

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 627 285**

51 Int. Cl.:

H04N 19/176 (2014.01)

H04N 19/117 (2014.01)

H04N 19/14 (2014.01)

H04N 19/82 (2014.01)

H04N 19/86 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **16.12.2011 E 15192776 (1)**

97 Fecha y número de publicación de la concesión europea: **01.03.2017 EP 3001682**

54 Título: **Control del filtrado de desbloqueo**

30 Prioridad:

01.03.2011 US 201161447862 P

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

27.07.2017

73 Titular/es:

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

72 Inventor/es:

**NORKIN, ANDREY;
ANDERSSON, KENNETH y
SJÖBERG, RICKARD**

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 627 285 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

Control del filtrado de desbloqueo

Sector técnico

5 Las presentes realizaciones se refieren en general al control del filtrado y en particular a controlar el filtrado de desbloqueo sobre fronteras de bloque en un fotograma de vídeo.

Antecedentes

10 Los filtros de desbloqueo se utilizan en los estándares de codificación de vídeo para combatir las aberraciones de bloqueo. Las aberraciones de bloqueo aparecen debido a que los fotogramas de vídeo originales se dividen en bloques que son procesados de manera relativamente independiente. Las aberraciones de bloqueo pueden, por ejemplo, aparecer debido a diferentes intra predicciones de los bloques, a los efectos de la cuantificación y a la compensación de movimiento. A continuación se describen dos variantes particulares de desbloqueo.

15 En la codificación de vídeo del estado del arte, tal como H.264, existe un filtrado de desbloqueo, denominado también filtro de bucle, tras la predicción y reconstrucción residual, pero antes del almacenamiento de la reconstrucción para una posterior referencia durante la codificación o la decodificación de los fotogramas siguientes. El filtrado de desbloqueo consiste en varias etapas tales como decisiones de filtrado, operaciones de filtrado, una función de recorte de cambios de valores de píxel. La decisión de filtrar o no la frontera se toma basándose en la evaluación de varias condiciones. Las decisiones de filtro dependen del tipo de macrobloque (MB – Macro Block, en inglés), de la diferencia del vector de movimiento (MV – Motion Vector, en inglés) entre bloques vecinos, de si los bloques vecinos han codificado residuos y de la estructura local de los bloques actuales y/o vecinos.

20 Entonces, la cantidad de filtrado para un píxel depende, entre otras cosas, de la posición de ese píxel con respecto a la frontera o límite de bloque y del valor del parámetro de cuantificación (QP – Quantization Parameter, en inglés) utilizado para la codificación residual.

La decisión de filtrado se basa en comparar tres diferencias de píxel con umbrales. Los umbrales están adaptados al parámetro de cuantificación (QP). Por ejemplo, sea una frontera de bloque vertical de

25 a b c d | e f g h

en la que a, b, c y d denotan los valores de píxel de los píxeles de una fila de píxeles en el bloque actual y, e, f, g y h denotan los correspondientes valores de píxel de los píxeles de una fila de píxeles correspondiente en el bloque vecino. Si se cumplen las condiciones siguientes la decisión de filtro es positiva, por ejemplo, $\text{abs}(d-e) < \text{thr1}$, $\text{abs}(c-d) < \text{thr2}$, $\text{abs}(e-f) < \text{thr2}$, donde thr1 y thr2 están adaptados basándose en el QP.

30 Existen dos modos de filtrado en H.264. En el primer modo de filtrado, denominado filtrado normal, el filtrado puede describirse como un valor delta con el cual el filtrado cambia el valor actual. El filtrado para los píxeles más cercanos a la frontera de bloque es $d' = d + \text{delta}$ y $e' = e - \text{delta}$, donde delta ha sido recortado hasta un umbral (threshold, en inglés) $\pm \text{thr3}$, hasta un valor que está restringido por el QP. Se permite con ello un mayor filtrado, por ejemplo, para QP alto que para QP bajo. El recortado puede describirse como $\text{delta_recortado} = \max(-\text{thr3}, \min(\text{thr3}, \text{delta}))$, donde thr3 está controlando la potencia de filtrado. Un valor mayor de thr3 significa que el filtrado es más potente, lo que significa que se producirá un efecto de filtrado de paso bajo más potente.

La potencia del filtrado puede aumentar si cualquiera de las siguientes dos condiciones se cumple también, por ejemplo $\text{abs}(b-d) < \text{thr2}$ y $\text{abs}(e-g) < \text{thr2}$. La potencia del filtro se adapta recortando menos el delta, por ejemplo permitiendo una mayor variación.

40 El segundo modo de filtrado, denominado filtrado potente, se aplica solo para fronteras de intra macrobloques, cuando se cumple la condición siguiente $\text{abs}(d-e) < \text{thr1}/4$.

Para más información acerca del filtrado de desbloqueo en H.264 se hace referencia a List et al., Adaptive Deblocking Filter, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, Julio de 2003.

45 En el documento Borrador de especificación HEVC (High Efficiency Video Coding) "Test model under Consideration", ITU-T SG16 WP3, JCTVC-B205, capítulo 6.5 In-loop filter process, el filtro de desbloqueo funciona de manera diferente al H.264. El filtrado se lleva a cabo si al menos uno de los bloques en el lado de la frontera es intra, o tiene coeficientes distintos de cero, o la diferencia entre los componentes del vector de movimiento de los bloques es mayor o igual a un píxel entero. Por ejemplo, cuando se filtra la frontera entre los bloques con una frontera de bloque vertical de

50 $p_3; p_2; p_1; p_0; | q_0; q_1; q_2; q_3;$

denotando p_j el valor de píxel del número de píxel j del número de fila i en el bloque actual, y denotando q_j el valor de píxel del número de píxel j del número de fila i en el bloque vecino, $i = 0 \dots 7$, $j = 0 \dots 3$, entonces debe cumplirse la también siguiente condición:

$$d = |p_{2_2} - 2 \times p_{1_2} + p_{0_2}| + |q_{2_2} - 2 \times q_{1_2} + q_{0_2}| + |p_{2_5} - 2 \times p_{1_5} + p_{0_5}| + |q_{2_5} - 2 \times q_{1_5} + q_{0_5}| < \beta$$

5 en la que β depende de QP. En la especificación HEVC mencionada anteriormente, existe una tabla de β , en la que β aumenta con QP.

Si las condiciones se cumplen y el filtrado se efectúa entre el bloque actual y el bloque vecino, tiene lugar uno de dos tipos de filtrado, denominados filtrado débil y potente, respectivamente. La elección entre filtrado potente y débil se realiza separadamente para cada línea dependiendo de las siguientes condiciones. Para cada línea $i = 0 \dots 7$, el filtrado potente se efectúa si todas las condiciones siguientes son ciertas; de lo contrario se efectúa un filtrado débil:

$$d < (\beta \gg 2)$$

$$(|p_{3_i} - p_{0_i}| + |q_{0_i} - q_{3_i}|) < (\beta \gg 3)$$

$$|p_{0_i} - q_{0_i}| < ((5 \times t_c + 1) \gg 1)$$

donde t_c y β dependen de QP, y \gg denota un operador de desplazamiento lógico hacia la derecha.

Un filtrado débil se lleva a cabo basado en las condiciones anteriores. El filtrado real funciona calculando un desfase (Δ), sumándolo al valor de píxel original y recortando la suma hasta un valor de píxel de salida filtrado en el rango de 0-255:

$$\Delta = \text{Clip}(-t_c, t_c, (13 \times (q_{0_i} - p_{0_i}) + 4 \times (q_{1_i} - p_{1_i}) - 5 \times (q_{2_i} - p_{2_i}) + 16) \gg 5)$$

$$p_{0_i} = \text{Clip}_{0-255}(p_{0_i} + \Delta)$$

$$q_{0_i} = \text{Clip}_{0-255}(q_{0_i} - \Delta)$$

$$p_{1_i} = \text{Clip}_{0-255}(p_{1_i} + \Delta/2)$$

$$q_{1_i} = \text{Clip}_{0-255}(q_{1_i} - \Delta/2)$$

donde la función de recorte $\text{Clip}(A, B, x)$ se define como $\text{Clip}(A, B, x) = A$ si $x < A$, $\text{Clip}(A, B, x) = B$ si $x > B$ y $\text{Clip}(A, B, x) = x$ si $A \leq x \leq B$, y $\text{Clip}_{0-255}(X)$ se define como $\text{Clip}(0, 255, x)$.

El modo de filtrado potente se lleva a cabo mediante el siguiente conjunto de operaciones:

$$p_{0_i} = \text{Clip}_{0-255}((p_{2_i} + 2 \times p_{1_i} + 2 \times p_{0_i} + 2 \times q_{0_i} + q_{1_i} + 4) \gg 3)$$

$$q_{0_i} = \text{Clip}_{0-255}((p_{1_i} + 2 \times p_{0_i} + 2 \times q_{0_i} + 2 \times q_{1_i} + q_{2_i} + 4) \gg 3)$$

$$p_{1_i} = \text{Clip}_{0-255}((p_{2_i} + p_{1_i} + p_{0_i} + q_{0_i} + 2) \gg 2)$$

$$q_{1_i} = \text{Clip}_{0-255}((p_{0_i} + q_{0_i} + q_{1_i} + q_{2_i} + 2) \gg 2)$$

$$p_{2_i} = \text{Clip}_{0-255}((2 \times p_{3_i} + 3 \times p_{2_i} + p_{1_i} + p_{0_i} + q_{0_i} + 4) \gg 3)$$

$$20 \quad q_{2_i} = \text{Clip}_{0-255}((p_{0_i} + q_{0_i} + q_{1_i} + 3 \times q_{2_i} + 2 \times q_{3_i} + 4) \gg 3)$$

Las decisiones de filtrado de desbloqueo de acuerdo con HEVC pueden llevar a un filtrado de desbloqueo impreciso sobre fronteras de bloque para ciertos bloques. En particular, bloques vecinos que tienen diferentes niveles de estructuras locales podrían ser manejados de manera incorrecta en HEVC filtrando demasiado uno de los bloques para producir con ello represiones, y filtrar eliminando estructuras locales en el bloque.

Matthias Narroschke et al: Reduction of operations in the critical path of the deblocking filter”, 95, reunión MPEG 24-1-2011-28-1-2011; Dagu; nº m18974, 22 de enero de 2011, se refiere a un método para la determinación de decisiones de filtro. Píxeles a ambos lados de la frontera de bloque se utilizan para determinar una decisión de filtrado, es decir cuántos píxeles deben considerarse para desbloqueo, pero la misma decisión de filtro se utiliza a ambos lados de la frontera de bloque.

El documento US 2010/142844 A1 hace referencia a una operación de desbloqueo en la que se determina una región de interés que determina el número de píxeles que deben ser filtrados. El número de píxeles puede ser diferente a cada lado de la frontera de bloque. La región de interés se determina sobre la base de N0 (para el bloque B0) y N1 (para el bloque B1) que son calculados sobre la base de una métrica de actividad y un recuento de aberraciones. El recuento de aberraciones se determina para cada bloque y se determina una actividad para cada fila y para cada columna de cada bloque. Durante la determinación, se utiliza el recuento de aberraciones para N0 píxeles a ambos lados de la frontera de bloque y, también durante la determinación, se utiliza el recuento de aberraciones para N1 píxeles a ambos lados de la frontera de bloque.

Sumario

Por ello, resulta necesario un control eficiente del filtrado de desbloqueo que pueda ser utilizado para reducir las aberraciones de bloqueo en las fronteras de bloque y que no tenga los inconvenientes mencionados anteriormente. Un objetivo general es proporcionar un control eficiente del filtrado de desbloqueo. Un objetivo particular es proporcionar decisiones de filtrado asimétricas sobre una frontera de bloque.

Esto se consigue mediante las características de las reivindicaciones independientes. Las realizaciones preferidas son el objeto de las reivindicaciones dependientes.

Las realizaciones consiguen decisiones de desbloqueo asimétricas que controlan el filtrado de desbloqueo para que se adapte a la estructura a cada lado de una frontera de bloque. Las decisiones asimétricas significan que la cantidad de filtrado aplicado a un lado de la frontera de bloque puede ser diferente de la cantidad de filtrado aplicado al otro lado de la frontera de bloque, proporcionando así una adaptación adicional a la estructura local. Esto mejora la calidad objetiva y subjetiva del vídeo.

Breve descripción de los dibujos

La invención, junto con otros objetos y ventajas de la misma, puede comprenderse mejor haciendo referencia a la descripción siguiente tomada junto con los dibujos que se acompañan, en los cuales:

- la Fig. 1 es un diagrama de flujo de un método para el control del filtrado de acuerdo con una realización;
- las Figs. 2A y 2B ilustran dos realizaciones de bloques vecinos y una frontera de bloque sobre los cuales puede aplicarse el filtrado de desbloqueo;
- la Fig. 3 es un diagrama de flujo que ilustra etapas adicionales, opcionales del método en la Fig. 1 de acuerdo con una realización;
- la Fig. 4 es un diagrama de flujo que ilustra una realización de las etapas de determinación en la Fig. 1;
- la Fig. 5 es un diagrama de flujo que ilustra etapas adicionales y opcionales del método en la Fig. 1 de acuerdo con otra realización;
- la Fig. 6 es un diagrama de flujo que ilustra etapas adicionales, opcionales del método de la Fig. 1 y una realización de las etapas de determinación de la Fig. 1;
- la Fig. 7 es un diagrama de flujo que ilustra una etapa adicional, opcional del método de la Fig. 1 de acuerdo con una realización;
- la Fig. 8 es un diagrama de bloques esquemático de una realización de un dispositivo de control del filtrado;
- la Fig. 9 es un diagrama de bloques esquemático de otra realización de un dispositivo de control del filtrado;
- la Fig. 10 es un diagrama de bloques esquemático de otra realización de un dispositivo de control del filtrado;
- la Fig. 11 es un diagrama de bloques esquemático de otra realización más de un dispositivo de control del filtrado;
- la Fig. 12 es un diagrama de bloques esquemático de una implementación de software de un dispositivo de control del filtrado en un ordenador de acuerdo con una realización;
- la Fig. 13 es un diagrama de bloques esquemático de un codificador de acuerdo con una realización;
- la Fig. 14 es un diagrama de bloques esquemático de un decodificador de acuerdo con una realización;

la Fig. 15 es un diagrama de bloques esquemático de un equipo de usuario de acuerdo con una realización;

la Fig. 16 es un diagrama de bloques esquemático de un equipo de usuario de acuerdo con otra realización; y

la Fig. 17 es una vista global esquemática de una porción de una red de comunicación que comprende un dispositivo de red de acuerdo con una realización.

5 Descripción detallada

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

Las realizaciones se refieren en general al control del filtrado y al control del filtrado de desbloqueo en fronteras de bloques en un fotograma de vídeo. El control del filtrado de las realizaciones proporciona decisiones de desbloqueo asimétricas con respecto a las fronteras de bloques mediante la toma de decisiones de filtrado independientes para los bloques de píxeles separados por una frontera de bloque. Esto significa que el filtrado de desbloqueo puede manejar bloques vecinos que tienen diferentes niveles de estructuras locales, para adaptar con ello el filtrado de desbloqueo particular a cada bloque sobre la base de sus estructuras locales.

Como es bien conocido en el sector, un fotograma de vídeo está dividido en bloques de píxeles no superpuestos que son codificados y decodificados de acuerdo con varios modos de intra e inter codificación disponibles. En general, un fotograma de vídeo está dividido en macrobloques no superpuestos de 16 x 16 píxeles. Tal macrobloque puede a su vez estar dividido en bloques más pequeños de diferentes tamaños, tal como 4 x 4 u 8 x 8 píxeles. No obstante, también son posibles bloques rectangulares de acuerdo con las realizaciones, tal como 4 x 8, 8 x 4, 8 x 16 o 16 x 8. Las realizaciones pueden aplicarse a cualquiera de tales bloques de píxeles, incluidos macrobloques o incluso bloques de píxeles más grandes.

En el reciente estándar Codificación de vídeo de alta eficiencia (HEVC – High Efficiency Video Coding, en inglés), se utilizan unidades de codificación (CU - Coding Units, en inglés), unidades de predicción (PU – Prediction Units, en inglés) y unidades de transformación (TU – Transform Units, en inglés). Las unidades de predicción se definen dentro de una unidad de codificación y contienen los modos de intra o inter predicción. Las unidades de transformación se definen dentro de una unidad de codificación y el tamaño de transformación más grande es 32 x 32 píxeles y el tamaño más pequeño es 4 x 4 píxeles. El tamaño de la CU varía actualmente de 64 x 64 píxeles (el más grande) a 8 x 8 píxeles (el más pequeño). De este modo, la CU más grande puede dividirse en CU más pequeñas, dependiendo el “nivel de granularidad” de las características locales del fotograma. Esto significa que la CU más grande puede dividirse en CU más pequeñas de diferentes tamaños. Las realizaciones pueden utilizarse asimismo junto con unidades de codificación tales que se consideran englobadas por la expresión “bloque de píxeles” tal como se utiliza en esta memoria.

Cada píxel asignado en el bloque tiene un valor de píxel respectivo. Los fotogramas de vídeo generalmente tienen valores de color asignados a los píxeles, donde los valores de color se representan en unos formatos de color definidos. Uno de los formatos de color comunes utiliza un componente de luminancia y dos componentes de crominancia para cada píxel, aunque existen otros formatos, tales como utilizar componentes rojo, verde y azul para cada píxel.

Tradicionalmente, el filtrado del componente de luminancia y el filtrado del componente de crominancia se realizan de manera separada, posiblemente empleando diferentes decisiones de filtrado y diferentes filtros de desbloqueo. Es, no obstante, posible, que las decisiones de filtrado de luminancia se utilicen en filtrado croma, como H.264. Las realizaciones pueden ser aplicadas al control del filtrado para el componente de luminancia, el componente de crominancia o tanto para el componente de luminancia como el componente de crominancia. En una realización particular, las realizaciones se aplican al control del filtrado de la luminancia o luma. Decisiones de filtrado, o partes de decisiones de filtrado para un componente, tal como luma, pueden utilizarse entonces cuando se toman decisiones de filtrado para otros componentes, tal como croma.

El filtrado de desbloqueo se efectúa sobre una frontera, borde o límite entre bloques vecinos. En consecuencia, tales fronteras pueden ser fronteras verticales 1, véase la Fig. 2A, entre dos bloques vecinos 10, 20 existentes a cada lado del fotograma de vídeo. De manera alternativa, las fronteras son fronteras horizontales 1, véase la Fig. 2B, entre dos bloques vecinos 10, 20, en los que un bloque 10 está situado por encima del otro bloque 20 en el fotograma de vídeo. En una realización particular, las fronteras verticales son filtradas primero empezando desde la frontera más a la izquierda y avanzando a través de las fronteras hacia el lado derecho en su orden geométrico. A continuación, las fronteras horizontales son filtradas empezando con la frontera de la parte superior y avanzando a través de las fronteras hacia la parte inferior en su orden geométrico. Las realizaciones no están, no obstante, limitadas a este orden de filtrado particular y pueden, de hecho, ser aplicadas a cualquier orden de filtrado predefinido. En una realización particular, las fronteras en el borde del fotograma de vídeo preferiblemente no se filtran y por ello están excluidas del filtrado de desbloqueo.

La Fig. 1 es un diagrama de flujo de un método para el control del filtrado aplicable a un bloque de múltiples píxeles en un fotograma de vídeo de acuerdo con una realización. El método de la Fig. 1 generalmente se inicia en la etapa S1 en la que se calcula un primer valor de decisión de filtro para el bloque sobre la base de al menos

$|p2_i - 2p1_i + p0_i|$, donde $p0_i$ denota un valor de píxel de un píxel 11 más cercano, en una primera línea de píxeles 11, 13, 15, 17 en el bloque 10, a una frontera de bloque 1 con un bloque vecino 20 de múltiples píxeles 21, 23, 25, 27 en el fotograma de vídeo, $p1_i$ denota un valor de píxel de un píxel 13 siguiente más cercano, en la primera línea 12 de píxeles 11, 13, 15, 17, a la frontera de bloque 1 y $p2_i$ denota un valor de píxel de un píxel 15 segundo siguiente más cercano, en la primera línea 12 de píxeles 11, 13, 15, 17, a la frontera de bloque 10.

La etapa S2, de manera correspondiente, calcula un segundo valor de decisión de filtro para el bloque sobre la base de al menos $|q2_i - 2q1_i + q0_i|$, donde $q0_i$ denota un valor de píxel de un píxel 21 en el bloque vecino 20 más cercano, en una correspondiente primera línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20, a la frontera de bloque 1, $q1_i$ denota un valor de píxel de un píxel 23 del bloque vecino 20 siguiente más cercano, en la correspondiente primera línea 22 de píxeles 21, 23, 25, 27, a la frontera de bloque 1 y $q2_i$ denota un valor de píxel de un píxel 25 en el bloque vecino 20 segundo siguiente más cercano, en la correspondiente primera línea 22 de píxeles 21, 23, 25, 27, a la frontera de bloque 1.

La primera línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 y la correspondiente primera línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 pertenecen a la misma línea horizontal de píxeles, es decir, la fila de píxeles, que se extiende sobre una frontera vertical 1, véase la Fig. 2A, o pertenecen a la misma línea vertical de píxeles, es decir, la columna de píxeles, que se extiende sobre una frontera horizontal 1, véase la Fig. 2B. Por ello, la primera línea 12 de píxeles 11, 13, 15, 17 y la correspondiente primera línea 22 de píxeles 21, 23, 25, 27 son perpendiculares a la frontera de bloque 1 entre el bloque 10 y el bloque vecino 20. Además, la primera línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 y la correspondiente primera línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 tienen el mismo número de línea. Por ejemplo, si el bloque 10 y el bloque vecino 20 comprenden cada uno N, tal como ocho, filas o columnas de píxeles, teniendo números de fila o columna $i = 0 \dots N-1$, entonces la primera línea 10 de píxeles 11, 13, 15, 17 tiene el número de línea i en el bloque 10, y la correspondiente primera línea 20 de píxeles 21, 23, 25, 27 tiene también el número de línea i , pero en el bloque vecino 20. Así, la primera línea 12 de píxeles 11, 13, 15, 17 en el bloque y la correspondiente primera línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 son líneas opuestas con respecto a la frontera de bloque 1.

