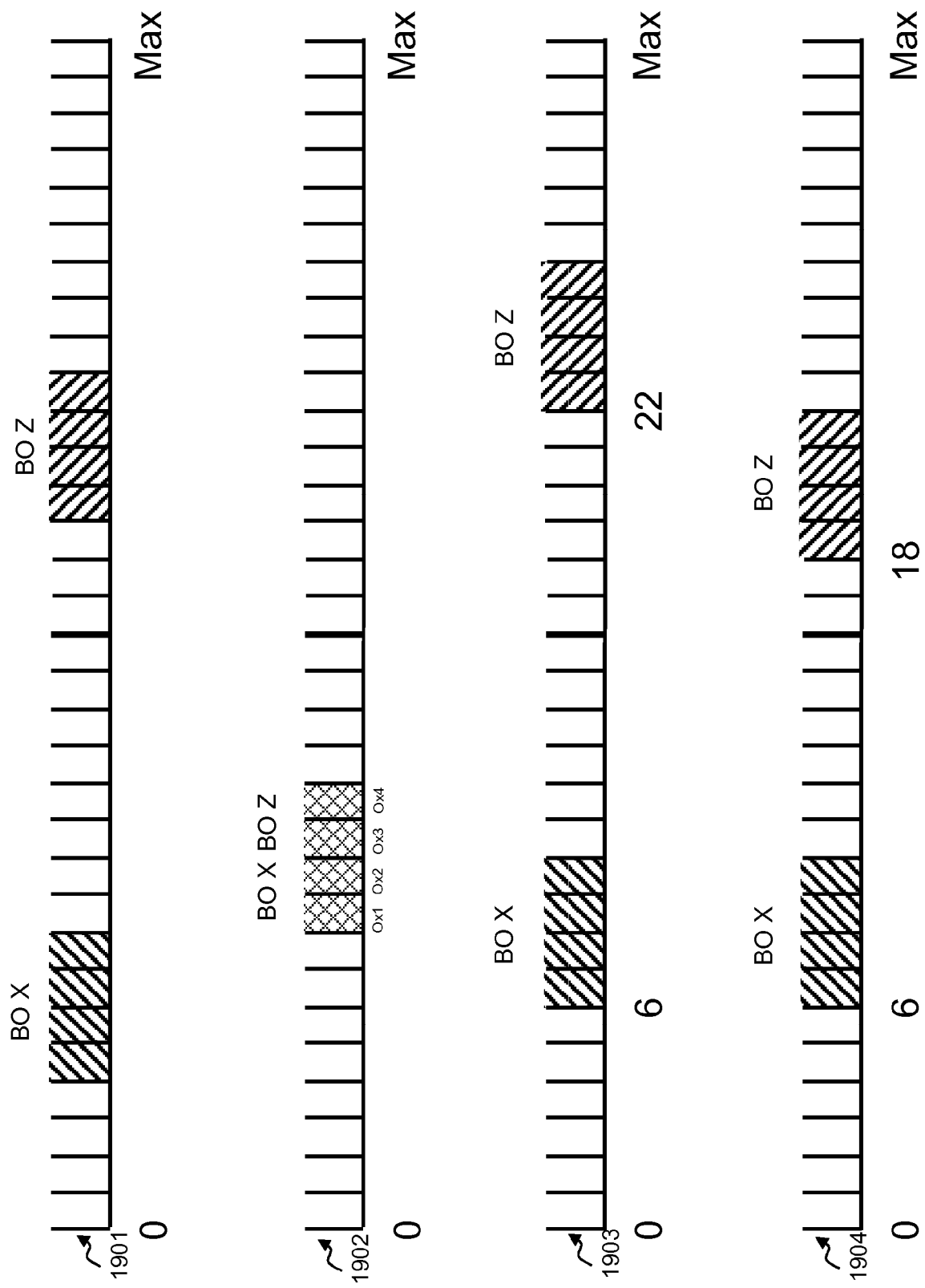


Figura 19

