

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 845 728**

51 Int. Cl.:

H04N 19/117 (2014.01)
H04N 19/136 (2014.01)
H04N 19/176 (2014.01)
H04N 19/463 (2014.01)
H04N 19/70 (2014.01)
H04N 19/523 (2014.01)
H04N 19/82 (2014.01)
H04N 19/86 (2014.01)
H04N 19/61 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **21.12.2011** **PCT/US2011/066666**
87 Fecha y número de publicación internacional: **28.06.2012** **WO12088357**
96 Fecha de presentación y número de la solicitud europea: **21.12.2011** **E 11850066 (9)**
97 Fecha y número de publicación de la concesión europea: **02.12.2020** **EP 2656611**

54 Título: **Filtrado de compensación de movimiento adaptativo al contenido para codificación de vídeo de alta eficiencia**

30 Prioridad:

21.12.2010 US 974940

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
27.07.2021

73 Titular/es:

INTEL CORPORATION (100.0%)
2200 Mission College Boulevard, M/S RNB-4-150
Santa Clara, CA 95054, US

72 Inventor/es:

PURI, ATUL;
SOCEK, DANIEL y
CHOI, CHANG-KEE

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 845 728 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

Filtrado de compensación de movimiento adaptativo al contenido para codificación de vídeo de alta eficiencia

5 **CAMPO TÉCNICO**

La presente descripción se refiere a la codificación y decodificación de señales de vídeo, y más en particular, a la codificación y decodificación basada en libro de códigos de filtros adaptativos utilizados para la predicción de movimiento compensado.

10 **ANTECEDENTES DE LA INVENCION**

El advenimiento de multimedia digital, tales como imágenes digitales, voz/audio, gráficos y vídeo, ha mejorado de manera significativa varias aplicaciones, así como abierto aplicaciones completamente nuevas debido a la relativa facilidad con la que ha permitido un almacenamiento, comunicación, transmisión, y búsquedas fiables y acceso al contenido. En general, las aplicaciones de multimedia digital han sido muchas, abarcando un amplio espectro que incluye entretenimiento, información, medicina y seguridad, y han beneficiado a la sociedad de muchas maneras. El contenido multimedia capturado por sensores tales como cámaras y micrófonos suele ser analógico y el proceso de digitalización en forma de modulación codificada por pulsos (PCM) lo vuelve digital. Sin embargo, inmediatamente después de la digitalización, la cantidad de datos resultantes puede ser bastante significativa, ya que es necesaria para la recreación de la representación analógica que necesitan los altavoces y/o la pantalla del televisor. Por tanto, la comunicación, el almacenamiento o la transmisión eficientes de un gran volumen de contenido multimedia digital requieren su compresión desde la forma PCM en bruto hasta una representación comprimida, y por tanto se han inventado muchas técnicas para la compresión de multimedia. A lo largo de los años, las técnicas de compresión de vídeo se han vuelto muy sofisticadas hasta el punto de que permiten lograr altos factores de compresión entre 10 y 100 mientras conservan una alta calidad psicovisual, a menudo similar al vídeo digital sin comprimir.

Si bien se ha logrado un gran progreso hasta la fecha en la técnica y ciencia de la compresión de vídeo (como lo demuestra la plétora de normas de codificación de vídeo impulsadas por organismos tales como MPEG-1, MPEG-2, H.263, MPEG-4 part2, MPEG-4 AVC/H.264, MPEG-4 SVC y MVC, así como normas patentadas impulsadas por la industria tales como Windows Media Vídeo, RealVídeo, On2 VP y similares), la demanda cada vez mayor de los consumidores por una calidad aún mayor, vídeo de alta definición, y actualmente vídeo en 3D (estéreo), disponible para acceder en cualquier momento y lugar, ha sido necesario la entrega a través de varios medios, tales como DVD/BD, transmisión por aire, cable/satélite, redes cableadas y móviles, a una diversidad de dispositivos clientes tales como PC/portátiles, televisores, decodificadores, consolas de juegos, reproductores/dispositivos multimedia portátiles y teléfonos inteligentes, lo que impulsa el deseo de niveles aún más altos de compresión de vídeo. En las normas impulsadas por los organismos normativos, lo que antecede se evidencia en el esfuerzo recientemente iniciado por ISO MPEG en Codificación de Vídeo de Alta Eficiencia (HEVC), que se espera que combine nuevas contribuciones tecnológicas y tecnología de los últimos años de trabajo exploratorio en H. 265 por el comité de normas de ITU-T.