De acuerdo con las realizaciones, "línea de píxeles" y "correspondiente línea de píxeles" se emplean para denotar una "fila de píxeles" y una "correspondiente fila de píxeles" en el caso de una frontera de bloque vertical como en la Fig. 2A, y denotan una "columna de píxeles" y una "correspondiente columna de píxeles" en el caso de una frontera de bloque horizontal como en la Fig. 2B.

La primera línea 12 de píxeles 11, 13, 15, 17 y la correspondiente primera línea 22 de píxeles 21, 23, 25, 27 podrían ser líneas predefinidas en el bloque 10 y en el bloque vecino 20, respectivamente. Así, la primera línea 12 de píxeles 11, 13, 15, 17 y la correspondiente primera línea 22 de píxeles 21, 23, 25, 27 tienen un número de línea i predefinido y fijo con respecto a cada frontera de bloque 1 para la cual se aplica un control del filtrado. De manera alternativa, la primera línea 12 de píxeles 11, 13, 15, 17 y la correspondiente primera línea 22 de píxeles 21, 23, 25, 27 podrían representar una línea actual y una línea correspondiente respectivamente, lo que se explica con más detalle en esta memoria.

El cálculo del primer valor de decisión de filtro en la etapa S1 y el cálculo del segundo valor de decisión de filtro en la etapa S2 pueden ser llevados a cabo en serie en cualquier orden, es decir, precediendo la etapa S1 a la etapa S2, o precediendo la etapa S2 a la etapa S1, o al menos parcialmente en paralelo. Los resultados de estas dos etapas S1, S2 son, así, un primer valor de decisión de filtro que se calcula basándose en valores de píxel en el bloque 10 y un segundo valor de decisión de filtro que se calcula basándose en valores de píxeles en el bloque vecino 20 en el otro lado de la frontera de bloque 1 con respecto al bloque 10. Más preferiblemente, el cálculo del primer valor de decisión se lleva a cabo basándose solo en valores de píxel en el bloque 10 y por lo tanto no basándose en ningún valor de píxel en el bloque vecino 20. De manera correspondiente, el segundo valor de decisión de filtro se calcula preferiblemente basándose solo en valores de píxeles en el bloque vecino 20 y no basándose en ningún valor de píxel en el bloque 10.

El primer valor de decisión de filtro en la etapa S1 se utiliza entonces en la etapa S3 para determinar cuántos píxeles en una línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 filtrar con respecto a la frontera de bloque 1. El segundo valor de decisión de filtro se utiliza de manera correspondiente en la etapa S4 para determinar cuántos píxeles en una línea 22 correspondiente de píxeles 21, 23, 25, 27 en el bloque vecino 20 filtrar con respecto a la frontera de bloque 1. Así, se calculan valores de decisión de filtro separados para cada lado o parte de una fila o columna de píxeles que se extienden sobre la frontera de bloque 1 y se toma una decisión de filtro respectiva para cada lado o parte sobre la base del valor de decisión de filtro particular calculado para ese lado o parte.

Esto debe compararse con la técnica anterior, en la que se calcula un solo o un conjunto de valores de decisión de filtro para una línea de píxeles y la línea de píxeles correspondiente, y en la que este valor de decisión de filtro o conjunto de valores de decisión de filtro se utiliza para decidir cuántos píxeles filtrar a cada lado de la frontera de bloque. Así, en la técnica anterior se filtra siempre el mismo número de píxeles para la correspondiente línea de píxeles en el bloque vecino, puesto que se hace para hacer coincidir la línea de píxeles en el bloque.

Las presentes realizaciones por el contrario permiten un control del filtrado y un filtrado de desbloqueo asimétricos, tomando una decisión de filtro separada para la línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 y otra, diferente, decisión de filtro para la correspondiente línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20. Esto significa que basándose en los valores de decisión de filtro primero y segundo particulares, podría seleccionarse un número de píxeles en la línea 12 de píxeles 11, 13, 15, 17 para el filtrado de desbloqueo y la modificación diferente o igual que el número de píxeles en la correspondiente línea 22 de píxeles 21, 23, 25, 27 que son seleccionados para el filtrado de desbloqueo y la modificación.

En general, en esta memoria pX_y denota el valor de píxel del número de píxel X con respecto a la frontera de bloque 1 en una línea de píxeles que tiene el número de línea y en el bloque 10. De manera correspondiente, qX_y denota el valor de píxel del número de píxel X con respecto a la frontera de bloque 1 en una correspondiente línea de píxeles que tiene el número de línea y en el bloque vecino 20.

Las etapas S3 y S4 pueden ser llevadas a cabo en serie en cualquier orden o de hecho al menos parcialmente en paralelo.

En una primera realización, las etapas S1 y S2 podrían ser llevadas a cabo una vez para una frontera de bloque 1 entre el bloque 10 y el bloque vecino 20 para calcular con ello un primer valor de decisión de filtro y un segundo valor de decisión de filtro que aplica a todas las líneas 12 de píxeles 11, 13, 15, 17 en el bloque 10 y a todas las líneas 22 de píxeles 21, 23, 25, 27 correspondientes en el bloque vecino 20, respectivamente. En tal planteamiento, el mismo primer número de píxeles son preferiblemente filtrados y modificados en cada línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 con respecto a la frontera de bloque 1, donde este primer número se determina sobre la base del primer valor de decisión de filtro calculado en la etapa S1. De manera correspondiente, el mismo segundo número de píxeles son preferiblemente filtrados y modificados en cada línea 22 de píxeles 21, 23, 25, 27 correspondiente en el bloque vecino 20 con respecto a la frontera de bloque 1, en la que este segundo número se determina sobre la base del segundo valor de decisión de filtro calculado en la etapa S2.

De manera alternativa, en una segunda realización el primer valor de decisión de filtro y el segundo valor de decisión de filtro aplican a un subconjunto de líneas 12 de píxeles 11, 13, 15, 17 en el bloque 10 y a un subconjunto correspondiente de las líneas 22 de píxeles 21, 23, 25, 27 correspondientes en el bloque vecino 20. Por ejemplo, un par de valores de decisión de filtro podrían ser utilizados para las primeras cuatro líneas 12 de píxeles 11, 13, 15, 17 en el bloque y las primeras cuatro líneas 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 y, utilizándose otro par de valores de decisión de filtro para las restantes cuatro líneas 12 de píxeles 11, 13, 15, 17 en el bloque y las restantes cuatro líneas 22 de píxeles 21, 23, 25, 27 correspondientes en el bloque vecino 20.

En una tercera realización, el cálculo en la etapa S1 se efectúa para cada línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 y se efectúa entonces una determinación separada en la etapa S3 para cada una de tales líneas 12 de píxeles 11, 13, 15, 17. En tal caso, el cálculo en la etapa S2 se lleva cabo de manera correspondiente para cada línea 22 de píxeles 21, 23, 25, 27 correspondiente en el bloque vecino 20 y se lleva a cabo una determinación separada en la etapa S4 para cada una de tales líneas 22 de píxeles 21, 23, 25, 27 correspondientes.

Así, en la tercera realización, la etapa S3 comprende determinar cuántos píxeles en la primera línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 filtrar con respecto a la frontera de bloque 1 sobre la base del primer valor de decisión de filtro calculado en la etapa S1. La etapa S4 comprende determinar cuántos píxeles en la primera línea 22 de píxeles 21, 23, 25, 27 correspondiente en el bloque vecino 20 filtrar con respecto a la frontera de bloque 1 sobre la base del segundo valor de decisión de filtro calculado en la etapa S2.

La siguiente parte describe cómo se aplica la tercera realización de manera separada para cada línea (fila o columna) que cruza la frontera de bloque 1. En este ejemplo el primer valor de decisión de filtro se define como $d_{pi} = |p2_i - 2p1_i + p0_i|$, y el segundo valor de decisión de filtro se define como $d_{qi} = |q2_i - 2q1_i + q0_i|$. El método comprende entonces:

- calcular d_{pi} , calcular d_{qi} para cada línea i que cruza la frontera de bloque.
- si $d_{pi} < thr1$
 - efectuar un filtrado normal de la línea i del bloque 10 actual, por ejemplo filtrar y modificar dos píxeles del límite o frontera de bloque;
- si no, es decir, si $d_{pi} \geq thr1$
 - no filtrar el segundo píxel del límite o frontera de bloque i en el bloque 10 actual o no filtrar ningún píxel en absoluto en la línea i del bloque actual 10;
- si $d_{qi} < thr2$

- efectuar un filtrado normal de la línea i del bloque vecino 20 actual, por ejemplo filtrar y modificar dos píxeles del límite o la frontera de bloque 1;
- si no, es decir, si $d_{qi} \geq \text{thr2}$
 - no filtrar el segundo píxel del límite o frontera de bloque 1 de la línea i en el bloque vecino 20 actual o no filtrar ningún píxel en absoluto en la línea i del bloque vecino 20;

Como se ilustra mediante el ejemplo anterior, la tercera realización del método en la Fig. 1 puede calcular valores de decisión de filtro primero y segundo separados y por lo tanto realizar determinaciones separadas de cuántos píxeles filtrar para cada fila o columna en el bloque 10 y el bloque vecino 20 con respecto a la frontera de bloque 1. Así, en esta tercera realización, los valores de decisión de filtro primero y segundo son valores de decisión específicos para una línea, es decir, calculados para cada línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 y para cada línea 22 de píxeles 21, 23, 25, 27 correspondiente en el bloque vecino 20.

En la primera realización, se utilizan valores de decisión de filtro específicos para un bloque. Así, en tal caso puede calcularse un único primer valor de decisión de filtro para el bloque 10 con respecto a la frontera de bloque 1, y aplicarlo a todas las líneas 12 de píxeles 11, 13, 15, 17 en el bloque 10 con respecto a la frontera de bloque 1 particular. De manera correspondiente, se calcula un único segundo valor de decisión de filtro para el bloque vecino 20 con respecto a la frontera de bloque 1, y se aplica a todas las líneas 22 de píxeles 21, 23, 25, 27 correspondientes en el bloque vecino 20 con respecto a la frontera de bloque 1 particular.

Un primer ejemplo de esta primera realización implica calcular un primer valor de decisión de filtro como $|p_{2_2} - 2p_{1_2} + p_{0_2}| + |p_{2_5} - 2p_{1_5} + p_{0_5}|$, donde p_{0_2} denota el valor de píxel del píxel más cercano, en la primera línea de píxeles, a la frontera de bloque 1, p_{1_2} denota el valor de píxel del píxel siguiente más cercano, en la primera línea de píxeles, a la frontera de bloque 1, p_{2_2} denota el valor de píxel del píxel segundo siguiente más cercano, en la primera línea de píxeles, a la frontera de bloque 1, p_{0_5} denota un valor de píxel de un píxel más cercano, en una segunda línea de píxeles en el bloque 10, a la frontera de bloque 1, p_{1_5} denota un valor de píxel de un píxel siguiente más cercano, en la segunda línea de píxeles, a la frontera de bloque 1 y p_{2_5} denota un valor de píxel de un píxel segundo siguiente más cercano, en la segunda línea de píxeles, a la frontera de bloque 1.

El segundo valor de decisión de filtro se calcula entonces preferiblemente como $|q_{2_2} - 2q_{1_2} + q_{0_2}| + |q_{2_5} - 2q_{1_5} + q_{0_5}|$, donde q_{0_2} denota el valor de píxel del píxel en el bloque vecino 20 más cercano, en la correspondiente primera línea de píxeles, a la frontera de bloque 1, q_{1_2} denota el valor de píxel del píxel en el bloque vecino 20 siguiente más cercano, en la primera línea de píxeles, a la frontera de bloque 1, q_{2_2} denota el valor de píxel del píxel en el bloque vecino 20 segundo siguiente más cercano, en la correspondiente primera línea de píxeles, a la frontera de bloque 1, q_{0_5} denota un valor de píxel de un píxel en el bloque vecino 20 más cercano, en una correspondiente segunda línea de píxeles en el bloque vecino 20, a la frontera de bloque 1, q_{1_5} denota un valor de píxel de un píxel del bloque vecino 20 siguiente más cercano, en la correspondiente segunda línea de píxeles, a la frontera de bloque 1 y q_{2_5} denota un valor de píxel de un píxel en el bloque vecino 20 segundo siguiente más cercano, en la correspondiente segunda línea de píxeles, a la frontera de bloque 1.

El primer valor de decisión de filtro se utiliza entonces para todas las líneas 12 de píxeles 11, 13, 15, 17 en el bloque 10 cuando se determina cuántos píxeles filtrar, y el segundo valor de decisión de filtro se utiliza para todas las líneas 22 de píxeles 21, 23, 25, 27 correspondientes en el bloque vecino 20 cuando se determina cuántos píxeles filtrar.

En este primer ejemplo de la primera realización, la primera línea de píxeles corresponde a la línea $i = 2$, y la correspondiente primera línea corresponde a la correspondientes línea $i = 2$ y la segunda línea de píxeles corresponde a la línea $i = 5$ y la segunda correspondiente línea corresponde a la correspondiente línea $i = 5$. En este caso el bloque 10 preferiblemente comprende ocho líneas y el bloque vecino 20 preferiblemente también comprende ocho líneas, es decir $i = 0-7$.

La siguiente parte ilustra un ejemplo de implementación de la primera realización. En este ejemplo de implementación el primer valor de decisión de filtro se define como $d_p = |p_{2_2} - 2p_{1_2} + p_{0_2}| + |p_{2_5} - 2p_{1_5} + p_{0_5}|$ y el segundo valor de decisión de filtro se define como $d_q = |q_{2_2} - 2q_{1_2} + q_{0_2}| + |q_{2_5} - 2q_{1_5} + q_{0_5}|$.

- calcular d_p , calcular d_q ;
- si $d_p < \text{thr1}$
 - efectuar un filtrado normal del bloque 10 actual, por ejemplo filtrar y modificar dos píxeles del límite o frontera de bloque 1;
- si no, es decir, si $d_p \geq \text{thr1}$

- no filtrar el segundo píxel del límite o frontera de bloque 1 o no filtrar ningún píxel en absoluto;
- si $d_q < thr2$
 - efectuar un filtrado normal del bloque vecino 20 actual, por ejemplo filtrar y modificar dos píxeles del límite o la frontera de bloque 1;
- 5 • si no, es decir, si $d_q \geq thr2$
 - no filtrar el segundo píxel del límite o frontera de bloque 1 o no filtrar ningún píxel en absoluto.

En un segundo ejemplo de la primera realización, el primer valor de decisión de filtro se calcula como un valor de decisión de filtro específico para un bloque sobre la base de los valores de píxeles en la línea $i = 3$ y la línea $i = 4$ en lugar de la línea $i = 2$ y la línea $i = 5$. Las líneas $i = 3$ e $i = 4$ correspondientes en el bloque vecino 20 son entonces utilizadas preferiblemente para calcular el segundo valor de decisión de filtro. El primer valor de decisión de filtro podría entonces calcularse como $|p_{23} - 2p_{13} + p_{03}| + |p_{24} - 2p_{14} + p_{04}|$ y el segundo valor de decisión de filtro se calcula como $|q_{23} - 2q_{13} + q_{03}| + |q_{24} - 2q_{14} + q_{04}|$, donde p_{03} denota el valor de píxel del píxel más cercano, en la primera línea de píxeles, a la frontera de bloque 1, p_{13} denota el valor de píxel del píxel siguiente más cercano, en la primera línea de píxeles, a la frontera de bloque 1, p_{23} denota el valor de píxel del píxel segundo siguiente más cercano, en la primera línea de píxeles, a la frontera de bloque 1, p_{04} denota un valor de píxel de un píxel más cercano, en una segunda línea de píxeles en el bloque 10, a la frontera de bloque 1, p_{14} denota un valor de píxel de un píxel siguiente más cercano, en la segunda línea de píxeles, a la frontera de bloque 1 y p_{24} denota un valor de píxel de un píxel segundo siguiente más cercano, en la segunda línea de píxeles, a la frontera de bloque 1, y q_{03} denota el valor de píxel del píxel en el bloque vecino 20 más cercano, en la correspondiente primera línea de píxeles, a la frontera de bloque 1, q_{13} denota el valor de píxel del píxel del bloque vecino 20 siguiente más cercano, en la correspondiente primera línea de píxeles, a la frontera de bloque 1, q_{23} denota el valor de píxel del píxel en el bloque vecino 20 segundo siguiente más cercano, en la correspondiente primera línea de píxeles, a la frontera de bloque 1, q_{04} denota un valor de píxel de un píxel en el bloque vecino 20 más cercano, en la correspondiente segunda línea de píxeles en el bloque vecino 20, a la frontera de bloque 1, q_{14} denota un valor de píxel de un píxel del bloque vecino 20 siguiente más cercano, en la correspondiente segunda línea de píxeles, a la frontera de bloque 1 y q_{24} denota un valor de píxel de un píxel en el bloque vecino 20 segundo siguiente más cercano, en la correspondiente segunda línea de píxeles, a la frontera de bloque 1.

En la segunda realización, un primer valor de decisión de filtro y un segundo valor de decisión de filtro se calculan para un grupo de cuatro líneas de píxeles y cuatro líneas de píxeles correspondientes. Esta segunda realización puede ser adecuada si el bloque y el bloque vecino tienen cada uno un tamaño de 4×4 píxeles. Además, la segunda realización podría utilizarse también para bloques de píxeles más grandes, tal como 8×8 píxeles. En el último caso, un par de decisiones de filtro se calculan para las primeras cuatro líneas de píxeles y las primeras cuatro líneas de píxeles correspondientes, y otro par de decisiones de filtro se calculan para las restantes líneas de píxeles y las restantes cuatro líneas de píxeles correspondientes.

Un primer ejemplo de la segunda realización calcula el primer valor de decisión de filtro como $|p_{20} - 2p_{10} + p_{00}| + |p_{23} - 2p_{13} + p_{03}|$ y el segundo valor de decisión de filtro se calcula como $|q_{20} - 2q_{10} + q_{00}| + |q_{23} - 2q_{13} + q_{03}|$. En tal caso, la línea de píxeles y la línea de píxeles correspondiente podrían ir desde el número de línea $i = 0$ al número de línea $i = 3$. Para bloques de píxeles más grandes, tal como $i = 0-7$, como se muestra en las Figs. 2A y 2B, el primer par de un valor de decisión de filtro y un segundo valor de decisión de filtro se calcula como $|p_{20} - 2p_{10} + p_{00}| + |p_{23} - 2p_{13} + p_{03}|$ y $|q_{20} - 2q_{10} + q_{00}| + |q_{23} - 2q_{13} + q_{03}|$. Este primer par de valores de decisión de filtro es aplicable a las primeras cuatro líneas de píxeles y a las primeras cuatro líneas de píxeles correspondientes, es decir, $i = 0-3$. El segundo par de un valor de decisión de filtro y un segundo valor de decisión de filtro se calcula entonces como $|p_{24} - 2p_{14} + p_{04}| + |p_{27} - 2p_{17} + p_{07}|$ y $|q_{24} - 2q_{14} + q_{04}| + |q_{27} - 2q_{17} + q_{07}|$. El segundo par de valores de decisión de filtro es entonces aplicable a las cuatro últimas líneas de píxeles y a las cuatro últimas líneas de píxeles correspondientes, es decir, $i = 4-7$.

Un segundo ejemplo de la segunda realización calcula el primer valor de decisión de filtro como $|p_{21} - 2p_{11} + p_{01}| + |p_{22} - 2p_{12} + p_{02}|$ y el segundo valor de decisión de filtro como $|q_{21} - 2q_{11} + q_{01}| + |q_{22} - 2q_{12} + q_{02}|$. En tal caso, la línea de píxeles y la línea de píxeles correspondiente podrían ir desde el número de línea $i = 0$ al número de línea $i = 3$. Para bloques de píxeles más grandes, tal como $i = 0-7$ como se muestra en las Figs. 2A y 2B, el primer par de un primer valor de decisión de filtro y un segundo valor de decisión de filtro se calcula como $|p_{21} - 2p_{11} + p_{01}| + |p_{22} - 2p_{12} + p_{02}|$ y $|q_{21} - 2q_{11} + q_{01}| + |q_{22} - 2q_{12} + q_{02}|$. Este primer par de valores de decisión de filtro es aplicable a las

primeras cuatro líneas de píxeles y las primeras cuatro líneas de píxeles correspondientes, es decir, $i = 0-3$. El segundo par de un primer valor de decisión de filtro y un segundo valor de decisión de filtro se calcula entonces como $|p_{2_5} - 2p_{1_5} + p_{0_5}| + |p_{2_6} - 2p_{1_6} + p_{0_6}|$ y $|q_{2_5} - 2q_{1_5} + q_{0_5}| + |q_{2_6} - 2q_{1_6} + q_{0_6}|$. El segundo par de valores de decisión de filtro es entonces aplicable a las cuatro últimas líneas de píxeles y a las cuatro últimas líneas de píxeles correspondientes, es decir, $i = 4-7$.