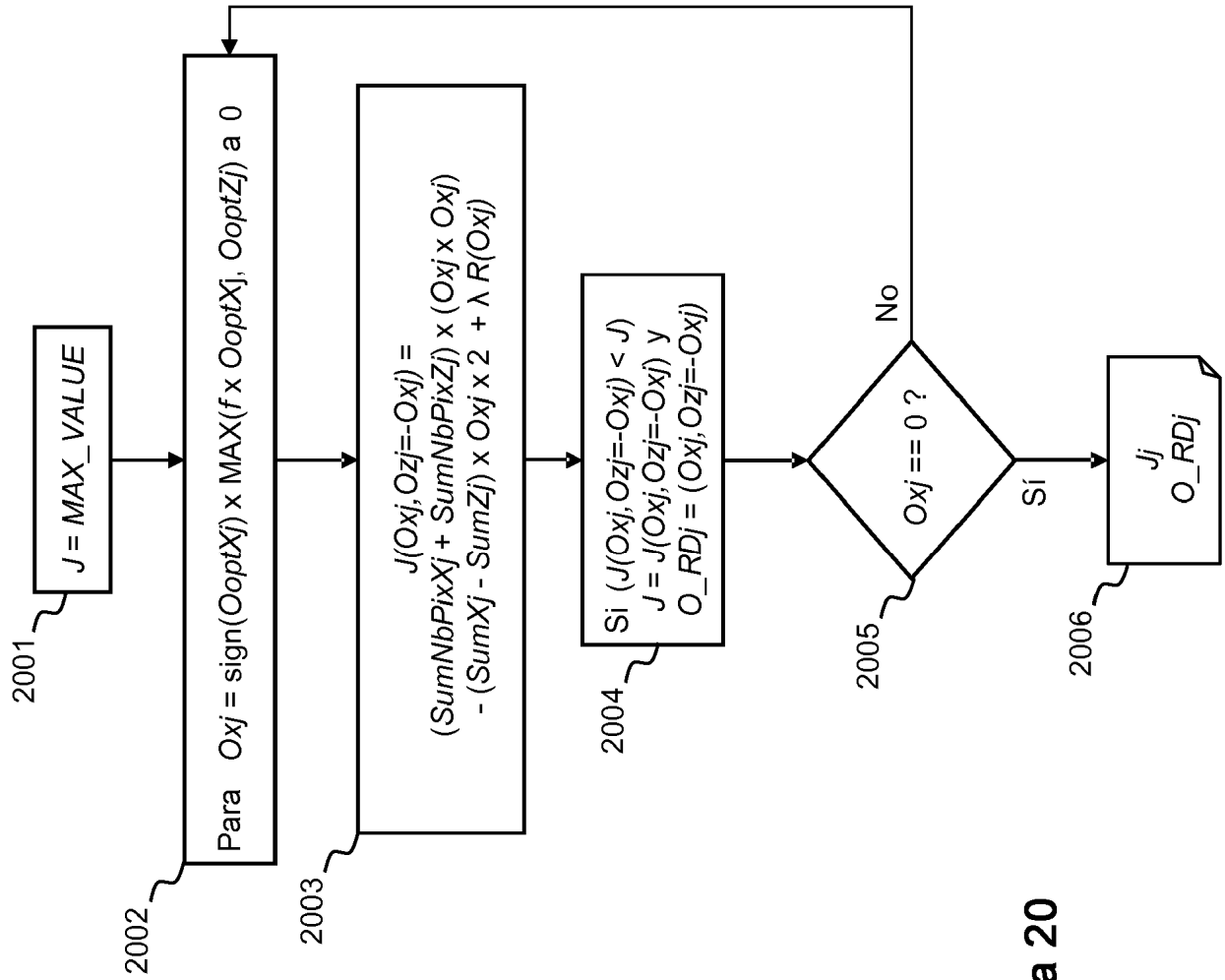


Figura 20

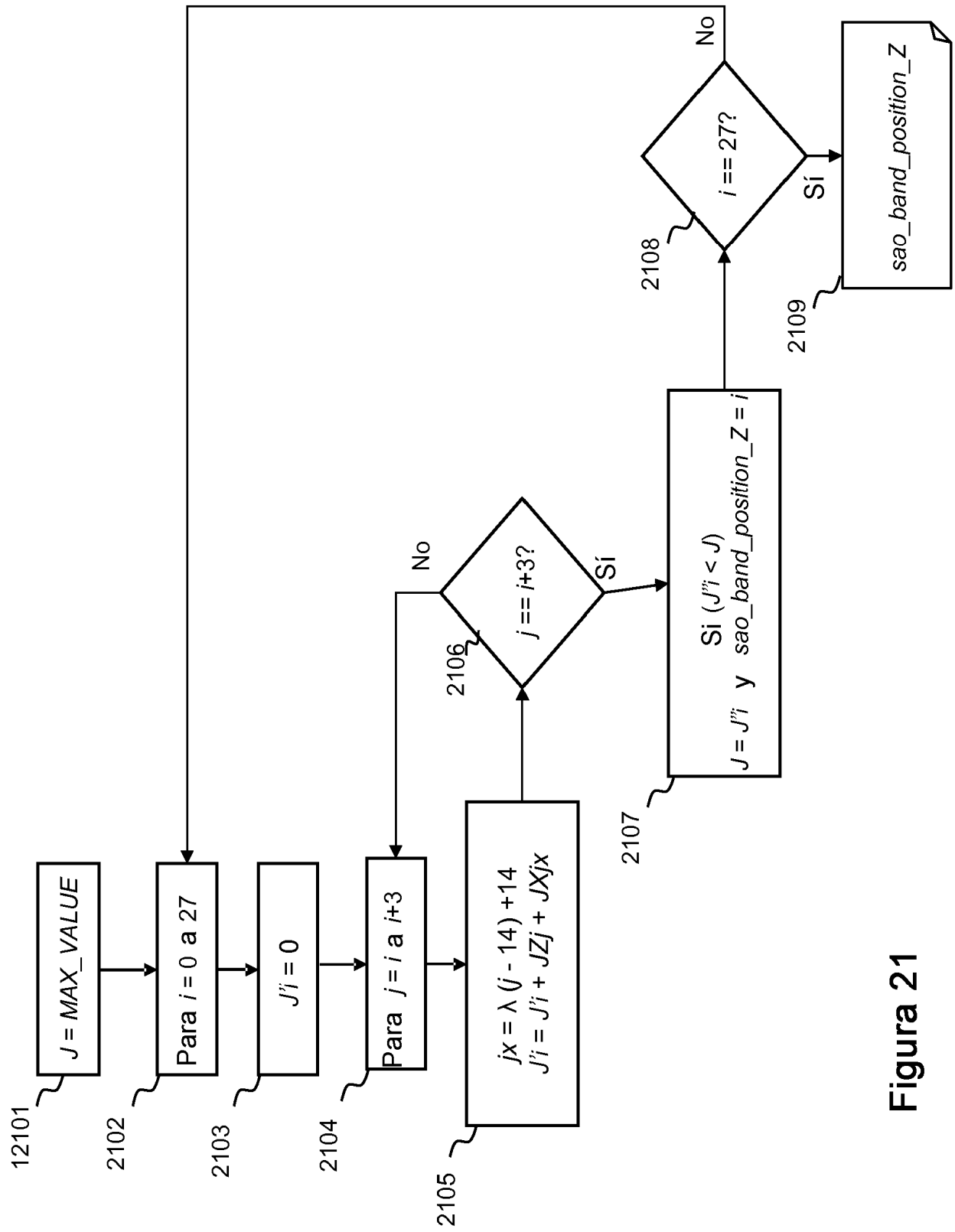