Todas las normas antes mencionadas emplean una trama general de codificación predictiva intertramas que implica reducir la redundancia temporal compensando el movimiento intertramas (o campos) de vídeo dividiendo primero una trama en bloques y asignando vectores de movimiento a cada bloque de una trama para ser codificado, con respecto a la trama decodificada anterior; estos vectores de movimiento se transmiten a continuación al decodificador y se utilizan para generar una trama de predicción compensada por movimiento que se diferencia con una trama decodificada anterior y codificada bloque por bloque, a menudo mediante codificación de transformación. Para una mayor eficiencia de codificación, se ha reconocido que los vectores de movimiento deben tener una precisión mayor que los píxeles enteros, por lo que MPEG-1 y MPEG-2 permiten una precisión de $\frac{1}{2}$ píxel mientras que las normas más recientes tales como MPEG-4 parte 2 (versión 2) vídeo y H.264 utilizan compensación de movimiento de precisión de $\frac{1}{4}$ píxel. Sin embargo, puesto que los píxeles reales de una trama solamente están disponibles con una precisión de píxeles enteros, se necesitan filtros especiales para interpolar un bloque de la trama anterior a una ubicación de subpíxeles según sea necesario para generar una predicción con compensación de movimiento. La norma H.264 especifica un conjunto de filtros fijos de filtros separables que se pueden utilizar para generar las 16 fases necesarias para la interpolación de $\frac{1}{4}$ píxel. Este conjunto de filtros fijos es teóricamente óptimo, ya que se deriva de la teoría de Weiner para obtener la máxima ganancia; sin embargo, algunos filtros en el conjunto de filtros fijos están limitados a 6 tomas, mientras que para otros se permiten hasta 9 tomas. Además, existe cierta pérdida de precisión en el proceso de integerización debido a la limitación de precisión. Si bien este tipo de predicción por lo general funciona en promedio, este o cualquier otro filtro fijo único, para imágenes o escenas específicas, puede tener una discrepancia con las características del contenido, por lo que hay margen de mejora.

En los últimos años ha existido una investigación sustancial en el área del filtrado para compensación de movimiento de $\frac{1}{4}$ píxel, que conduce al desarrollo del filtrado de movimiento adaptativo. En el filtrado adaptativo, los coeficientes no son fijos y, por tanto, no se conocen a priori; es decir, los coeficientes se calculan a partir del contenido mismo y varían con el tiempo. El procedimiento matemático utilizado para calcular los coeficientes de filtro se basa en la ecuación de Wiener Hopf.

Aunque el procedimiento matemático de Weiner Hopf puede calcular un conjunto de coeficientes de filtro óptimo, existen problemas importantes en la integración práctica de este método en un sistema de codificación de vídeo. Por tanto, en el contexto del desarrollo de H.265 y/o HEVC, se han hecho varias propuestas para abordar las diversas deficiencias, así como para aumentar la adaptabilidad para mejorar la ganancia de codificación. Estas propuestas se pueden enumerar brevemente de la siguiente manera.