Este concepto de la segunda realización puede extenderse al caso en el que el primer valor de decisión de filtro se calcula sobre la base de los valores de píxel de los píxeles presentes en un subconjunto de las líneas de píxeles en el bloque y el segundo valor de decisión de filtro se calcula sobre la base de los valores de píxel de los píxeles presentes en un subconjunto de las líneas de píxeles correspondientes en el bloque vecino. Así, en este concepto general de la segunda realización el primer valor de umbral de filtro podría calcularse como $|p_{2_i} - 2p_{1_i} + p_{0_i}| + |p_{2_j} - 2p_{1_j} + p_{0_j}|$, donde i, j representan diferentes números de línea en el intervalo 0 a $N-1$, denotando N el número total de líneas de píxeles en el bloque y el bloque vecino e $i \neq j$. El segundo valor de decisión de filtro se calcula entonces preferiblemente como $|q_{2_j} - 2q_{1_j} + q_{0_j}| + |q_{2_i} - 2q_{1_i} + q_{0_i}|$. Este concepto puede, por supuesto, extenderse con el subconjunto que contiene más de dos líneas de píxeles o las líneas de píxeles correspondientes.

En un ejemplo relacionado de las realizaciones primera y tercera, el primer valor de decisión de filtro se calcula como $\overline{\omega_i}|p_{2_i} - 2p_{1_i} + p_{0_i}| + \overline{\omega_j}|p_{2_j} - 2p_{1_j} + p_{0_j}|$ y el segundo valor de decisión de filtro se calcula como $\overline{\omega_i}|q_{2_i} - 2q_{1_i} + q_{0_i}| + \overline{\omega_j}|q_{2_j} - 2q_{1_j} + q_{0_j}|$. $\overline{\omega_i}, \overline{\omega_j}$ representan diferentes pesos específicos para una línea. Este concepto puede extenderse asimismo al caso con más de dos líneas de píxeles y dos líneas de píxeles correspondientes. En un ejemplo particular, a una línea de píxeles o línea de píxeles correspondiente que está más cerca del medio del bloque o del bloque vecino podría asignársele un peso comparativamente mayor en comparación con una línea de píxeles o línea de píxeles correspondiente que está más cerca de uno de los bordes del bloque o del bloque vecino.

En una cuarta realización, se utiliza una combinación de valores de decisión de filtro específicos para un bloque y específicos para una línea para determinar cuántos píxeles filtrar para las líneas de píxeles en el bloque y las líneas de píxeles correspondientes en el bloque vecino. La Fig. 3 ilustra esquemáticamente tal realización. El método se inicia en la etapa S10, en la que un tercer valor de decisión de filtro se calcula como $|p_{2_2} - 2p_{1_2} + p_{0_2}| + |p_{2_5} - 2p_{1_5} + p_{0_5}|$, donde p_{0_2} denota el valor de píxel del píxel más cercano, en una segunda línea de píxeles en el bloque 10, a la frontera de bloque 1, p_{1_2} denota el valor de píxel del píxel siguiente más cercano, en la segunda línea de píxeles, a la frontera de bloque 1, p_{2_2} denota el valor de píxel del píxel segundo siguiente más cercano, en la segunda línea de píxeles, a la frontera de bloque 1, p_{0_5} denota un valor de píxel de un píxel más cercano a, en una tercera línea de píxeles en el bloque 10, a la frontera de bloque 1, p_{1_5} denota un valor de píxel de un píxel siguiente más cercano, en la tercera línea de píxeles, a la frontera de bloque 1 y p_{2_5} denota un valor de píxel de un píxel segundo siguiente más cercano, en la tercera línea de píxeles, a la frontera de bloque 1. La segunda línea de píxeles preferiblemente corresponde al número de línea 2 en el bloque 10, y la tercera línea de píxeles preferiblemente corresponde al número de línea 5 en el bloque 10, véanse las Figs. 2A y 2B.

La siguiente etapa S11 calcula un cuarto valor de decisión de filtro como $|q_{2_2} - 2q_{1_2} + q_{0_2}| + |q_{2_5} - 2q_{1_5} + q_{0_5}|$, donde q_{0_2} denota el valor de píxel del píxel en el bloque vecino 20 más cercano, en una segunda línea de píxeles correspondiente en el bloque vecino 20, a la frontera de bloque 1, q_{1_2} denota el valor de píxel del píxel en el bloque vecino 20 siguiente más cercano, en la correspondiente segunda línea de píxeles, a la frontera de bloque 1, q_{2_2} denota el valor de píxel del píxel en el bloque vecino 20 segundo siguiente más cercano, en la correspondiente segunda línea de píxeles, a la frontera de bloque 1, q_{0_5} denota un valor de píxel de un píxel en el bloque vecino 20 más cercano a, en una correspondiente tercera línea de píxeles en el bloque vecino 20, la frontera de bloque 1, q_{1_5} denota un valor de píxel de un píxel en el bloque vecino 20 siguiente más cercano, en la correspondiente tercera línea de píxeles, a la frontera de bloque 20 y q_{2_5} denota un valor de píxel de un píxel en el bloque vecino 20 segundo siguiente más cercano, en la correspondiente tercera línea de píxeles, a la frontera de bloque 1. La segunda correspondiente línea de píxeles preferiblemente corresponde al número de línea 2 en el bloque vecino 20, y la tercera correspondiente línea de píxeles del píxel preferiblemente corresponde al número de línea 5 en el bloque vecino 20, véanse las Figs. 2A y 2B.

Las etapas S10 y S11 pueden ser realizadas en serie en cualquier orden o al menos parcialmente en paralelo.

La siguiente etapa S12 compara el tercer valor de decisión de filtro calculado en la etapa S10 con un tercer valor de umbral (T_3). Si el tercer valor de decisión de filtro es menor que el tercer valor de umbral, el método continúa hacia las etapas S1 y después a la S3 de la Fig. 1. Así, en tal caso se calcula un valor de decisión de filtro específico para una línea o primero respectivo para cada línea i en el bloque 10, donde i preferiblemente va de 0 a 7. Este primer valor de decisión de filtro se calcula entonces como $|p_{2_i} - 2p_{1_i} + p_{0_i}|$ en la etapa S1 de la Fig. 1. La etapa S3 de la Fig. 1 determina cuántos píxeles en la línea i de píxeles en el filtro 10 filtrar con respecto a frontera de bloque 1

sobre la base del primer valor de decisión de filtro calculado para la línea i de píxeles en la etapa S1. Este procedimiento se lleva a cabo para cada línea de píxeles en el bloque 10. Así, con un bloque 10 como el ilustrado en las Figs. 2A y 2B, las etapas S1 y S3 se realizarán ocho veces. El método continúa entonces hacia la etapa S13 de la Fig. 3. De manera correspondiente, si el tercer valor de decisión de filtro no es menor que el primer valor de umbral en la etapa S12, el método continúa hacia la etapa S13.

La etapa S13 compara el cuarto valor de decisión de filtro calculado en la etapa S11 con un cuarto valor de umbral (T_4). Si el cuarto valor de decisión de filtro es menor que el cuarto umbral, el método continúa hacia las etapas S2 y S4 de la Fig. 1. Un valor de decisión de filtro específico para una línea respectiva o segundo se calcula para cada línea i correspondiente en el bloque vecino 20 como $|q2_i - 2q1_i + q0_i|$ en la etapa S2 de la Fig. 1. La etapa S4 de la Fig. 3 determina cuántos píxeles en la correspondiente línea i de píxeles en el bloque vecino 20 filtrar con respecto a la frontera de bloque 1 sobre la base del segundo valor de decisión de filtro calculado para la correspondiente línea i de píxeles en la etapa S2. Este procedimiento se lleva a cabo para cada línea de píxeles correspondiente en el bloque vecino 20. El método finaliza entonces. De manera correspondiente, si el cuarto valor de decisión de filtro no es menor que el segundo valor de umbral en la etapa S13, el método finaliza.

El bucle formado por las etapas S12, S1 y S3 puede ser realizado secuencialmente en cualquier orden con respecto al bucle formado por las etapas S13, S2 y S4 o al menos parcialmente en paralelo.

En un ejemplo de esta cuarta realización se utiliza una combinación de decisiones de filtro asimétricas basadas en un bloque y basadas en una línea. En este ejemplo, el tercer valor de decisión de filtro se calcula como $d_p = |p2_2 - 2p1_2 + p0_2| + |p2_5 - 2p1_5 + p0_5|$ y el cuarto valor de decisión de filtro se calcula como $d_q = |q2_2 - 2q1_2 + q0_2| + |q2_5 - 2q1_5 + q0_5|$. Los valores de decisión de filtro específicos para una línea, es decir, los valores de decisión de filtro primero y segundo, se calculan como $d_{pi} = |p2_i - 2p1_i + p0_i|$ y $d_{qi} = |q2_i - 2q1_i + q0_i|$ para una línea y para el correspondiente número de línea i , respectivamente.

- Calcular d_p , calcular d_q ;
- si $d_p < thr1$
- para cada línea i
 - calcular d_{pi} para la línea i
 - si $d_{pi} < thr1$
 - efectuar un filtrado normal de la línea i del bloque 10 actual, por ejemplo dos píxeles del límite o la frontera de bloque 1;
 - si no, es decir, si $d_{pi} \geq thr1$
 - no filtrar el segundo píxel del límite o frontera de bloque 1 de la línea i en el bloque 10 actual o no filtrar ningún píxel en absoluto en la línea i del bloque actual 10;
- si no, es decir, si $d_p \geq thr1$
 - no filtrar el segundo píxel del límite o frontera de bloque 1 o no filtrar ningún píxel en absoluto;
- si $d_q < thr1$
- para cada línea i
 - calcular d_{qi} para la línea i
 - si $d_{qi} < thr2$
 - efectuar un filtrado normal de la línea i del bloque vecino 20, por ejemplo dos píxeles del límite o la frontera de bloque 1;
 - si no, es decir, si $d_{qi} \geq thr2$
 - no filtrar el segundo píxel del límite o frontera de bloque 1 de la línea i en el bloque vecino 20 o no filtrar ningún píxel en absoluto en la línea i del bloque vecino 20;
- si no, es decir, si $d_q \geq thr2$

- o no filtrar el segundo píxel del límite o frontera de bloque 1 o no filtrar ningún píxel en absoluto.

En el ejemplo descrito anteriormente, el mismo valor de umbral ha sido utilizado cuando se compara el tercer valor de decisión de filtro y los primeros valores de decisión de filtro, es decir, $thr1$, y el mismo valor de umbral ha sido utilizado cuando se compara el cuarto valor de decisión de filtro y los segundos valores de decisión de filtro, es decir, $thr2$. En un planteamiento alternativo, se utiliza un tercer valor de umbral para el tercer valor de decisión de filtro, se utiliza un primer valor de umbral para el primer valor de decisión de filtro, se utiliza un cuarto valor de umbral para el cuarto valor de decisión de filtro y se utiliza un segundo valor de umbral para los segundos valores de umbral. En una realización particular, los valores de umbral tercero y cuarto son iguales y los valores de umbral primero y segundo son iguales.

La Fig. 4 es un diagrama de flujo que ilustra una realización particular de las etapas de determinación S3 y S4 de la Fig. 1. El método continúa desde la etapa S2 en la Fig. 1. Una etapa S20 siguiente compara el primer valor de decisión de filtro (d_p) calculado en la etapa S1 en la Fig. 1 con un primer valor de umbral (T_1). Si el primer valor de decisión de filtro es menor que el primer valor de umbral el método continúa desde la etapa S20 hacia la etapa S21. La etapa S21 determina filtrar dos píxeles en la línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 con respecto a la frontera de bloque 1. Estos dos píxeles son preferiblemente el píxel 11 más cercano a la frontera de bloque 1 y el píxel 13 siguiente más cercano a la frontera de bloque 1 en la línea 12 de píxeles 11, 13, 15, 17. No obstante, si el primer valor de decisión de filtro no es menor que el primer umbral en la etapa S20, el método continúa por el contrario hacia la etapa S22. Una primera realización de la etapa S22 determina filtrar un píxel en la línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 con respecto a la frontera de bloque 1. Este píxel 11 es preferiblemente el píxel 11 más cercano a la frontera de bloque 1 en la línea 12 de píxeles 11, 13, 15, 17. Una segunda realización de la etapa S22 determina no filtrar ningún píxel en la línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 con respecto a la frontera de bloque 1.

Las etapas S23 a S25 efectúan la correspondiente determinación para la correspondiente línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20. Así, la etapa S23 compara el segundo valor de decisión de filtro (d_q) calculado en la etapa S2 en la Fig. 1 con un segundo valor de umbral (T_2). Si el segundo valor de decisión de filtro es menor que el segundo valor de umbral, el método continúa hacia la etapa S24. La etapa S24 determina filtrar dos píxeles en la correspondiente línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 con respecto a la frontera de bloque 1. Estos dos píxeles 21, 23 son preferiblemente el píxel 21 más cercano a la frontera de bloque 1 y el píxel 23 siguiente más cercano a la frontera de bloque 1 en la correspondiente línea 22 de píxeles 21, 23, 25, 27. Si el segundo valor de decisión de filtro no es menor que el segundo umbral, el método continúa por el contrario hacia la etapa S25 desde la etapa S23. Una primera realización de la etapa S25 determina filtrar un píxel en la correspondiente línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 con respecto a la frontera de bloque 1. Este píxel 21 es preferiblemente el píxel 21 más cercano a la frontera de bloque 1 en la correspondiente línea 22 de píxeles 21, 23, 25, 27. Una segunda realización de la etapa S25 determina no filtrar ningún píxel en la correspondiente línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino con respecto a la frontera de bloque 1.

Las etapas S20, S21 y S22 pueden ser realizadas antes de, a continuación de o al menos en paralelo a las etapas S23, S24 y S25.

Este concepto puede extenderse utilizando más de un valor de umbral por valor de decisión de filtro. Por ejemplo, si $d_p < T_1$, se filtran dos píxeles en la línea 12 de píxeles 11, 13, 15, 17, si $T_1 \leq d_p < T_1'$ se filtran un píxel en la línea 12 de píxeles 11, 13, 15, 17 y si $d_p \geq T_1'$, no se filtra ningún píxel en la línea 12 de píxeles 11, 13, 15, 17. En este caso, $T_1 < T_1'$. De manera correspondiente, si $d_q < T_2$, se filtran dos píxeles en la correspondiente línea 22 de píxeles 21, 23, 25, 27, si $T_2 \leq d_q < T_2'$ se filtran un píxel en la correspondiente línea 22 de píxeles 21, 23, 25, 27 y si $d_q \geq T_2'$, no se filtra ningún píxel en la correspondiente línea 22 de píxeles 21, 23, 25, 27. En este caso, $T_2 < T_2'$.

Así, en un aspecto general, cuanto más cerca esté el valor de decisión primero o segundo de cero mayor filtrado se aplica a la línea particular o a la línea correspondiente de píxeles filtrando y posiblemente modificando más píxeles en la línea o en la línea correspondiente de píxeles en comparación con un valor de decisión de filtro primero o segundo mayor. Esto significa que un valor de decisión de filtro primero o segundo de cero o bajo implica ninguna o pocas estructuras, en lugar de un área bastante uniforme en el fotograma de vídeo. De manera correspondiente, un valor de decisión de filtro primero o segundo alto generalmente refleja estructuras locales en el área en el fotograma de vídeo, cuyas estructuras locales no deben ser reprimidas o filtradas eliminándolas.

Esta realización disminuye la complejidad de cálculo relacionada con el filtrado de desbloqueo, puesto que el filtrado del segundo píxel del límite del bloque puede producirse con menor frecuencia en comparación con la solución de HEVC de la técnica anterior.

Los valores de umbral explicados anteriormente y utilizados para comparar los diferentes valores de decisión de filtro dependen preferiblemente del parámetro de cuantificación (QP) asignado al bloque o al bloque vecino. La Fig. 5 ilustra esquemáticamente tal planteamiento. El método se inicia en la etapa S30 en la que el primer valor de umbral, con el que se compara el primer valor de decisión de filtro (véase la etapa S20 en la Fig. 4), se determina sobre la base de un parámetro de cuantificación asociado con el bloque 10. De manera correspondiente, la etapa S31 determina el segundo valor de umbral, con el que se compara el segundo valor de umbral (véase la etapa S23 en la

Fig. 4), basándose en un parámetro de cuantificación asociado con el bloque vecino 20 y/o con un parámetro de cuantificación asociado con el bloque 10.

Por ejemplo, T_1 y T_2 se determinan sobre la base del parámetro β , que se determina a partir del valor de QP del bloque 10 o del bloque vecino 20. En una realización particular, el parámetro β se lee de una tabla basada en el valor de QP, véase la Tabla 1 que sigue.

5

Tabla 1 – Valores de β y de QP

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7	8
QP	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
β	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32	34	36
QP	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	
β	38	40	42	44	46	48	50	52	54	56	58	60	62	64	64	64	64	64	

En realizaciones particulares $T_1 = T_2 = \beta/6$ o $T_1 = T_2 = (\beta + \beta \gg 1) \gg 3$. Como una variante más de la realización, los umbrales pueden leerse de tablas separadas, es decir, $T_1 =$ función (QP), $T_2 =$ función (QP). Asimismo, los valores de umbral tercero y cuarto mencionados anteriormente se determinan preferiblemente sobre la base del parámetro de cuantificación asociado con el bloque y con el bloque vecino, respectivamente.

10

La Fig. 6 es un diagrama de flujo que ilustra cómo puede utilizarse el control del filtrado de las realizaciones en conexión con un proceso de filtrado. El método se inicia en la etapa S40, en la que un primer desfase o valor delta Δ

$$\frac{9 \times (q0_j - p0_j) - 3 \times (q1_j - p1_j)}{16}$$

se calcula sobre la base de $\frac{9 \times (q0_j - p0_j) - 3 \times (q1_j - p1_j)}{16}$, donde $p0_j$ denota un valor de píxel de un píxel 11 más cercano a, en una línea 12 de píxeles 11, 13, 15, 17, la frontera de bloque 1, $p1_j$ denota un valor de píxel de un píxel 13 siguiente más cercano, en la línea 12 de píxeles 11, 13, 15, 17, a la frontera de bloque 1, $q0_j$ denota un valor de píxel de un píxel 21 en el bloque vecino 20 más cercano a, en la correspondiente línea 22 de píxeles 21, 23, 25, 27, la frontera de bloque 1 y $q1_j$ denota un valor de píxel de un píxel 23 del bloque vecino 20 siguiente más cercano a, en la correspondiente línea 22 de píxeles 21, 23, 25, 27, la frontera de bloque 1.

15

Este primer desfase se utiliza en la etapa S41 para modificar el valor de píxel del píxel 11 más cercano a la frontera de bloque 1 en la línea 12 de píxeles 11, 13, 15, 17 sumando el primer desfase al valor de píxel, es decir, $p0'_j = p0_j + \Delta$. La etapa S41 modifica asimismo el valor de píxel del píxel 21 más cercano a la frontera de bloque 1 en la correspondiente línea 22 de píxeles 21, 23, 25, 27 restando el primer desfase del valor de píxel, es decir, $q0'_j = q0_j - \Delta$. El método continúa entonces hacia las etapas S1 y S2 de la Fig. 1, en las que se calculan los valores de decisión de filtro primero (d_p) y segundo (d_q). Una etapa S42 siguiente compara el primer valor de decisión de filtro con un primer valor de umbral (T_1). Esta etapa S42 corresponde a la etapa S20 en la Fig. 4. Si el primer valor de decisión de filtro es menor que el umbral el método continúa hacia la etapa S43.

20

25

$$\frac{p0_j + p2_j - 2p1_j + 2\Delta}{4}$$

La etapa S43 calcula un segundo desfase o valor delta Δ_p sobre la base de $\frac{p0_j + p2_j - 2p1_j + 2\Delta}{4}$, donde $p2_j$ denota un valor de píxel de un píxel 15 segundo siguiente más cercano a, en la línea 12 de píxeles 11, 13, 15, 17, la frontera de bloque 1. El segundo desfase se utiliza entonces en la etapa S44 para modificar el valor de píxel del píxel 13 siguiente más cercano a la frontera de bloque 1 en la línea 12 de píxeles 11, 13, 15, 17 sumando el segundo desfase al valor de píxel, es decir, $p1'_j = p1_j + \Delta_p$.

30

El método continúa entonces hacia la etapa S45. El método continúa asimismo en la Fig. 6 de la etapa S42 a la etapa S45 si el primer valor de umbral no es menor que el primer umbral.

35

La etapa S45 compara el segundo valor de decisión de filtro con un segundo umbral (T_2) Esta etapa S45 corresponde a la etapa S23 de la Fig. 4. Si el segundo valor de decisión de filtro es menor que el segundo umbral, el método continúa hacia la etapa S46.

$$\frac{q0_j + q2_j - 2q1_j - 2\Delta}{4}$$

La etapa S46 calcula un tercer desfase Δ_q sobre la base de $\frac{q0_j + q2_j - 2q1_j - 2\Delta}{4}$, donde $q2_j$ denota un valor de píxel de un píxel 25 en el bloque vecino 20 segundo siguiente más cercano a, en la línea 22 correspondiente de píxeles 21, 23, 25, 27, la frontera de bloque 1. El tercer desfase se utiliza en la etapa S47 para modificar el valor de

40

píxel del píxel 23 siguiente más cercano a la frontera de bloque 1 en la línea 22 correspondiente de píxeles 21, 23, 25, 27 sumando el tercer desfase al valor de píxel, es decir, $q'_{1j} = q_{1j} + \Delta q$.

Las etapas S42, S43 y S44 pueden efectuarse en serie en cualquier orden o al menos parcialmente en paralelo con las etapas S45, S46 y S47.