Figura 21

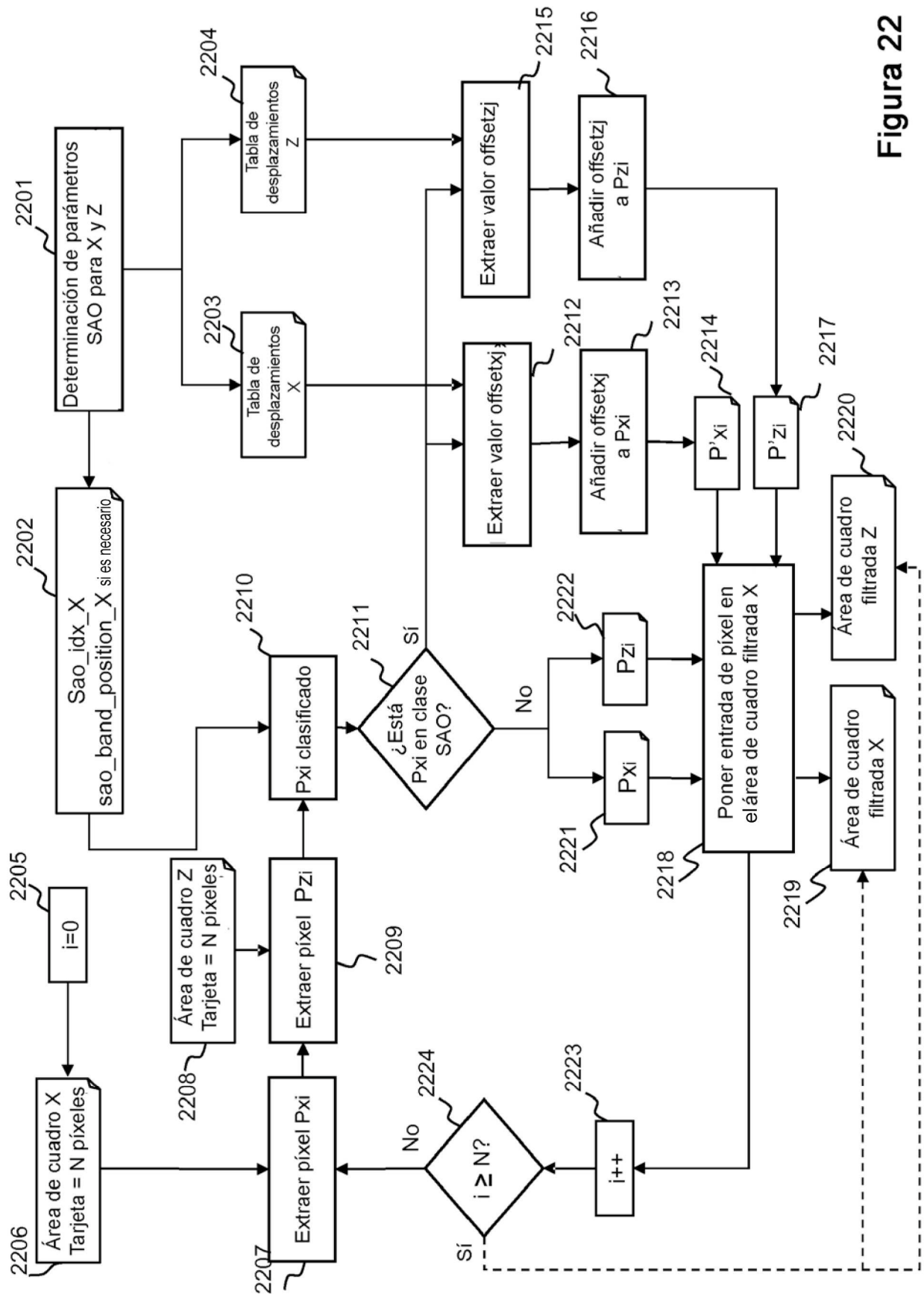