Elección entre filtros no separables o separables: el cálculo de filtros no separables puede ser más intensivo en cálculo, pero los filtros teóricamente no separables pueden dar como resultado una calidad superior. Sin embargo, los filtros no separables también requieren aproximadamente el doble de coeficientes de los filtros separables, por lo que la sobrecarga de codificación de los filtros no separables puede ser alta. En general, para el filtrado de movimiento, los filtros separables pueden proporcionar una ganancia de codificación razonable y, por lo tanto, ofrecer una mejor compensación general.

La compensación del número de iteraciones frente a la ganancia para cada iteración - La solución iterativa de la ecuación de Weiner Hopf requiere un número de iteraciones para converger y dar buenos resultados. El número de iteraciones necesarias depende de qué tan lejos esté el conjunto de filtros predeterminado (utilizado inicialmente para la primera iteración) de los resultados óptimos. Si están cerca, a menudo hasta 4 iteraciones pueden ser suficientes, pero si son muy diferentes, pueden ser necesarias de 16 a 20 iteraciones. Si la diferencia entre el filtro predeterminado y el filtro óptimo es mucha, detenerse en 4 iteraciones no dará como resultado la mayoría de las ganancias.

Precisión de varios coeficientes para la mejor compensación de coste de bit frente a calidad - Al mantener ciertos coeficientes en un filtro ajustado a una precisión más baja, pueden resultar algunos ahorros de bits. De hecho, la precisión máxima de un coeficiente de filtro a menudo puede limitarse a 8 bits y, en algunos casos excepcionales, la precisión utilizada puede limitarse a 10 bits.

Reducción del coste de los bits de coeficiente para la transmisión al decodificador (valores reales, diferencial, límites en las actualizaciones) - El número de coeficientes que se enviarán al decodificador debe ser limitado, como por ejemplo, un conjunto de filtros. En condiciones normales no separables puede requerir el envío de hasta 120 coeficientes (con un coste de bits de 650-950 bits por trama) mientras que incluso un conjunto de filtros separables puede requerir el envío de 45-60 coeficientes (con un coste de bits de 400-550 bits por trama). Con codificación diferencial (codificar el conjunto de filtros de la trama actual de forma diferencial con el conjunto de filtros de la trama anterior, o incluso con respecto al conjunto de filtros predeterminado) o poniendo límites a las actualizaciones, este recuento de bits se puede reducir un poco, aunque con cierta pérdida de calidad.

Múltiples conjuntos de filtros dentro de una imagen para mejorar las ganancias generales - Mientras que incluso un conjunto de filtros Wiener calculó cada trama, puede dar ganancias sobre, digamos, el conjunto de filtros estándar H.264, al tener la opción de múltiples conjuntos de filtros dentro de una trama (tal como sobre la base de bloques o en segmentos) puede generar mayores ganancias. Sin embargo, un problema es que el coste de bits de incluso 2 conjuntos de filtros por trama puede ser excesivo en comparación con las ganancias adicionales. Por tanto, es necesario gestionar el coste de bits del coeficiente de filtro cuando se utilizan varios conjuntos de filtros.

La complejidad de la optimización de la distorsión de frecuencia ("RDO"), que puede implicar de otro modo múltiples iteraciones en el mapa de cambio de filtro para obtener los mejores resultados. Cuando se utilizan varios filtros, para obtener una alta ganancia, a menudo es necesario utilizar la optimización de la distorsión de la tasa de una manera iterativa para obtener el tamaño de bloque correcto para la selección del filtro que produce una buena compensación entre la frecuencia de cambio de filtro y las ganancias.

Coste de bits de sobrecarga del mapa de bloques cuando se utilizan múltiples filtros - Cuando se utilizan varios filtros (incluyendo el caso de cambiar entre un filtro normal y un filtro calculado), el coste del mapa de conmutación puede ser sustancial. Por ejemplo, si uno cambiara entre dos filtros sobre una base de macrobloque utilizando 1 bit para el macrobloque, entonces para una secuencia de Formato Intermedio Común ("CIF"), se añadirían 396 bits de sobrecarga adicional, además del coste de bits para posiblemente enviar dos conjuntos de filtros.