5 En lo anterior, los desfases primero, segundo y tercero se calculan sobre la base de ecuaciones particulares de valores de píxeles. Esto significa que el primer desfase se calcula como función de $\frac{9 \times (q_{0j} - p_{0j}) - 3 \times (q_{1j} - p_{1j})}{16}$, el segundo desfase se calcula como función de $\frac{p_{0j} + p_{2j} - 2p_{1j} + 2\Delta}{4}$ y el

tercer desfase se calcula como función de $\frac{q_{0j} + q_{2j} - 2q_{1j} - 2\Delta}{4}$. Diferentes tales funciones son posibles y pueden utilizarse en las etapas S40, S43 y S46. Tales funciones podrían entonces definirse de manera que los cálculos de los desfases se realicen de manera eficiente en hardware. En tal caso, se prefiere generalmente no tener ninguna división y/o definir las funciones de manera que los desfases serán un valor entero. En una realización, $(X + 8) \gg 4$ se utiliza como una expresión entera de $X/16$, donde \gg denota una operación de desplazamiento lógico hacia la derecha. Así, en una realización particular, la etapa S40 calcula el primer desfase basado en o preferiblemente igual a $(9 \times (q_{0j} - p_{0j}) - 3 \times (q_{1j} - p_{1j}) + 8) \gg 4$. Representaciones enteras correspondientes de los desfases segundo y tercero podrían ser $((p_{0j} + p_{2j} + 1) \gg 1) - p_{1j} + \Delta \gg 1$ y $((q_{0j} + q_{2j} + 1) \gg 1) - q_{1j} - \Delta \gg 1$.

En un ejemplo los valores de píxeles modificados como resultado del desbloqueo se calculan a continuación. En este ejemplo, el primer valor de decisión de filtro se define como $d_p = |p_{22} - 2p_{12} + p_{02}| + |p_{25} - 2p_{15} + p_{05}|$ y el segundo valor de decisión de filtro se calcula como $d_q = |q_{22} - 2q_{12} + q_{02}| + |q_{25} - 2q_{15} + q_{05}|$.

$$\Delta = \frac{9 \times (q_0 - p_0) - 3 \times (q_1 - p_1)}{16}$$

$$p'_0 = p_0 + \Delta$$

$$q'_0 = q_0 - \Delta$$

20

si $d_p < \text{thrP}$

$$\Delta_p = \frac{p_0 + p_2 - 2p_{1j} + 2\Delta}{4}$$

$$p'_1 = p_1 + \Delta_p$$

si $d_q < \text{thrQ}$

$$\Delta q = \frac{q_0 + q_2 - 2q_1 - 2\Delta}{4}$$

$$q_1' = q_1 + \Delta q$$

Fórmulas exactas para el cálculo del ejemplo anterior en el lenguaje de programación pueden parecerse al texto siguiente. Aquí, la función Clip3 está recortando los valores de salida hasta un rango entre los dos primeros argumentos de función.

```

Int xCalcDP(Pel* piSrc, Int iOffset)
{
5  return abs (piSrc [-iOffset*3]-2*piSrc[-iOffset*2]+piSrc[-
iOffset]);
}
Int xCalcDQ(Pel* piSrc, Int iOffset)
{
return abs (piSrc[0]-2*piSrc[iOffset]+piSrc[iOffset*2]);
Int iDP=xCalcDP(piTmpSrc+iSrcStep*(iIdx*uiPelsInPart+iBlkIdx*
DEBLOCK_SMALLEST_BLOCK+2), iOffset)+xCalcDP(piTmpSrc+iSrcStep*
(iIdx*uiPelsInPart+iBlkIdx*DEBLOCK_SMALLEST_BLOCK+5), iOffset);
Int iDQ=xCalcDQ(piTmpSrc+iSrcStep*(iIdx*uiPelsInPart+iBlkIdx*
DEBLOCK_SMALLEST_BLOCK+2), iOffset)+xCalcDQ(piTmpSrc+iSrcStep*(
Idx*uiPelsInPart+iBlkIdx*DEBLOCK_SMALLEST_BLOCK+5), iOffset);
Int iSideThreshold=iBeta/6;
Bool bFilterP=(iDP<iSideThreshold);
Bool bFilterQ=(iDQ<iSideThreshold);
delta=(9*(m4-m3)-3*(m5-m2)+8)>>4;
if (abs(delta)<iThrCut)
{
Int tc2=tc>>1;
delta=Clip3(-tc, tc, delta);
piSrc[-iOffset]=Clip((m3+delta));
piSrc[0]=Clip((m4-delta));
if (bFilterP)
{
Int delta1=Clip3(-tc2,tc2, (((m1+m3+1)>>1)-m2+delta)>>1);
piSrc[-iOffset*2]=Clip((m2+delta1));
}
if (bFilterQ)
{
Int delta2=Clip3(-tc2,tc2, (((m6+m4+1)>>1)-m5-delta)>>1);
piSrc[iOffset]=Clip((m5+delta2));
}
}
}

```

La Fig. 7 es un diagrama de flujo que ilustra etapas opcionales adicionales del método de la Fig. 1. El método continúa desde la etapa S2 en la Fig. 1. Una etapa siguiente S50 compara una suma del primer valor de decisión y el segundo valor de decisión con un valor de umbral (T). Si la suma no es menor que el valor de umbral, el método

5 finaliza. Así, en tal caso no se aplica ningún filtrado en absoluto al bloque 10 y al bloque vecino 20 con respecto a la frontera de bloque 1 particular. El bloque 10 y el bloque vecino 20 comprenden entonces gran cantidad de estructuras locales, que no deben ser filtradas para eliminarlas. No obstante, si la suma es menor que el valor de umbral el método continúa hacia las etapas S3 y S4 en la Fig. 1, en las que la determinación de cuántos píxeles filtrar se lleva a cabo sobre la base del primer valor de decisión de filtro (etapa S3) o el segundo valor de decisión de filtro (etapa S4).

Esta realización tiene una ventaja de que no requiere muchos cálculos adicionales, puesto que los valores para las decisiones de filtro primera y segunda se utilizan también para decidir si filtrar la frontera de bloque en absoluto.

10 Las realizaciones descritas en esta memoria consiguen decisiones de desbloqueo asimétricas, que controlan el filtrado de desbloqueo para que se adapte a la estructura a cada lado de una frontera de bloque. Las decisiones asimétricas significan que la cantidad de filtrado aplicado a un lado de la frontera de bloque puede ser diferente de la cantidad de filtrado aplicada al otro lado de la frontera de bloque, proporcionando así una adaptación adicional a la estructura local. Esto mejora la calidad objetiva y subjetiva del vídeo.

15 La Fig. 8 es un diagrama de bloques esquemático de una realización de un dispositivo de control del filtrado 100. El dispositivo de control del filtrado 100 comprende un primer calculador de valor de decisión 110 configurado para calcular un primer valor de decisión de filtro para un bloque 10 en un fotograma de vídeo basado en al menos $|p2_i - 2p1_i + p0_i|$. El dispositivo de control del filtrado 100 comprende asimismo un segundo calculador de valor de decisión 120 configurado para calcular un segundo valor de decisión de filtro diferente para el bloque 10 sobre la base de $|q2_i - 2q1_i + q0_i|$.

20 Un primer determinador de píxel 130 o primera unidad de determinación de píxel o procesador está configurado para determinar cuántos píxeles en una línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 filtrar con respecto a una frontera de bloque 1 sobre la base del primer valor de decisión de filtro calculado mediante el primer calculador de valor de decisión 110. Un segundo determinador de píxel 140 o segunda unidad de determinación o procesador de píxel se proporciona en el dispositivo de control del filtrado 100 para determinar cuántos píxeles en una correspondiente línea 22 de píxeles 21, 23, 25, 27 en un bloque vecino 20 del fotograma de vídeo filtrar con respecto a la frontera de bloque sobre la base del segundo valor de decisión de filtro calculado por el segundo calculador de valor de decisión 120.

30 En una realización, el primer determinador de píxel 130 está configurado para determinar cuántos píxeles en la primera línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 filtrar con respecto a la frontera de bloque 1 sobre la base del primer valor de decisión de filtro calculado por el primer calculador de valor de decisión 110 para la primera línea 12 de píxeles 11, 13, 15, 17. El segundo determinador de píxel 140 de manera correspondiente determina en esta realización cuántos píxeles en la correspondiente primera línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 filtrar con respecto a la frontera de bloque 1 sobre la base del segundo valor de decisión de filtro calculado por el segundo calculador de valor de decisión 120 para la correspondiente primera línea 22 de píxeles 21, 23, 25, 27.

35 En otra realización, el primer calculador de valor de decisión 110 está configurado para calcular el primer valor de decisión de filtro como $|p2_2 - 2p1_2 + p0_2| + |p2_5 - 2p1_5 + p0_5|$ y el segundo calculador de valor de decisión 120 está configurado para calcular el segundo valor de decisión de filtro como $|q2_2 - 2q1_2 + q0_2| + |q2_5 - 2q1_5 + q0_5|$.

40 En otra realización más, el primer calculador de valor de decisión 110 está configurado para calcular el primer valor de decisión de filtro de un primer par como $|p2_0 - 2p1_0 + p0_0| + |p2_3 - 2p1_3 + p0_3|$ y el segundo calculador de valor de decisión 120 está configurado para calcular el segundo valor de decisión de filtro del primer par como $|q2_0 - 2q1_0 + q0_0| + |q2_3 - 2q1_3 + q0_3|$. El primer calculador de valor de decisión de filtro 110 está asimismo configurado para calcular el primer valor de decisión de filtro de un segundo par como $|p2_4 - 2p1_4 + p0_4| + |p2_7 - 2p1_7 + p0_7|$ y el segundo calculador de valor de decisión de filtro 120 está asimismo configurado para
45 calcular el segundo valor de decisión de filtro de un segundo par como $|q2_4 - 2q1_4 + q0_4| + |q2_7 - 2q1_7 + q0_7|$.

50 La Fig. 9 es un diagrama de bloques esquemático de otra realización de un dispositivo de control del filtrado 100. El dispositivo de control del filtrado 100 comprende, en esta realización y además del primer calculador de valor de decisión de filtro 110, el segundo calculador de valor de decisión 120, el primer determinador de píxel 130 y el segundo determinador de píxel 140, un tercer calculador de valor de decisión 150. El tercer calculador de valor de decisión 150 está entonces configurado para calcular un tercer valor de decisión de filtro como $|p2_2 - 2p1_2 + p0_2| + |p2_5 - 2p1_5 + p0_5|$. Un cuarto calculador de valor de decisión 160 está asimismo implementado en el dispositivo de control del filtrado 100 y configurado para calcular un cuarto valor de decisión de filtro como $|q2_2 - 2q1_2 + q0_2| + |q2_5 - 2q1_5 + q0_5|$.

En esta realización, el primer calculador de valor de decisión de filtro 110 está configurado para calcular el primer valor de decisión de filtro si el tercer valor de decisión de filtro calculado por el tercer calculador de valor de decisión 150 es inferior a un tercer valor de umbral. Si el tercer valor de decisión es menor que el tercer umbral, el primer calculador de valor de decisión 110 calcula un primer valor de umbral para cada línea i 12 de píxeles 11, 13, 15, 17 en el bloque 10 como $|p2_i - 2p1_i + p0_i|$. El primer determinador de píxel 130 determina a continuación, si el tercer valor de decisión de filtro es menor que el tercer valor de umbral y para cada línea i 12 de píxeles 11, 13, 15, 17 en el bloque 10, cuántos píxeles en la línea i 12 de píxeles 11, 13, 15, 17 en el bloque 10 filtrar con respecto a la frontera de bloque 1 sobre la base del primer valor de decisión de filtro calculado por el primer calculador de valor de decisión 110 para la línea i 12 de píxeles 11, 13, 15, 17.

El segundo calculador de valor de decisión 120 responde preferiblemente a una comparación entre el cuarto valor de decisión de filtro y un cuarto valor de umbral. Así, si el cuarto valor de decisión de filtro calculado por el cuarto calculador de valor de decisión 160 es menor que el cuarto umbral, el segundo calculador de valor de decisión 120 calcula un segundo valor de decisión de filtro como $|q2_i - 2q1_i + q0_i|$ para cada línea i 22 de píxeles 21, 23, 25, 27 correspondiente en el bloque vecino 20. El segundo determinador de píxel 240 está configurado para determinar, si el cuarto valor de decisión de filtro es menor que el cuarto valor de umbral, y para cada línea i 22 de píxeles 21, 23, 25, 27 correspondiente en el bloque vecino 20, cuántos píxeles en la correspondiente línea i 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 filtrar con respecto a la frontera de bloque 1 sobre la base del segundo valor de decisión de filtro calculado para la correspondiente línea i 22 de píxeles 21, 23, 25, 27.

La Fig. 10 es un diagrama de bloques esquemático de otra realización de un dispositivo de control del filtrado 100. Además de las unidades 110-140 de la realización ilustrada en la Fig. 8, el dispositivo de control del filtrado 100 comprende un primer comparador 180 configurado para comparar el primer valor de decisión de filtro calculado por el primer calculador de valor de decisión 110 con un primer valor de umbral. Un segundo comparador 182 está configurado de manera correspondiente para comparar el segundo valor de decisión de filtro calculado por el segundo calculador de valor de decisión 120 con un segundo valor de umbral.

En esta realización, el primer determinador de píxel 130 está configurado para determinar filtrar dos píxeles en la línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 con respecto a la frontera de bloque 1 si el primer valor de decisión de filtro es menor que el primer valor de umbral tal como determina el primer comparador 180. No obstante, si el primer valor de decisión de filtro no es menor que el primer valor de umbral, el primer determinador de píxel 130 está por el contrario configurado para determinar filtrar un píxel en la línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 con respecto a la frontera de bloque 1. De manera alternativa, el primer determinador de píxel 130 está por el contrario configurado para determinar no filtrar ningún píxel en la línea 12 de píxeles 11, 13, 15, 17 en el bloque 10 con respecto a la frontera de bloque 1.

El segundo determinador de píxel 140 está configurado para determinar filtrar dos píxeles en la correspondiente línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 con respecto a la frontera de bloque 1 si el segundo valor de decisión de filtro es menor que el segundo valor de umbral tal como determina el segundo comparador 182. No obstante, si el segundo valor de decisión de filtro no es menor que el segundo valor de umbral, el segundo determinador de píxel 140 está por el contrario configurado para determinar filtrar un píxel en la correspondiente línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 con respecto a la frontera de bloque 1. De manera alternativa, el segundo determinador de píxel 140 está por el contrario configurado para determinar no filtrar ningún píxel en la correspondiente línea 22 de píxeles 21, 23, 25, 27 en el bloque vecino 20 con respecto a la frontera de bloque 1.

En una realización el dispositivo de control del filtrado 100 de la Fig. 10 comprende un primer calculador de desfase

$$\frac{9 \times (q0_j - p0_j) - 3 \times (q1_j - p1_j)}{16}$$

configurado para calcular un primer desfase mediante $\frac{9 \times (q0_j - p0_j) - 3 \times (q1_j - p1_j)}{16}$. Un primer modificador de píxel 190 del dispositivo de control del filtrado 100 está configurado para modificar el valor de píxel del píxel 11 más cercano a la frontera de bloque 1 en la línea 12 de píxeles 11, 13, 15, 17 sumando el primer desfase al valor de píxel de este píxel 11. Un segundo modificador de valor de píxel 192 está configurado para modificar el valor de píxel del píxel 21 más cercano a la frontera de bloque 1 en la correspondiente línea 12 de píxeles 21, 23, 25, 27 restando el primer desfase del valor de píxel de este píxel 21.

Un segundo calculador de desfase 183 está preferiblemente implementado en el dispositivo de control del filtrado 100 para calcular un segundo desfase si el primer valor de decisión de filtro es menor que el primer valor de umbral según determina el primer comparador 180. El segundo desfase se calcula entonces sobre la base $\frac{p0_j + p2_j - 2p1_j + 2\Delta}{4}$

de $\frac{p0_j + p2_j - 2p1_j + 2\Delta}{4}$. Un tercer modificador de píxel 194 es operado si el primer valor de decisión de filtro es menor que el primer valor de umbral. En tal caso, el tercer modificador de píxel 194 está configurado para modificar el valor de píxel del píxel 13 siguiente más cercano a la frontera de bloque 1 en la línea 12 de píxeles 11, 13, 15, 17 sumando el segundo desfase al valor de píxel de este píxel 13.

Un tercer calculador de desfase 185 está configurado para calcular un tercer desfase mediante $\frac{q0_j + q2_j - 2q1_j - 2\Delta}{4}$

si el segundo valor de decisión de filtro es menor que el segundo valor de umbral según determina el segundo comparador 182. Si el segundo valor de decisión de filtro es menor que el segundo valor de umbral, un cuarto modificador de píxel 196 del dispositivo de control del filtrado 100 está configurado para modificar el valor de píxel del píxel 23 siguiente más cercano a la frontera de bloque 1 en la correspondiente línea 22 de píxeles 21, 23, 25, 27 sumando el tercer desfase al valor de píxel de este píxel 23.

Las realizaciones del dispositivo de control del filtrado 100 explicadas en lo anterior en conexión con las Figs. 9 y 10 comparan los valores de decisión de filtro con valores de umbral respectivos. En una realización, tales valores de umbral son calculados por el dispositivo de control del filtrado 100 para la frontera de bloque 1 particular. El dispositivo de control del filtrado 100 entonces comprende preferiblemente un determinador de umbral 170 o procesador o unidad de determinación de umbral configurado para determinar el primer valor de umbral mediante el primer comparador 180 en la Fig. 10 y el tercer valor de umbral utilizado por el dispositivo de control del filtrado 100 en la Fig. 9 sobre la base de un parámetro de cuantificación asociado con el bloque 10. De manera correspondiente, el determinador de umbral 170 de manera correspondiente preferiblemente determina el segundo valor de umbral utilizado por el segundo comparador 182 en la Fig. 10 y el cuarto valor de umbral utilizado por el dispositivo de control del filtrado 100 en la Fig. 9 sobre la base de un parámetro de cuantificación asociado con el bloque vecino 20.

La Fig. 11 es un diagrama de bloques esquemático de otra realización más de un dispositivo de control del filtrado 100. Además de las unidades 110-140 de la realización ilustrada en la Fig. 8, el dispositivo de control del filtrado 100 comprende en esta realización un tercer comparador 184 configurado para comparar una suma del primer valor de decisión de filtro y el segundo valor de decisión de filtro con un valor de umbral. Si la suma es igual o mayor que el valor de umbral, los determinadores de píxel primero y segundo 130, 140 no determinarán ningún número de píxeles para filtrar, puesto que no debe aplicarse ningún filtro al bloque 10 y al bloque vecino 20 con respecto a la frontera de bloque 1 particular. No obstante, si la suma es menor que el umbral, los determinadores de píxel primero y segundo 130, 140, son operados para determinar el número de píxeles para filtrar sobre la base de los valores de decisión primero o segundo, respectivamente.

Las realizaciones del dispositivo de control del filtrado 100 explicado anteriormente y descritas en las Figs. 8-11 pueden combinarse. Por ejemplo, el tercer calculador de valor 150 y el cuarto calculador de valor 160 de la Fig. 9 pueden ser implementados en cualquiera de las realizaciones descritas en las Figs. 10 u 11. De manera correspondiente, el tercer comparador 184 de la Fig. 11 puede ser implementado en cualquiera de las realizaciones descritas en las Figs. 9 o 10.

Aunque las respectivas unidades 110-196 descritas junto con las Figs. 8-11 han sido descritas como unidades 110-196 separadas físicamente en el dispositivo de control del filtrado 100, y todas ellas pueden ser circuitos de propósito especial, tal como ASIC (circuitos integrados específicos para una aplicación – Application Specific Integrated Circuits, en inglés). Realizaciones alternativas del dispositivo de control del filtrado 100 son posibles cuando algunas o todas las unidades 110-196 están implementadas como módulos de programa informático que se ejecutan en un procesador de propósito general. Tal realización se describe en la Fig. 12.

La Fig. 12 ilustra esquemáticamente una realización de un ordenador 70 que tiene una unidad de proceso 72, tal como un DSP (Procesador de señal digital – Digital Signal Processor, en inglés) o CPU (Unidad de proceso central – Central Processing Unit, en inglés). La unidad de proceso central 72 puede ser una sola unidad o una pluralidad de unidades para llevar a cabo diferentes etapas del método descrito en esta memoria. El ordenador 70 comprende asimismo una unidad de entrada/salida (I/O – Input/Output, en inglés) 71 para recibir fotogramas de vídeo grabados o generados o fotogramas de vídeo codificados y enviar fotogramas de vídeo codificados o datos de vídeo decodificados. La unidad I/O 71 ha sido ilustrada como una sola unidad en la Fig. 12, pero puede asimismo estar en forma de una unidad de entrada separada y una unidad de salida separada.

Además, el ordenador 70 comprende al menos un producto de programa informático 73 en forma de una memoria no volátil, por ejemplo una EEPROM (Memoria de solo lectura programable borrable eléctricamente – Electrically Erasable Programmable Read-Only Memory, en inglés), una memoria rápida o una unidad de disco. El producto de programa informático 73 comprende un programa informático 74, que comprende medios de código que, cuando corren o son ejecutados por el ordenador 70, tal como por la unidad de proceso 72, hacen que el ordenador 70 lleve a cabo las etapas del método descritas anteriormente en conexión con la Fig. 1. Por ello, en una realización, el medio de código en el programa informático 74 comprende un primer módulo de cálculo de valor de decisión (DVC – Decision Value Calculating, en inglés) 310 para calcular el primer valor de decisión de filtro para un bloque, un segundo módulo de cálculo de valor de píxel 320 para calcular el segundo valor de decisión de filtro para el bloque, un primer módulo de determinación de píxel (PD – Pixel Determining, en inglés) 330 para determinar cuántos píxeles en la línea 12 de píxeles 11, 13, 15, 17 filtrar y un segundo módulo de determinación de píxel 340 para determinar cuántos píxeles en la correspondiente línea 22 de píxeles 21, 23, 25, 27 filtrar. Estos módulos 310-340 esencialmente llevan a cabo las etapas del diagrama de flujo en la Fig. 1 cuando se ejecutan sobre la unidad de

proceso 72. Así, cuando los diferentes módulos 310-340 son ejecutados en la unidad de proceso 72 corresponden a las correspondientes unidades 110-140 de las Figs. 8-11.