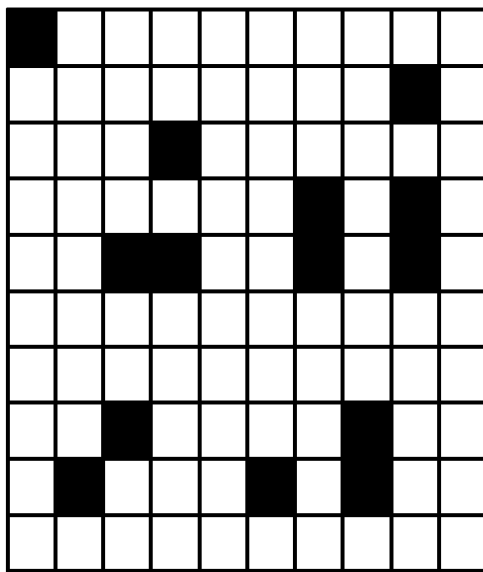
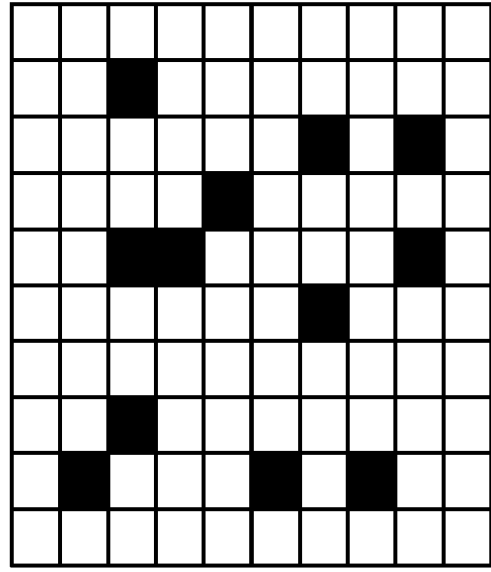