5 El programa informático 74 puede comprender adicionalmente un tercer módulo de cálculo de valor de decisión, un cuarto módulo de cálculo de valor de decisión, un módulo de determinación de umbral, un primer módulo comparador, segundo módulo comparador, un tercer módulo comparador, un primer módulo de cálculo de desfase, un segundo módulo de cálculo de desfase, un tercer módulo de cálculo de desfase, un primer módulo de modificación de píxel para realizar la operación de las correspondientes unidades 150-196 en las Figs. 9-11.

El ordenador 70 de la Fig. 12 puede ser un equipo de usuario o encontrarse en un equipo de usuario. En tal caso, el equipo de usuario puede comprender adicionalmente o estar conectado a una pantalla para mostrar datos de vídeo.

10 El dispositivo de control del filtrado de las Figs. 8-11 se utiliza preferiblemente en la codificación de vídeo. Funciona y está por lo tanto implementado preferiblemente tanto en un codificador de vídeo como en un decodificador de vídeo. El decodificador de vídeo puede ser implementado preferiblemente en hardware pero también en software. Lo mismo ocurre para el codificador de vídeo.

15 La Fig. 13 es un diagrama de bloques esquemático de un codificador 40 para codificar un bloque de píxeles en un fotograma de vídeo de una secuencia de vídeo de acuerdo con una realización.

20 Un bloque actual de píxeles se predice realizando una estimación de movimiento mediante un estimador de movimiento 50 a partir de un bloque de píxeles ya proporcionado en el mismo fotograma o en un fotograma previo. El resultado de la estimación de movimiento es un vector de movimiento o de desplazamiento asociado con el bloque de referencia, en el caso de inter predicción. El vector de movimiento es utilizado por un compensador de movimiento 50 para obtener una inter predicción del bloque de píxeles.

25 Un intra pronosticador 49 calcula una intra predicción del bloque actual de píxeles. Las salidas del estimador/compensador de movimiento 50 y el intra pronosticador 49 son introducidas en un selector 51, que selecciona una intra predicción o una inter predicción para el bloque de píxeles actual. La salida del selector 51 es introducida en un calculador de error en forma de sumador 41 que recibe también los valores de píxeles del bloque de píxeles actual. El sumador 41 calcula y produce un error residual como la diferencia en los valores de píxeles entre el bloque de píxeles y su predicción.

30 El error es transformado en un transformador 42, tal como mediante una transformada de coseno discreta, y cuantificado mediante un cuantificador 43 seguido por la codificación en un codificador 44, tal como mediante un codificador de entropía. En inter codificación, también el vector de movimiento estimado es llevado al codificador 44 para generar la representación codificada del bloque actual de píxeles.

35 El error residual transformado y cuantificado para el bloque de píxeles actual se proporciona también a un cuantificador inverso 45 y a un transformador inverso 46 para obtener el error residual original. Este error es sumado mediante un sumador 47 a la salida de predicción de bloque del compensador de movimiento 50 o al intra pronosticador 49 para crear un bloque de píxeles de referencia que puede ser utilizado en la predicción y la codificación de un bloque de píxeles siguiente. Este nuevo bloque de referencia es procesado primero mediante un dispositivo de control del filtrado 100 de acuerdo con las realizaciones para controlar cualquier filtrado de desbloqueo que se aplique al bloque de referencia para combatir cualquier aberración de bloqueo. El nuevo bloque de referencia procesado es entonces almacenado temporalmente en una memoria temporal de fotogramas 48, donde está disponible para el intra pronosticador 49 y el estimador/compensador de movimiento 50.

40 La Fig. 14 es un diagrama de bloques esquemático correspondiente de un decodificador 60 que comprende un dispositivo de control del filtrado 100 de acuerdo con las realizaciones. El decodificador 60 comprende un decodificador 61, tal como el decodificador de entropía, para decodificar una representación codificada de un bloque de píxeles para obtener un conjunto de errores residuales cuantificados y transformados. Estos errores residuales son descuantificados en un cuantificador inverso 62 y transformados inversamente mediante un transformador inverso 63 para obtener un conjunto de errores residuales.

45 Estos errores residuales son sumados en un sumador 64 a los valores de píxeles de un bloque de píxeles de referencia. El bloque de referencia se determina mediante un estimador/compensador de movimiento 67 o un intra pronosticador 66, dependiendo de si se lleva a cabo una inter o intra predicción. Un selector 68 es interconectado por ello al sumador 64 y al estimador/compensador de movimiento 67 y al intra pronosticador 66. El bloque de píxeles decodificado resultante obtenido del sumador 64 es introducido en un dispositivo de control del filtrado 100 de acuerdo con las realizaciones para controlar cualquier filtro de desbloqueo que se aplique para combatir cualquier aberración de bloqueo. El bloque de píxeles filtrado es obtenido del decodificador 60 y es además proporcionado preferiblemente de manera temporal a una memoria temporal de fotogramas 65, y puede ser utilizado como un bloque de referencia de píxeles para la decodificación de un bloque de píxeles siguiente. La memoria temporal de fotogramas 65 está por ello conectada al estimador/compensador de movimiento 67 para proporcionar los bloques de píxeles almacenados al estimador/compensador de movimiento 67.

La salida del sumador 64 es preferiblemente introducida también en el intra pronosticador 66 para ser utilizada como un bloque de píxeles de referencia no filtrado.

En las realizaciones descritas en las Figs. 13 y 14, el dispositivo de control del filtrado 100 controla el filtrado de desbloqueo en forma de un llamado filtrado en bucle. En una implementación alternativa en el decodificador 60 el dispositivo de control del filtrado 100 está dispuesto para efectuar el llamado filtrado de pos-proceso. En tal caso, el dispositivo de control del filtrado 100 opera en los fotogramas obtenidos fuera del bucle formado por el sumador 64, la memoria temporal de fotogramas 65, el intra pronosticador 66, el estimador/compensador de movimiento 67 y el selector 68. Entonces, típicamente, no se realiza ningún filtrado de desbloqueo ni control del filtrado en el codificador.

La Fig. 15 es un diagrama de bloques esquemático de un equipo de usuario o terminal de medios 80 que aloja un decodificador 60 con un dispositivo de control del filtrado. El equipo de usuario 80 puede ser cualquier dispositivo que tenga funciones de decodificación de medios que opere en un flujo de vídeo codificado o fotogramas de vídeo codificados para decodificar con ello los fotogramas de vídeo y permitir el acceso al vídeo. Ejemplos no limitativos de tales dispositivos incluyen teléfonos móviles y otros reproductores de medios portátiles, tabletas, ordenadores de sobremesa, agendas, grabadores de vídeo personales, reproductores de multimedia, servidores de vídeo en tiempo real, decodificadores, TV, ordenadores, decodificadores, consolas de juegos, etc. El equipo de usuario 80 comprende una memoria 84 configurada para almacenar fotogramas de vídeo codificados. Estos fotogramas de vídeo codificados pueden haber sido generados por el propio equipo de usuario 80. De manera alternativa, los fotogramas de vídeo codificados son generados por algún otro dispositivo y transmitidos de manera inalámbrica o transmitidos por cable al equipo de usuario 80. El equipo de usuario 80 comprende entonces un transmisor receptor (transmisor y receptor) o un puerto de entrada y salida 82 para conseguir la transferencia de datos.

Los fotogramas de vídeo codificados son llevados a la memoria 84 a un decodificador 60, tal como el decodificador ilustrado en la Fig. 14. El decodificador 60 comprende un dispositivo de control del filtrado 100 de acuerdo con las realizaciones. El decodificador 60, a continuación, decodifica los fotogramas de vídeo codificados en fotogramas de vídeo decodificados. Los fotogramas de vídeo decodificados son proporcionados a un reproductor de medios 86 que está configurado para convertir los fotogramas de vídeo decodificados en datos de vídeo que son reproducibles en un visualizador o pantalla 88 del, o que está conectada al, equipo de usuario 80.

En la Fig. 15, el equipo de usuario 80 ha sido ilustrado comprendiendo tanto el decodificador 60 como el reproductor de medios 86, estando el decodificador 60 implementado como parte del reproductor de medios 86. Esto debe, no obstante, considerarse como un ejemplo meramente ilustrativo pero no limitativo de una realización de implementación para el equipo de usuario 80. Asimismo, son posibles implementaciones distribuidas en las que el decodificador 60 y el reproductor de medios 86 son proporcionados en dos dispositivos separados físicamente y dentro del alcance del equipo de usuario 80 tal como se utiliza en esta memoria. El visualizador 88 podría asimismo proporcionarse como un dispositivo separado conectado al equipo de usuario 80, en el que tiene lugar el proceso real de los datos.

La Fig. 16 ilustra otra realización de un equipo de usuario 80 que comprende un codificador, tal como el codificador de la Fig. 13, que comprende un dispositivo de control del filtrado de acuerdo con las realizaciones. El codificador 40 es entonces configurado para codificar fotogramas de vídeo recibidos por la unidad I/O 82 y/o generados por el propio equipo de usuario 80. En el último caso, el equipo de usuario 80 preferiblemente comprende un motor o grabador de medios, tal como en forma de o conectado a (una cámara (de vídeo)). El equipo de usuario 80 puede comprender asimismo opcionalmente un reproductor de medios 86, tal como un reproductor de medios 86 con un decodificador y un dispositivo de control del filtrado de acuerdo con las realizaciones, y un visualizador 88.

Como se ilustra en la Fig. 17, el codificador 40 y/o el decodificador 60, tal como se ilustran en las Figs. 13 y 14, pueden ser implementados en un dispositivo de red 30 que se encuentra o pertenece a un nodo de red en una red de comunicación 32 entre una unidad de emisión 34 y un equipo de usuario de recepción 36. Tal dispositivo de red 30 puede ser un dispositivo para convertir vídeo de acuerdo con un estándar de codificación de vídeo en otro estándar de codificación de vídeo, por ejemplo, si se ha establecido que el equipo de usuario de recepción 36 solo sea capaz de o prefiera otro estándar de codificación de vídeo distinto del enviado desde la unidad de emisión 34. El dispositivo de red 30 puede ser en forma de o estar comprendido en una estación de base de radio, un Nodo B o cualquier otro nodo de red en una red de comunicación 32, tal como una red de radio.

Las realizaciones descritas anteriormente deben ser entendidas como ejemplos ilustrativos de la presente invención. Resultará evidente para los expertos en la materia que pueden efectuarse varias modificaciones, combinaciones y cambios a las realizaciones sin separarse del alcance de la presente invención. En particular, pueden combinarse diferentes soluciones parciales en las diferentes realizaciones en otras configuraciones, cuando sea técnicamente posible. El alcance de la presente invención está, no obstante, definido por las reivindicaciones adjuntas.

REIVINDICACIONES

1. Método para el control del filtrado aplicable a un bloque (10) de múltiples píxeles (11, 13, 15, 17) en un fotograma de vídeo, teniendo cada píxel (11, 13, 15, 17) un valor de píxel respectivo, comprendiendo el citado método:

5 calcular (S1) un primer valor de decisión de filtro para el citado bloque (10), utilizando solo píxeles del citado bloque (10), basados en:

$$|p2_i - 2p1_i + p0_i|$$

10 donde $i = 0 \dots N-1$ y N es el número de líneas de píxeles del citado bloque (10), siendo una línea de píxeles una fila o una columna de píxeles, donde $p0_i$ denota un valor de píxel de un píxel (11) más cercano a, en una primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10), una frontera de bloque (1) a un bloque vecino (20) de múltiples píxeles (21, 23, 25, 27) en el citado fotograma de vídeo, $p1_i$ denota un valor de píxel de un píxel (13) siguiente más cercano a, en la citada primera línea (12) de píxeles (11, 13, 15, 17), la citada frontera de bloque (1) y $p2_i$ denota un valor de píxel de un píxel (15) segundo siguiente más cercano a, en la citada primera línea (12) de píxeles (11, 13, 15, 17), la citada frontera de bloque (1);

15 calcular (S2) un segundo valor de decisión de filtro para el citado bloque vecino (20), utilizando solo píxeles del citado bloque vecino (20) basados en:

$$|q2_i - 2q1_i + q0_i|$$

20 donde $i = 0 \dots N-1$ y N es el número de líneas de píxeles del citado bloque vecino (20), siendo una línea de píxeles una fila o una columna de píxeles, donde $q0_i$ denota un valor de píxel de un píxel (21) en el citado bloque vecino (20) más cercano a, en una correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20), la citada frontera de bloque (1), $q1_i$ denota un valor de píxel de un píxel (23) del citado bloque vecino (20) siguiente más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1) y $q2_i$ denota un valor de píxel de un píxel (25) en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1); donde la primera línea (12) de píxeles y la correspondiente primera línea (22) en el bloque vecino (20) son líneas opuestas con respecto a la frontera de bloque (1) y perpendiculares a la frontera de bloque (1),

25 determinar (S3) cuántos píxeles en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) filtrar con respecto a la citada frontera de bloque (1) sobre la base de una comparación (S20) del citado primer valor de decisión de filtro con un primer valor de umbral; y

30 determinar (S4) cuántos píxeles en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) filtrar con respecto a la citada frontera de bloque (1) sobre la base de una comparación (S23) del citado segundo valor de decisión de filtro con un segundo valor de umbral, en el que

calcular (S1) el citado primer valor de decisión de filtro comprende calcular (S1) el citado valor de decisión de filtro como:

$$|p2_0 - 2p1_0 + p0_0| + |p2_3 - 2p1_3 + p0_3|$$

35 donde $p0_0$ denota el citado valor de píxel del citado píxel más cercano a, en una primera línea de píxeles, la citada frontera de bloque (1), $p1_0$ denota el citado valor de píxel del citado píxel siguiente más cercano, en la citada línea de píxeles, la citada frontera de bloque (1), $p2_0$ denota el citado valor de píxel del citado píxel segundo siguiente más cercano a, en la citada primera línea de píxeles, la citada frontera de bloque (1), $p0_3$ denota un valor de píxel de un píxel más cercano a, en la segunda línea de píxeles en el citado bloque (10), la citada frontera de bloque (1), $p1_3$ denota un valor de píxel de un píxel siguiente más cercano a, en la citada segunda línea de píxeles, la citada frontera de bloque (1), $p2_3$ denota un valor de píxel de un píxel segundo siguiente más cercano a, en la citada segunda línea de píxeles, la citada frontera de bloque (1), y

calcular (S2) el citado segundo valor de decisión de filtro comprende calcular (S2) el citado segundo valor de decisión de filtro como:

$$|q2_0 - 2q1_0 + q0_0| + |q2_3 - 2q1_3 + q0_3|$$

45 donde $q0_0$ denota el citado valor de píxel del citado píxel del citado bloque vecino (20) más cercano a, en la citada correspondiente primera línea de píxeles, la citada frontera de bloque (1), $q1_0$ denota el citado valor de píxel del citado píxel del citado bloque vecino (20) siguiente más cercano a, en la citada línea de píxeles correspondiente, la citada frontera de bloque (1), $q2_0$ denota el citado valor de píxel del citado píxel en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada línea de píxeles correspondiente, la citada frontera de bloque (1), $q0_3$ denota un valor de píxel de un píxel en el citado bloque vecino (20) más cercano a, en una correspondiente segunda línea de píxeles en el citado bloque vecino (20), la citada frontera de bloque (1), $q1_3$ denota un valor de píxel de un píxel del citado bloque vecino (20) siguiente más cercano a, en una correspondiente segunda línea de píxeles, la citada frontera de bloque (1) y $q2_3$ denota un valor de píxel de un píxel en el citado bloque vecino (20) siguiente más cercano a, en la citada correspondiente segunda línea de píxeles, la citada frontera de bloque (1).

2. El método de acuerdo la reivindicación 1, en el que:

la determinación (S3) de cuántos píxeles en la citada primera línea (12) de píxel (11, 13, 15, 17) en el citado bloque (10) filtrar comprende:

comparar (S20) el citado primer valor de decisión de filtro con un primer valor de umbral;

5 determinar (S21) filtrar dos píxeles en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) con respecto a la citada frontera de bloque (1) si el citado primer valor de decisión de filtro es menor que el citado primer valor de umbral; y

determinar (S22) filtrar un píxel en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) con respecto a la citada frontera de bloque (1) si el citado primer valor de decisión de filtro es igual o superior al citado primer valor de umbral; y

10 la determinación (S4) de cuántos píxeles en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) filtrar comprende:

comparar (S23) el citado segundo valor de decisión de filtro con un segundo valor de umbral;

15 determinar (S24) filtrar dos píxeles en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque (20) con respecto a la citada frontera de bloque (1) si el citado segundo valor de decisión de filtro es menor que el citado segundo valor de umbral; y

determinar (S25) filtrar un píxel en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) con respecto a la citada frontera de bloque (1) si el citado segundo valor de decisión de filtro es igual o superior al citado segundo valor de umbral.

3. El método de acuerdo con cualquiera de las reivindicaciones 1 a 2, en el que

20 la determinación (S3) de cuántos píxeles en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) para filtrar comprende:

comparar (S20) el citado primer valor de decisión de filtro con un primer valor de umbral;

25 determinar (S21) filtrar dos píxeles en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) con respecto a la citada frontera de bloque (1) si el citado primer valor de decisión de filtro es menor que el citado primer valor de umbral; y

determinar (S22) no filtrar ningún píxel en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) con respecto a la citada frontera de bloque (1) si el citado primer valor de decisión de filtro es igual o superior al citado primer valor de umbral; y

30 la determinación (S4) de cuántos píxeles en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) filtrar comprende:

comparar (S23) el citado segundo valor de decisión de filtro con un segundo valor de umbral;

determinar (S24) filtrar dos píxeles en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) con respecto a la citada frontera de bloque (1) si el citado segundo valor de decisión de filtro es menor que el citado segundo valor de umbral; y

35 determinar (S25) no filtrar ningún píxel en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque (20) con respecto a la citada frontera de bloque (1) si el citado segundo valor de decisión de filtro es igual o superior al citado segundo valor de umbral.

4. Un dispositivo de control del filtrado (100) que comprende:

40 un primer calculador de valor de decisión (110) configurado para calcular un primer valor de decisión de filtro para un bloque (10) de múltiples píxeles (11, 13, 15, 17) en un fotograma de vídeo, utilizando solo píxeles del citado bloque (10), basados en:

$$|p2_i - 2p1_i + p0_i|$$

45 donde $i = 0 \dots N-1$ y N es el número de líneas de píxeles del citado bloque (10), siendo una línea de píxeles una fila o una columna de píxeles, donde $p0_i$ denota un valor de píxel de un píxel (11) más cercano a, en una primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10), una frontera de bloque (1) a un bloque vecino (20) de múltiples píxeles (21, 23, 25, 27) en el citado fotograma de vídeo, $p1_i$ denota un valor de píxel de un píxel (13) siguiente más cercano a, en la citada primera línea (12) de píxeles (11, 13, 15, 17), la citada frontera de bloque (1) y $p2_i$ denota un valor de píxel de un píxel (15) segundo siguiente más cercano a, en la citada primera línea (12) de píxeles (11, 13, 15, 17), la citada frontera de bloque (1);

50 un segundo valor de decisión (120) configurado para calcular un segundo valor de decisión de filtro para un bloque vecino (20), utilizando solo píxeles del citado bloque vecino (20), basados en:

$$|q2_i - 2q1_i + q0_i|$$

donde $i = 0 \dots N-1$ y N es el número de líneas de píxeles del citado bloque vecino (20), siendo una línea de píxeles una fila o una columna de píxeles, donde $q0_i$ denota un valor de píxel de un píxel (21) en el citado bloque vecino (20) más cercano a, en una correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20), la citada frontera de bloque (1), $q1_i$ denota un valor de píxel de un píxel (23) del citado bloque vecino (20) siguiente más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1) y $q2_i$ denota un valor de píxel de un píxel (25) en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1); donde la primera línea (12) de píxeles y la correspondiente primera línea (22) en el bloque vecino (20) son líneas opuestas con respecto a la frontera de bloque (1) y perpendiculares a la frontera de bloque (1);

un primer determinador de píxel (130) configurado para determinar cuántos píxeles en una primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) filtrar con respecto a la citada frontera de bloque (1) basándose en el citado primer valor de decisión de filtro calculado mediante el citado primer calculador de valor de decisión (110); y

un segundo determinador de píxel (140) configurado para determinar cuántos píxeles en una correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) filtrar con respecto a la citada frontera de bloque (1) basándose en el citado segundo valor de decisión de filtro calculado mediante el citado segundo calculador de valor de decisión (120), en el que

comprende además un primer comparador (180) configurado para comparar el citado primer valor de decisión de filtro calculado por el citado primer calculador de valor de decisión (110) con un primer valor de umbral; y

un segundo comparador (182) configurado para comparar el citado segundo valor de decisión de filtro calculado por el citado segundo calculador de valor de decisión (120) con un segundo valor de umbral, en el que

el citado primer calculador de valor de decisión (110) está configurado para calcular el citado primer valor de decisión de filtro como:

$$|p2_0 - 2p1_0 + p0_0| + |p2_3 - 2p1_3 + p0_3|$$

donde $p0_0$ denota el citado valor de píxel del citado píxel más cercano a, en la citada primera línea de píxeles, la citada frontera de bloque (1), $p1_0$ denota el citado valor de píxel del citado píxel siguiente más cercano, en la citada línea de píxeles, a la citada frontera de bloque (1), $p2_0$ denota el citado valor de píxel del citado píxel segundo siguiente más cercano a, en la citada primera línea de píxeles, la citada frontera de bloque (1), $p0_3$ denota un valor de píxel de un píxel más cercano a, en una segunda línea de píxeles en el citado bloque (10), la citada frontera de bloque (1), $p1_3$ denota un valor de píxel de un píxel siguiente más cercano a, en la citada segunda línea de píxeles, la citada frontera de bloque (1), y $p2_3$ denota un valor de píxel de un píxel segundo siguiente más cercano a, en la citada segunda línea de píxeles, la citada frontera de bloque (1), y

el citado segundo calculador de valor de decisión (120) está configurado para calcular el citado segundo valor de decisión de filtro como:

$$|q2_0 - 2q1_0 + q0_0| + |q2_3 - 2q1_3 + q0_3|$$

donde $q0_0$ denota el citado valor de píxel del citado píxel del citado bloque vecino (20) más cercano a, en la citada correspondiente primera línea de píxeles, la citada frontera de bloque (1), $q1_0$ denota el citado valor de píxel del citado píxel del citado bloque vecino (20) siguiente más cercano a, en la citada línea de píxeles correspondiente, la citada frontera de bloque (1), $q2_0$ denota el citado valor de píxel del citado píxel en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada línea de píxeles correspondiente, la citada frontera de bloque (1), $q0_3$ denota un valor de píxel de un píxel en el citado bloque vecino (20) más cercano a, en una correspondiente segunda línea de píxeles en el citado bloque vecino (20),

la citada frontera de bloque (1), $q1_3$ denota un valor de píxel de un píxel en el citado bloque vecino (20) siguiente más cercano a, en la citada correspondiente segunda línea de píxeles, la citada frontera de bloque (1) y $q2_3$ denota un valor de píxel de un píxel en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada correspondiente segunda línea de píxeles, la citada frontera de bloque (1).

5. El dispositivo de acuerdo con la reivindicación 4, que comprende:

el citado primer determinador de píxel (130) está configurado para i) determinar filtrar dos píxeles en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) con respecto a la citada frontera de bloque (1) si el citado primer valor de decisión de filtro es menor que el citado primer valor de umbral según determina el primer comparador (180), e ii) determinar filtrar un píxel en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) con respecto a la citada frontera de bloque (1) si el citado primer valor de decisión de filtro es menor o igual que el citado primer valor de umbral según determina el citado primer comparador (180); y

el citado segundo determinador de píxel (140) está configurado para i) determinar filtrar dos píxeles en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) con respecto a la citada frontera de bloque (1) si el citado segundo valor de decisión de filtro es menor que el citado segundo valor de

umbral según determina el segundo comparador (182) e ii) determinar filtrar un píxel en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) con respecto a la citada frontera de bloque (1) si el citado segundo valor de decisión de filtro es mayor o igual que el citado segundo valor de umbral según determina el citado segundo comparador (182).

5 6. El dispositivo de acuerdo con cualquiera de las reivindicaciones 4 a 5, que comprende además:

un primer comparador (180) configurado para comparar el citado primer valor de decisión de filtro calculado por el primer calculador de valor de decisión (110) con un primer valor de umbral; y

un segundo comparador (182) configurado para comparar el citado segundo valor de decisión de filtro calculado por el citado segundo calculador de valor de decisión (120) con un segundo valor de umbral, en el que

10 el citado primer determinador de píxel (130) está configurado para i) determinar filtrar dos píxeles en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) con respecto a la citada frontera de bloque (1) si el citado primer valor de decisión de filtro es menor que el citado primer valor de umbral según determina el citado primer comparador (180), e ii) determinar no filtrar ningún píxel en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) con respecto a la citada frontera de bloque (1) si el citado primer valor de decisión de filtro es mayor o igual que el citado primer valor de umbral según determina el citado primer comparador (180); y

15 el citado segundo determinador de píxel (140) está configurado para i) determinar filtrar dos píxeles en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) con respecto a la citada frontera de bloque (1) si el citado segundo valor de decisión de filtro es menor que el citado segundo valor de umbral según determina el citado segundo comparador (182), e ii) determinar no filtrar ningún píxel en la citada primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) con respecto a la citada frontera de bloque (1) si el citado segundo valor de decisión de filtro es mayor o igual que el citado segundo valor de umbral según determina el citado segundo comparador (182).

25 7. El dispositivo de acuerdo con cualquiera de las reivindicaciones 4 a 6, que comprende además un determinador de umbral (170) configurado para i) determinar el citado primer valor de umbral sobre la base de un parámetro de cuantificación asociado con el citado bloque (10), e ii) determinar el citado segundo valor de umbral sobre la base de un parámetro de cuantificación asociado con el citado bloque vecino (20).

8. El dispositivo de acuerdo con cualquiera de las reivindicaciones 4 a 7, que comprende además:

un primer calculador de desfase (181) configurado para calcular un primer desfase Δ basado en

$$\frac{9 \times (q0_j - p0_j) - 3 \times (q1_j - p1_j)}{16}$$

30 , en el que $p0_j$ denota un valor de píxel de un píxel (11) cercano a, en la citada primera línea (12) de píxeles (11, 13, 15, 17), la citada frontera de bloque (1), $p1_j$ denota un valor de píxel de un píxel (13) siguiente más cercano a, en la citada primera línea (12) de píxeles (11, 13, 15, 17), la citada frontera de bloque (1), $q0_j$ denota un valor de píxel de un píxel (21) en el citado bloque vecino (20) más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1) y $q1_j$ denota un valor de píxel de un píxel (23) del citado bloque vecino (20) siguiente más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1);

35 un primer modificador de píxel (190) configurado para modificar el citado valor de píxel del citado píxel (11) más cercano a la citada frontera de bloque (1) en la citada primera línea (12) de píxeles (11, 13, 15, 17) sumando el citado primer desfase calculado mediante el citado primer calculador de desfase (181) al citado valor de píxel del citado píxel (11) más cercano a la citada frontera de bloque (1) en la citada primera línea (12) de píxeles (11, 13, 15, 17);

40 un segundo modificador de píxel (192) configurado para modificar el citado valor de píxel del citado píxel (21) en el citado bloque vecino (20) más cercano a la citada frontera de bloque (1) en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) restando el citado primer desfase calculado mediante el citado primer calculador de desfase (181) del citado valor de píxel del citado píxel (21) en el citado bloque vecino (20) más cercano a la citada frontera de bloque (1) en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27);

45 un segundo calculador de desfase (183) configurado para calcular, si el citado primer valor de decisión de filtro es menor que el citado primer valor de umbral según determina el citado primer comparador (180), un segundo desfase

$$\frac{p0_j + p2_j - 2p1_j + 2\Delta}{4}$$

basado en , en el que $p2_j$ denota un valor de píxel de un píxel (15) segundo siguiente más cercano, en la citada primera línea (12) de píxeles (11, 13, 15, 17), a la citada frontera de bloque (1);

50 un tercer modificador de píxel (194) configurado para modificar, si el citado primer valor de decisión de filtro es menor que el citado primer umbral según determina el citado primer comparador (180), el citado valor de píxel del citado píxel (13) siguiente más cercano a la citada frontera de bloque (1) en la citada primera línea (12) de píxeles (11, 13, 15, 17) sumando el citado segundo desfase calculado por el citado segundo calculador de desfase (183) al

citado valor de píxel del citado píxel (13) siguiente más cercano a la citada frontera de bloque (1) en la citada primera línea (12) de píxeles (11, 13, 15, 17);

un tercer calculador de desfase (185) configurado para calcular, si el citado segundo valor de decisión de filtro es menor que el citado segundo valor de umbral según determina el citado segundo comparador (182), un tercer

$$q0_j + q2_j - 2q1_j - 2\Delta$$

5 desfase en $\frac{4}{4}$, en el que $q2_j$ denota un valor de píxel de un píxel (25) en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1); y un cuarto modificador de píxel (196) configurado para modificar, si el citado segundo valor de decisión de filtro es menor que el citado segundo umbral según determina el citado segundo comparador (182), el citado valor de píxel del citado píxel (23) en el citado bloque vecino (20) siguiente más cercano a la citada frontera de bloque (1) en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) sumando el citado tercer desfase calculado mediante el citado tercer calculador de desfase (185) al citado valor de píxel del citado píxel (23) en el citado bloque vecino (20) siguiente más cercano a la citada frontera de bloque (1) en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27).

9. El dispositivo de acuerdo con cualquiera de las reivindicaciones 4 a 8, que comprende además un tercer comparador (184) configurado para comparar una suma del citado primer valor de decisión de filtro y el citado segundo valor de decisión de filtro con un valor de umbral, en el que el citado primer determinador de píxel (130) está configurado para determinar, si la citada suma del citado primer valor de decisión de filtro y el citado segundo valor de decisión de filtro es menor que el citado valor de umbral según determina el citado tercer comparador (184), cuántos píxeles en la citada primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) filtrar con respecto a la citada frontera de bloque (1) sobre la base del citado valor de decisión de filtro calculado por el citado primer calculador de valor de decisión (110); y

el citado segundo determinador de píxel (140) está configurado para determinar, si la citada suma del citado primer valor de decisión de filtro y el citado segundo valor de decisión de filtro es menor que el citado valor de umbral según determina el citado tercer comparador (184), cuántos píxeles en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) filtrar con respecto a la citada frontera de bloque (1) sobre la base del citado segundo valor de decisión de filtro calculado mediante el citado segundo calculador de valor de decisión (120).

10. Un codificador (40) que comprende un dispositivo de control del filtrado (100) de acuerdo con cualquiera de las reivindicaciones 4 a 9.

11. Un equipo de usuario (80) que comprende:

una memoria (84) configurada para almacenar fotogramas de vídeo; y

un codificador (40) de acuerdo con la reivindicación 10, configurado para codificar los citados fotogramas de vídeo en fotogramas de vídeo codificados, en el que la citada memoria (84) está además configurada para almacenar los citados fotogramas de vídeo codificados.

12. Un decodificador (60), que comprende un dispositivo de control del filtrado (100) de acuerdo con cualquiera de las reivindicaciones 4 a 9.

13. Un equipo de usuario (80) que comprende:

una memoria (84) configurada para almacenar fotogramas de vídeo codificados;

un decodificador (60) de acuerdo con la reivindicación 12, configurado para decodificar los citados fotogramas de vídeo codificados en fotogramas de vídeo decodificados; y

un reproductor de medios (86), configurado para convertir los citados fotogramas de vídeo decodificados en datos de vídeo reproducibles en un visualizador (88).

14. Un programa informático (74) para el control del filtrado de un bloque (10) de múltiples píxeles (11, 13, 15, 17) en un fotograma de vídeo, teniendo cada píxel (11, 13, 15, 17) un valor de píxel respectivo, comprendiendo el citado programa informático (74) medios de código que cuando son ejecutados en un ordenador (70) hacen que el ordenador (70):

calcule un primer valor de decisión de filtro para el citado bloque (10), utilizando solo píxeles del citado bloque (10), sobre la base de:

$$|p2_i - 2p1_i + p0_i|$$

50 donde $i = 0 \dots N-1$ y N es el número de líneas de píxeles del citado bloque (10), siendo una línea de píxeles una fila o una columna de píxeles, donde $p0_i$ denota un valor de píxel de un píxel (11) más cercano a, en una primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10), una frontera de bloque (1) a un bloque vecino (20) de múltiples píxeles (21, 23, 25, 27) en el citado fotograma de vídeo, $p1_i$ denota un valor de píxel de un píxel (13) siguiente más cercano a, en la citada primera línea (12) de píxeles (11, 13, 15, 17, 19), la citada frontera de bloque

(1), y p_{2i} denota un valor de píxel de un píxel (13) segundo siguiente más cercano a, en la citada primera línea (12) de píxeles (11, 13, 15, 17), la citada frontera de bloque (1);

calcular un segundo valor de decisión de filtro para el citado bloque vecino (20), utilizando solo píxeles del citado bloque vecino (20) sobre la base de:

5
$$|q_{2i} - 2q_{1i} + q_{0i}|$$

 donde $i = 0 \dots N-1$ y N es el número de líneas de píxeles del citado bloque vecino (20), siendo una línea de píxeles una fila o una columna de píxeles, donde q_{0i} denota un valor de píxel de un píxel (21) en el citado bloque vecino (20) más cercano a, en una correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20), la frontera de bloque (1), q_{1i} denota un valor de píxel de un píxel (23) del citado bloque vecino (20) siguiente más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1) y q_{2i} denota un valor de píxel de un píxel (25) en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada correspondiente primera línea (22) de píxeles (21, 23, 25, 27), la citada frontera de bloque (1); donde la primera línea (12) de píxeles y la correspondiente primera línea (22) en el bloque vecino (20) son líneas opuestas con respecto a la frontera de bloque (1) y perpendiculares a la frontera de bloque (1),

15 un primer determinador de píxel (130) configurado para determinar cuántos píxeles en una primera línea (12) de píxeles (11, 13, 15, 17) en el citado bloque (10) filtrar con respecto a la citada frontera de bloque (1) basándose en el citado primer valor de decisión de filtro calculado mediante el citado primer calculador de valor de decisión (110); y
 un segundo determinador de píxel (140) configurado para determinar cuántos píxeles en una correspondiente primera línea (22) de píxeles (21, 23, 25, 27) en el citado bloque vecino (20) filtrar con respecto a la citada frontera de bloque (1) basándose en el citado segundo valor de decisión de filtro calculado mediante el citado segundo calculador de valor de decisión (120), en el que

20 comprende además un primer comparador (180) configurado para comparar el citado primer valor de decisión de filtro calculado por el citado primer calculador de valor de decisión (110) con un primer valor de umbral; y

25 un segundo comparador (182) configurado para comparar el citado segundo valor de decisión de filtro calculado por el citado segundo calculador de valor de decisión (120) con un segundo valor de umbral, en el que

el citado primer calculador de valor de decisión (110) está configurado para calcular el citado primer valor de decisión de filtro como:

$$|p_{20} - 2p_{10} + p_{00}| + |p_{23} - 2p_{13} + p_{03}|$$

 30 donde p_{00} denota el citado valor de píxel del citado píxel más cercano a, en la citada primera línea de píxeles, la citada frontera de bloque (1), p_{10} denota el citado valor de píxel del citado píxel siguiente más cercano, en la citada línea de píxeles, a la citada frontera de bloque (1), p_{20} denota el citado valor de píxel del citado píxel segundo siguiente más cercano a, en la citada primera línea de píxeles, la citada frontera de bloque (1), p_{03} denota un valor de píxel de un píxel más cercano a, en una segunda línea de píxeles en el citado bloque (10), la citada frontera de bloque (1), p_{13} denota un valor de píxel de un píxel siguiente más cercano a, en la citada segunda línea de píxeles, la citada frontera de bloque (1), y p_{23} denota un valor de píxel de un píxel segundo siguiente más cercano a, en la citada segunda línea de píxeles, la citada frontera de bloque (1), y

35 el citado segundo calculador de valor de decisión (120) está configurado para calcular el citado segundo valor de decisión de filtro como:

$$|q_{20} - 2q_{10} + q_{00}| + |q_{23} - 2q_{13} + q_{03}|$$

 40 donde q_{00} denota el citado valor de píxel del citado píxel del citado bloque vecino (20) más cercano a, en la citada correspondiente primera línea de píxeles, la citada frontera de bloque (1), q_{10} denota el citado valor de píxel del citado píxel del citado bloque vecino (20) siguiente más cercano a, en la citada línea de píxeles correspondiente, la citada frontera de bloque (1), q_{20} denota el citado valor de píxel del citado píxel en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada línea de píxeles correspondiente, la citada frontera de bloque (1), q_{03} denota un valor de píxel de un píxel en el citado bloque vecino (20) más cercano a, en una correspondiente segunda línea de píxeles en el citado bloque vecino (20),

45 la citada frontera de bloque (1), q_{13} denota un valor de píxel de un píxel en el citado bloque vecino (20) siguiente más cercano a, en la citada correspondiente segunda línea de píxeles, la citada frontera de bloque (1) y q_{23} denota un valor de píxel de un píxel en el citado bloque vecino (20) segundo siguiente más cercano a, en la citada correspondiente segunda línea de píxeles, la citada frontera de bloque (1).

50 15. Un producto de programa informático (73) que comprende un medio de código legible por ordenador y un programa informático (74) de acuerdo con la reivindicación 14, almacenado en el citado medio legible por ordenador.