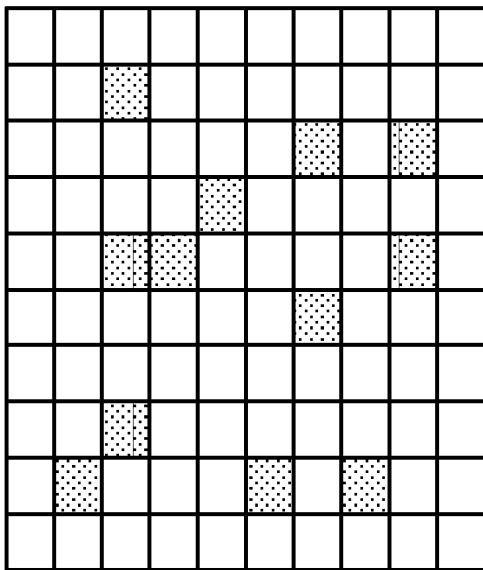
Figura 22



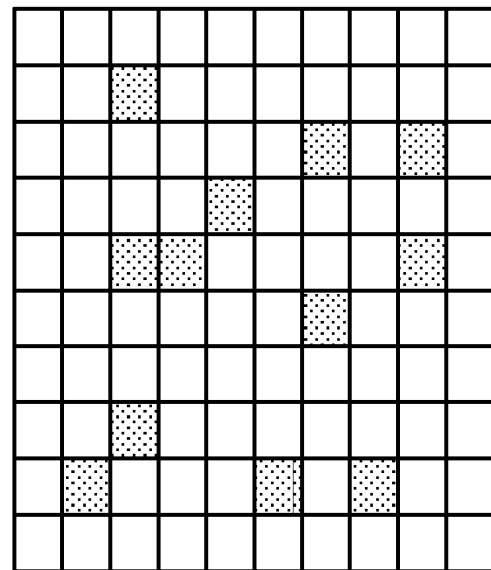
2302



2304



2301



2303

Figura 23

REFERENCIAS CITADAS EN LA DESCRIPCIÓN

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.

Literatura no patente citada en la descripción

- Sample adaptive offset for HEVC. **CHIH-MING FU et al.** MULTIMEDIA SIGNAL PROCESSING (MMSP). IEEE, 17 October 2011, 1-5