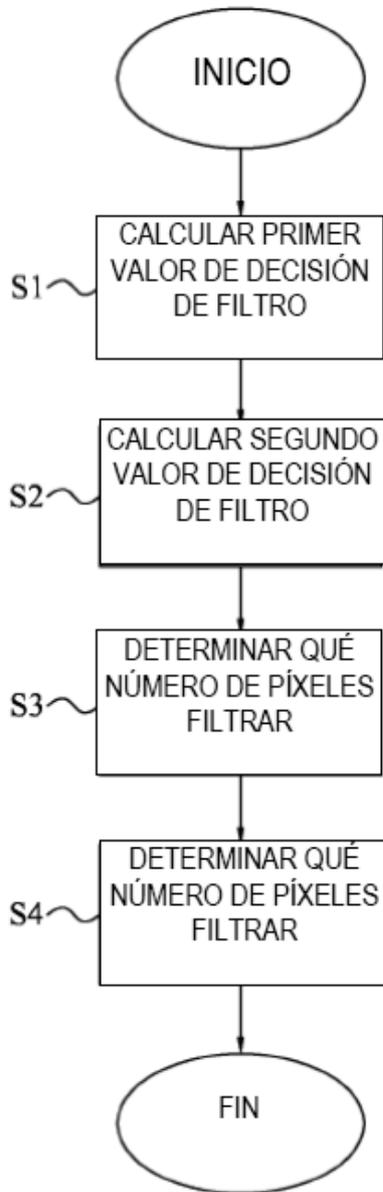


Fig. 1

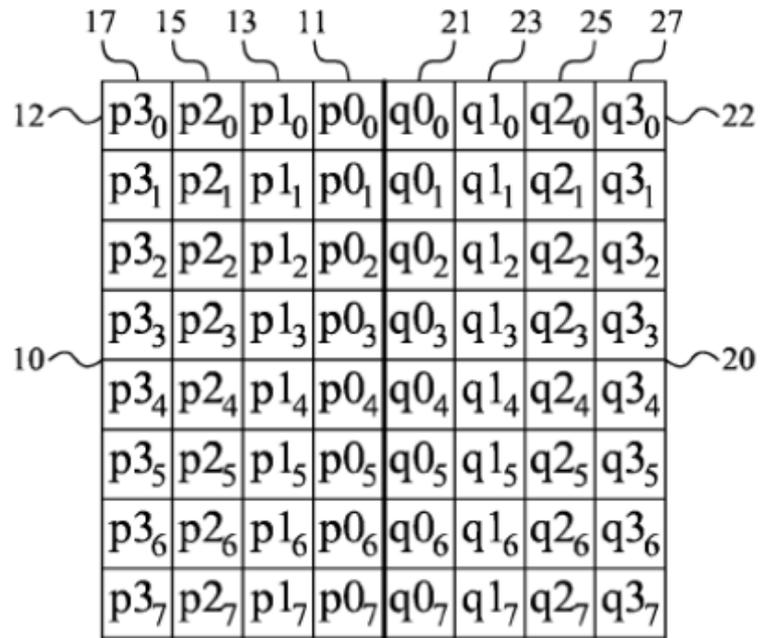


Fig. 2A

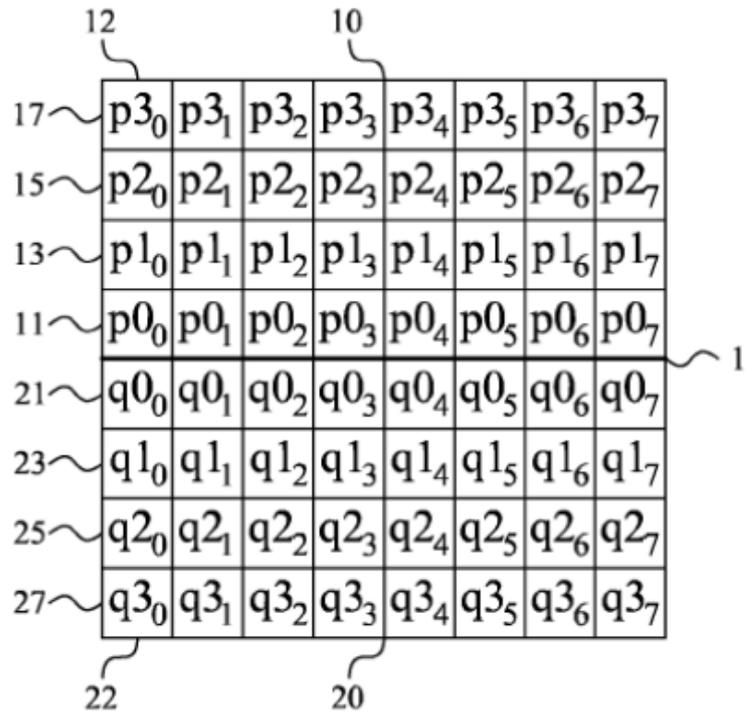


Fig. 2B

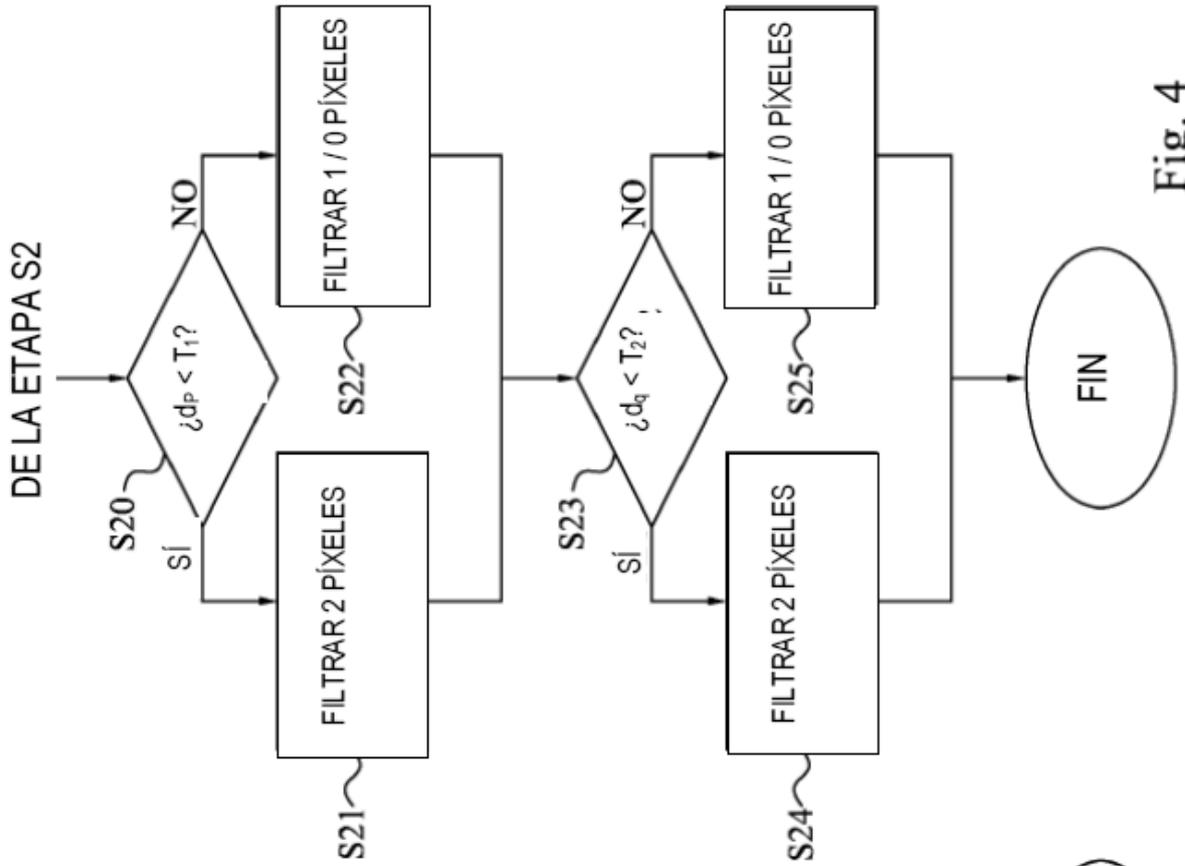


Fig. 4

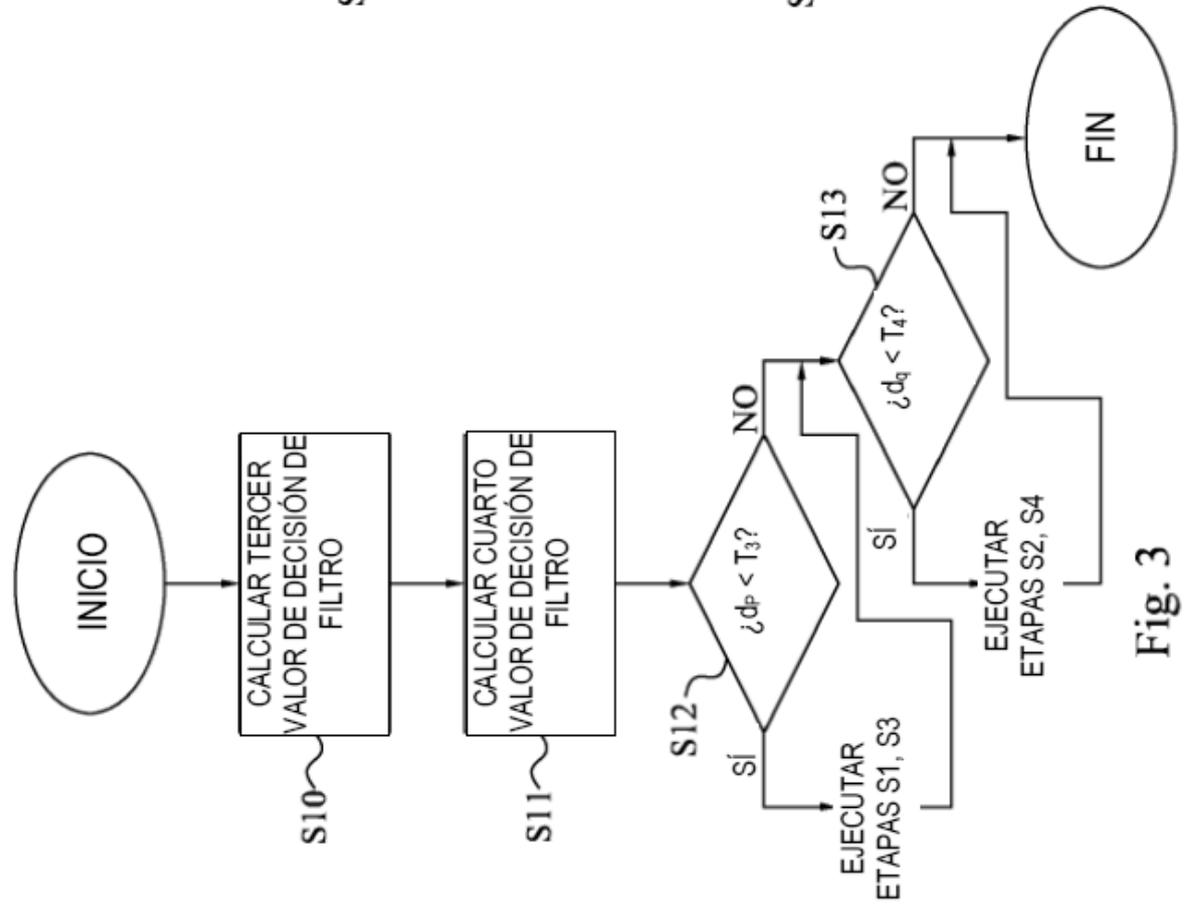


Fig. 3

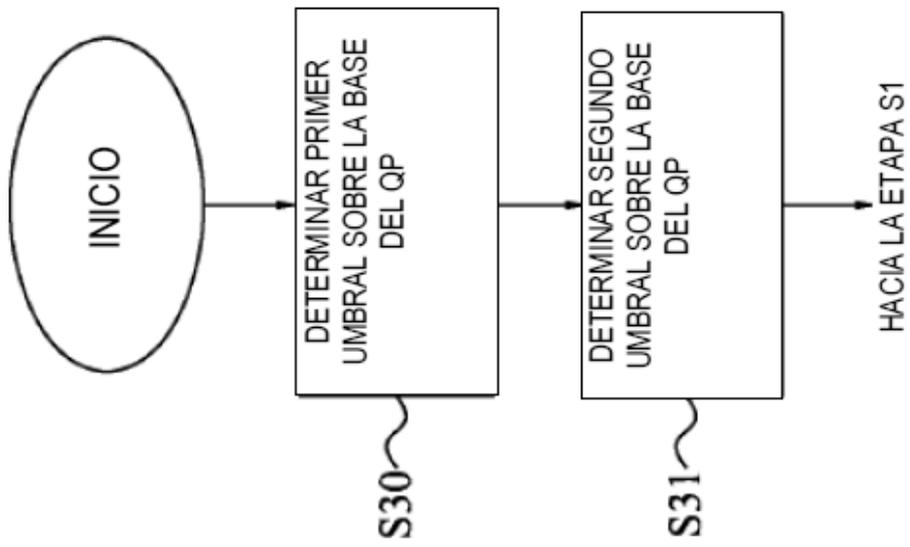


Fig. 5

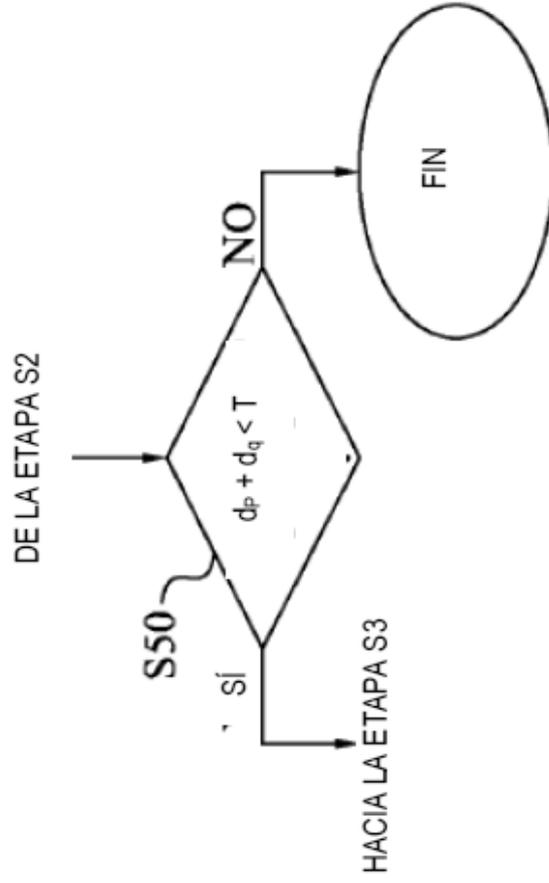


Fig. 7

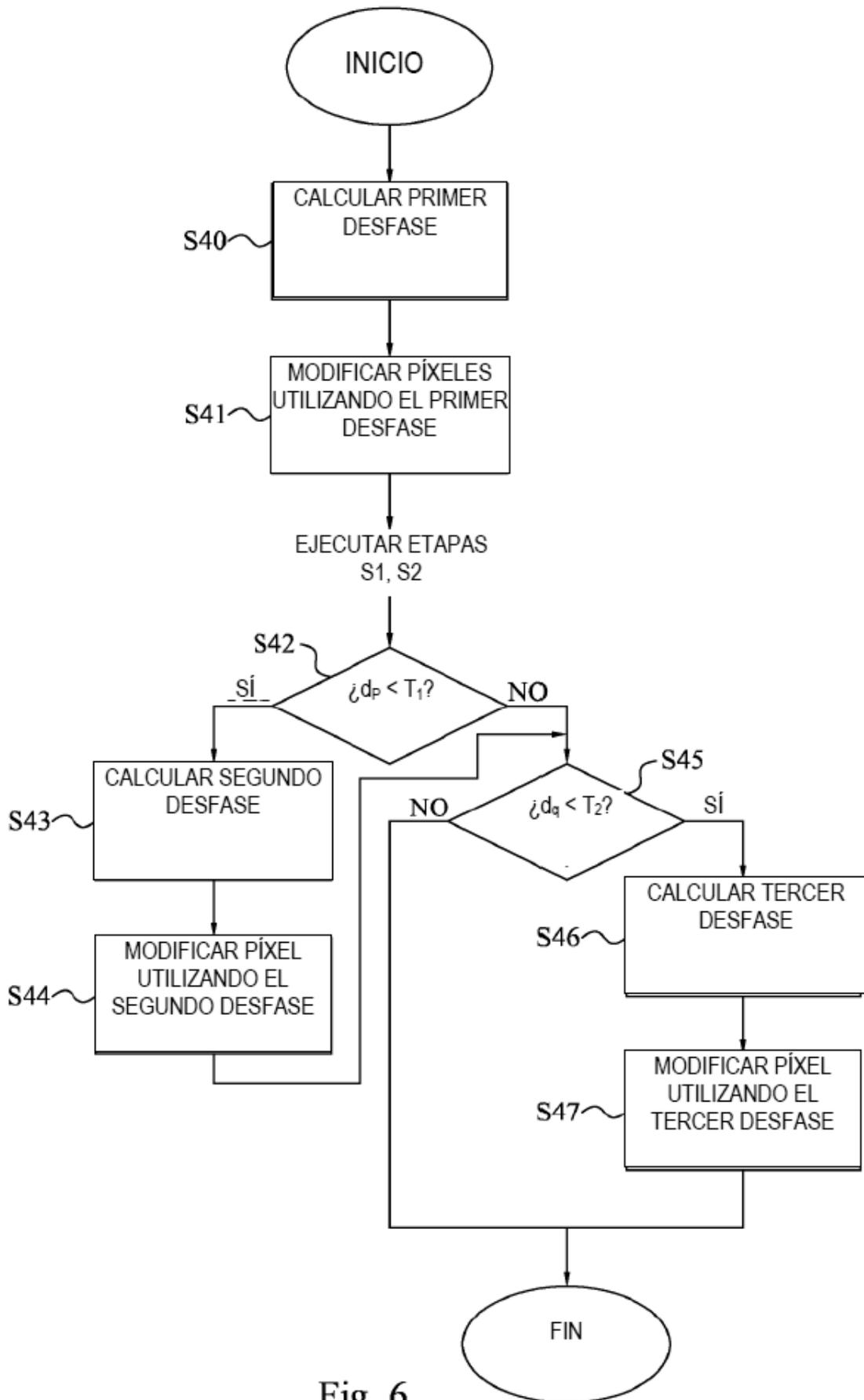


Fig. 6

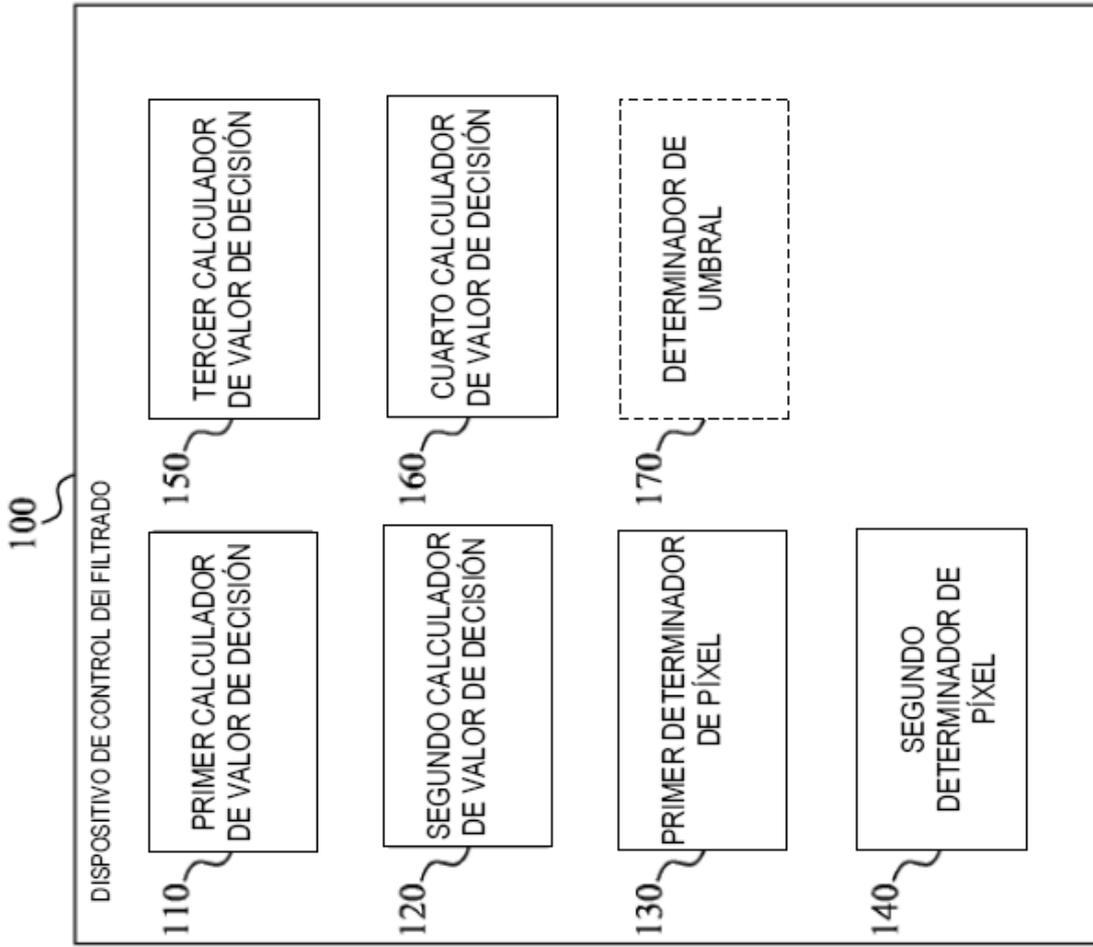


Fig. 9

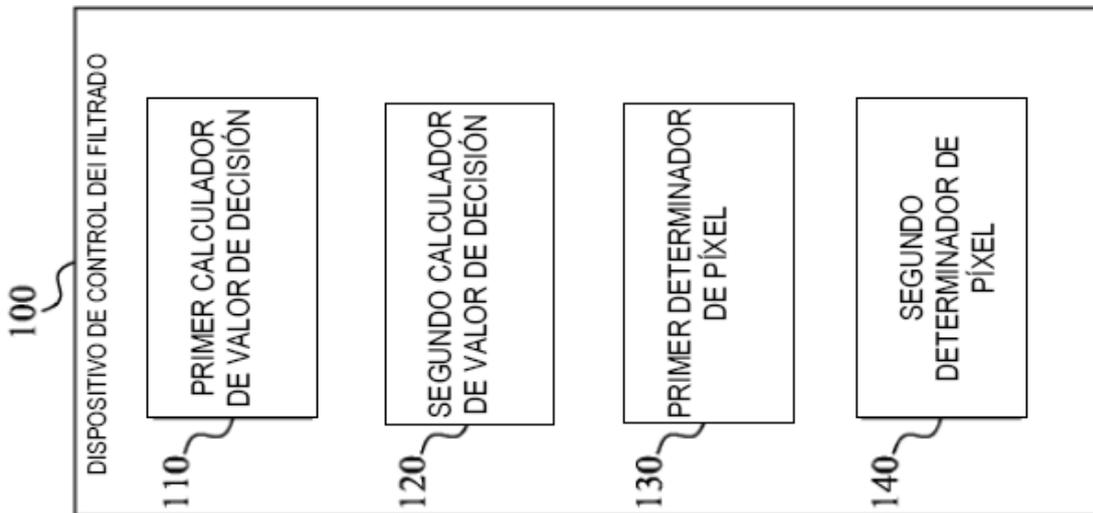


Fig. 8

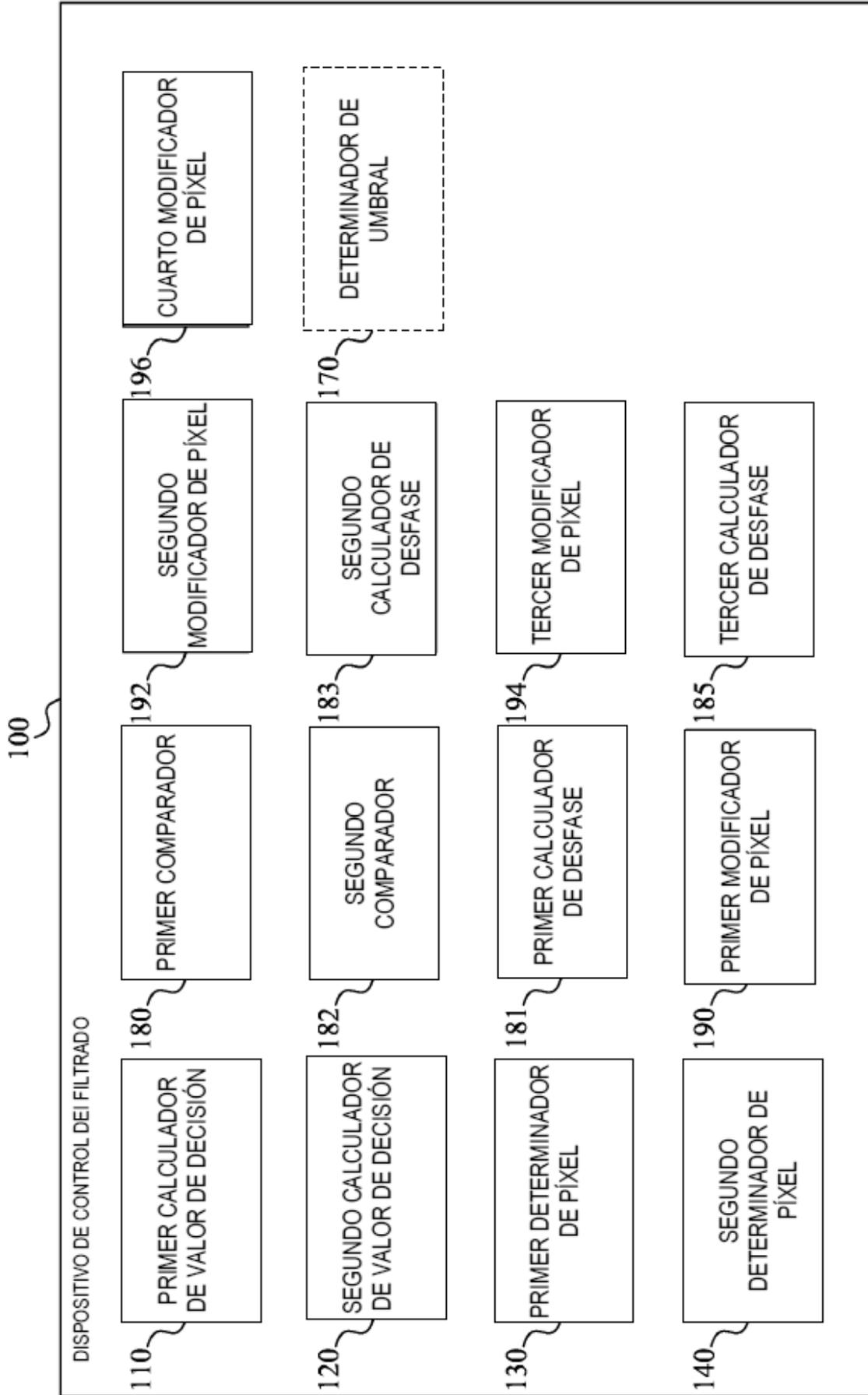


Fig. 10

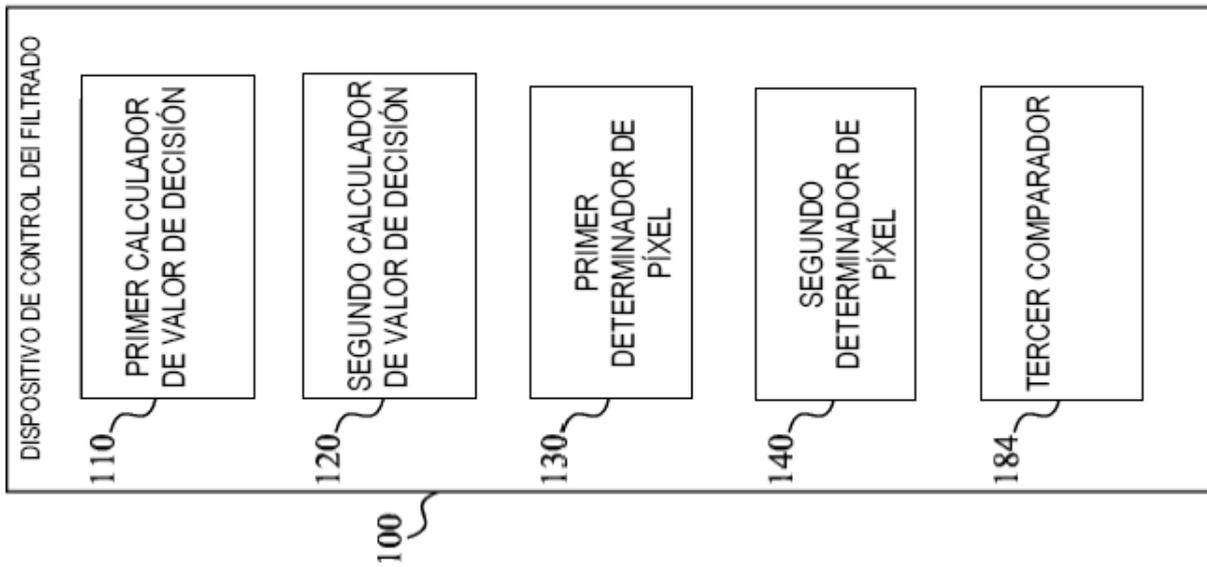


Fig. 11

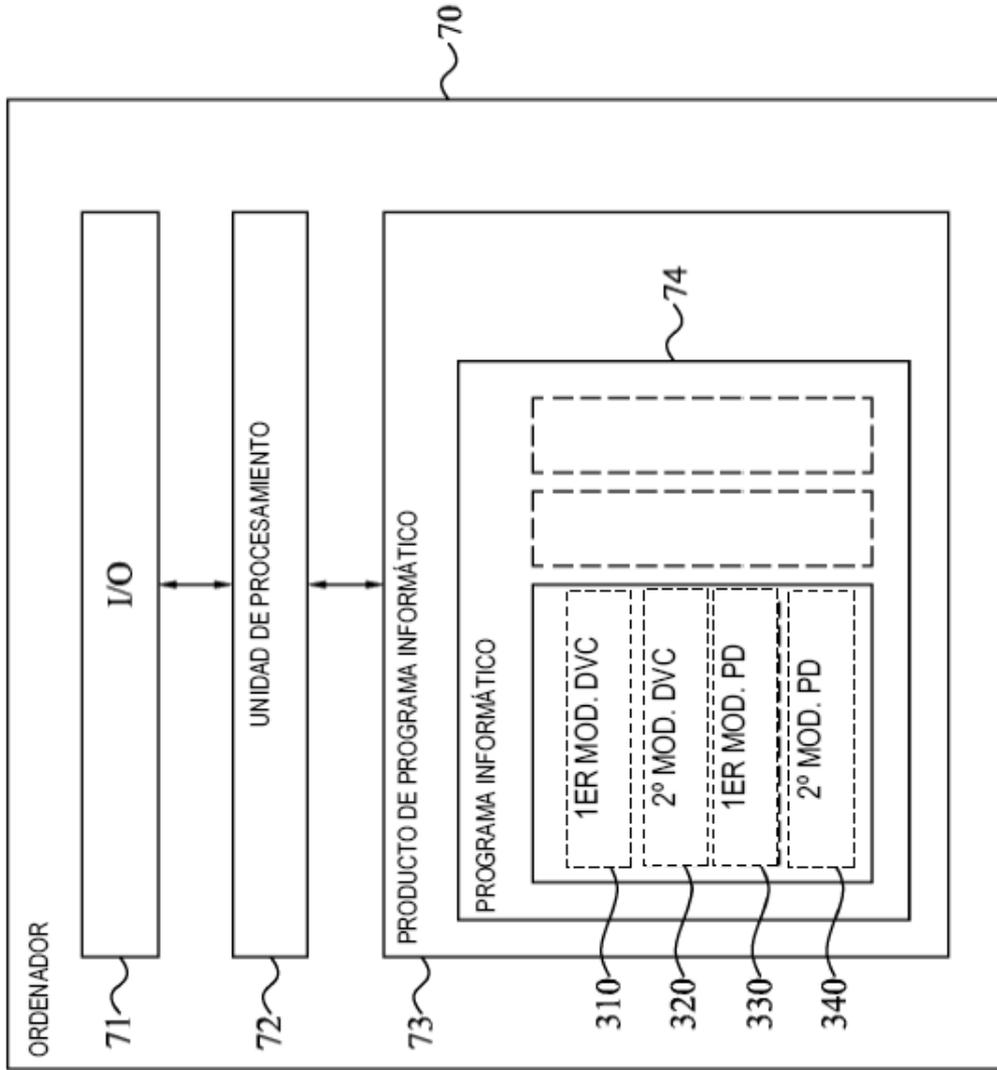


Fig. 12

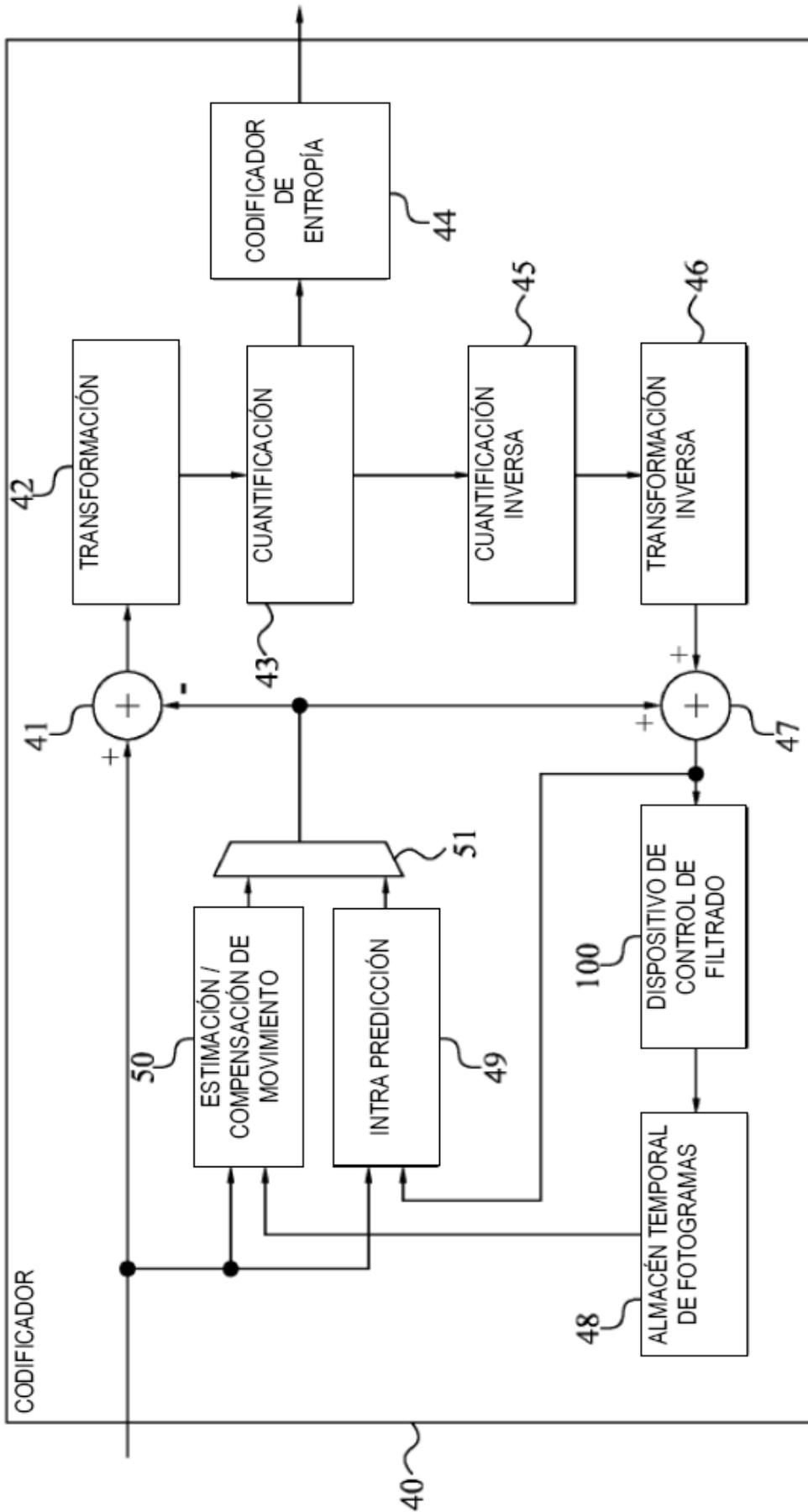


Fig. 13

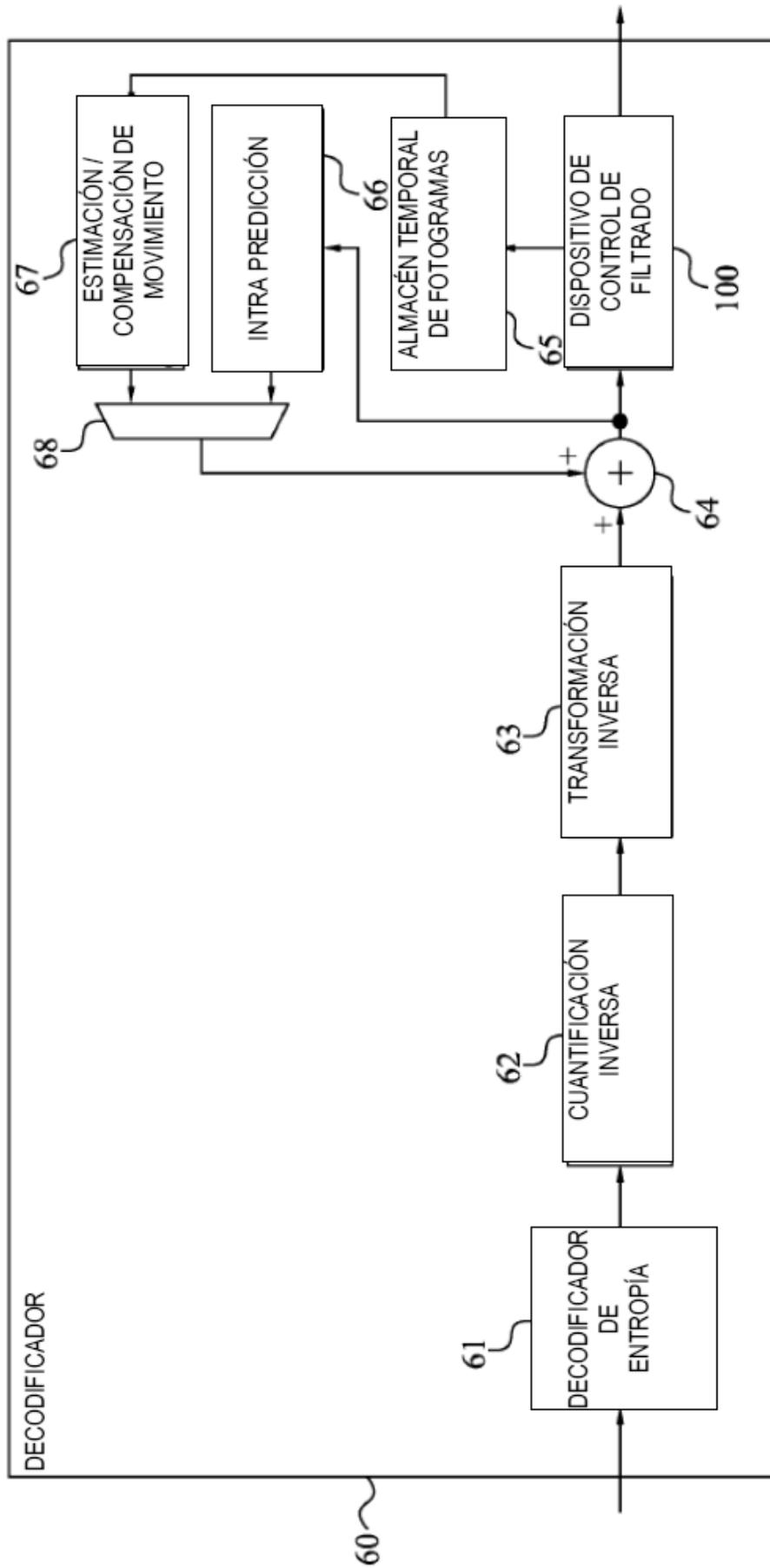


Fig. 14

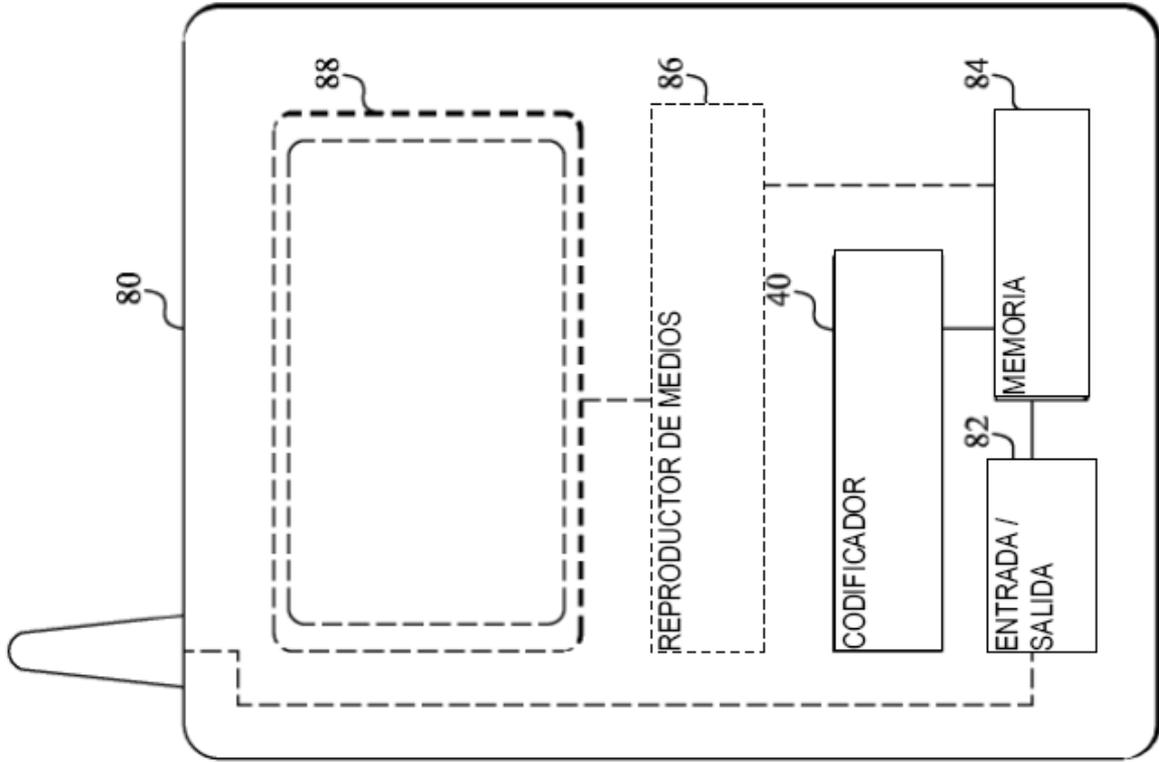


Fig. 16

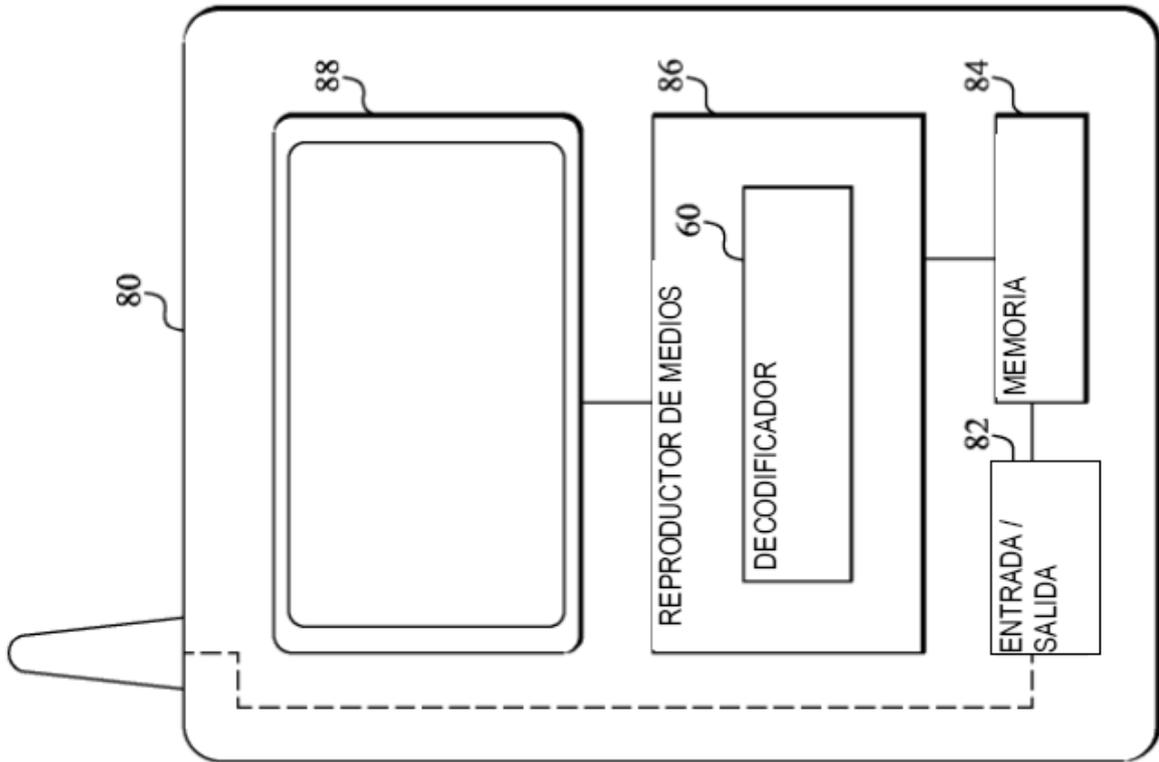


Fig. 15

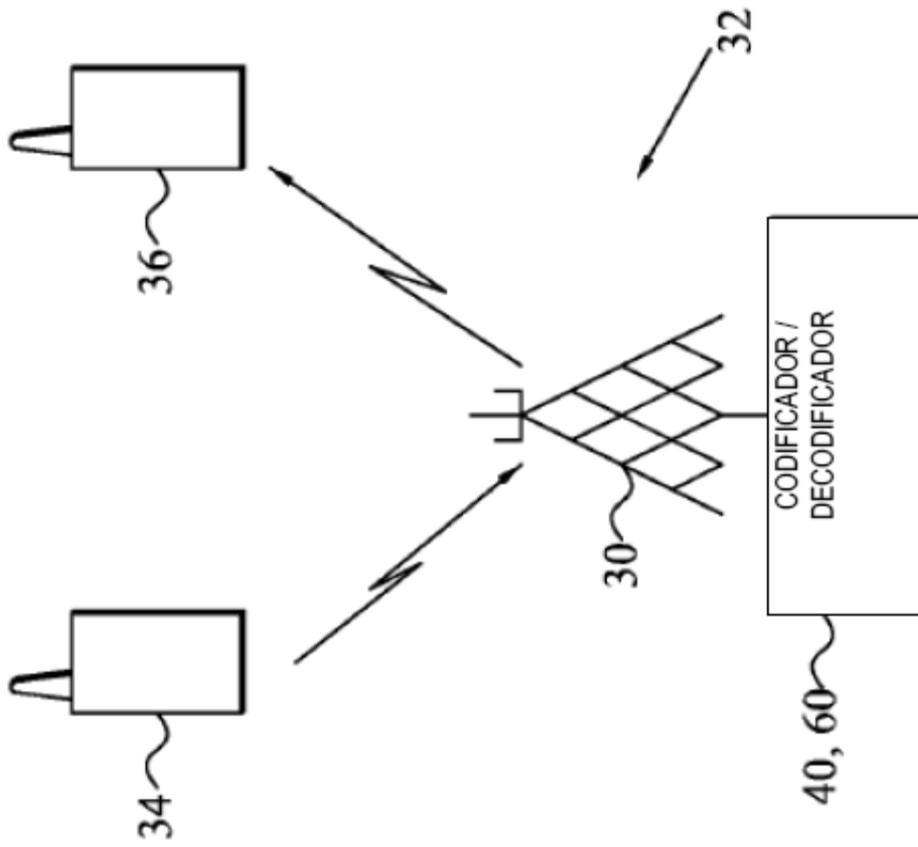


Fig. 17