Ganancias adicionales mediante el uso de filtros de posición entera y filtros con compensaciones - Se han realizado algunos intentos para reducir las ganancias calculando un filtro para la posición entera en caso de que la mejor posición de $\frac{1}{4}$ de píxel sea la posición entera. Además, se han realizado algunos experimentos utilizando filtros con compensaciones para obtener mayores ganancias. Ambos casos requieren bits adicionales de sobrecarga.

Tal como se señaló con anterioridad, aunque se han sugerido muchas técnicas y variaciones, la sobrecarga del recuento de bits del coeficiente de los métodos actuales, incluso después de la codificación diferencial, es demasiado alta. Además, las técnicas existentes son computacionalmente complejas debido a la necesidad de calcular sobre la marcha múltiples iteraciones de conjuntos de coeficientes. Además, la única forma de que el sistema actual sea muy adaptable es mediante el uso extensivo de RDO para múltiples filtros conmutados, lo que significa una sobrecarga y complejidad adicionales.

Se puede encontrar información adicional relacionada con el filtrado de compensación de movimiento adaptativo en las siguientes referencias, para todos los fines:

T. Wedi, "Filtro de interpolación adaptativa para predicción con compensación de movimiento", Proc. En t. Conf sobre procesamiento de imágenes (ICIP) 2002, págs. 509-511, 2002.

V. Vatis et al, "Codificación de coeficientes de filtro de interpolación de Wiener adaptativo no separable bidimensional", Proc. SPIE Visual Communications and Image Processing (VCIP) 2005, julio de 2005.

V. Vatis y Joern Ostermann, "Filtro de interpolación no separable adaptable localmente para H.264/AVC", Proc. Int. Conf on Image Processing (ICIP) 2006, octubre de 2006.

S. Wittmann y T. Wedi, "Filtro de interpolación adaptable separable para codificación de vídeo", Proc. Int. Conf on Image Processing (ICIP) 2008, págs. 2500-2503, 2008.

El documento "Modelo de prueba en consideración - JCTVC-B205 (borrador 007)", 6 de octubre de 2010 Kemal Ugur et al.: "Codificación de vídeo de alto rendimiento, baja complejidad y la norma HEVC emergente", TRANSACCIONES IEEE EN CIRCUITOS Y SISTEMAS PARA TECNOLOGÍA DE VÍDEO, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 20, nº 12, 1 de diciembre de 2010 (2010-12-01), páginas 1688-1697.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La Figura 1 muestra un diagrama de bloques general de un codificador intertramas de la técnica anterior que emplea predicción de movimiento compensado que utiliza filtrado de movimiento fijo.

La Figura 2 muestra un decodificador de la técnica anterior, correspondiente al codificador de la técnica anterior, y funciona de manera similar al bucle de decodificación local en el codificador.

La Figura 3 muestra una parte de una trama de vídeo organizada como una cuadrícula de píxeles que se divide, además, en una cuadrícula de *subpíxeles*, en donde se calcula un bloque de predicción de movimiento compensado.

Las Figuras 4a-o muestran un mapeo de correspondencia de 6/9 tomas en una matriz de filtro de 6x6 para cada posición de subpíxel.

La Figura 5 muestra un diagrama de bloques de un codificador intertramas de la técnica anterior que emplea un predictor de movimiento compensado que utiliza filtrado de movimiento adaptativo generalizado.

La Figura 6 muestra un diagrama de bloques de un decodificador intertramas de la técnica anterior que emplea un predictor de movimiento compensado que utiliza filtrado de movimiento adaptativo generalizado.

Las Figuras 7a-f muestran varios modos de partición para dividir una imagen.

La Figura 8 muestra un diagrama de bloques de un codificador intertramas que emplea un predictor de movimiento compensado que utiliza un codificador de libro de códigos de filtrado de movimiento de conformidad con una forma de realización.

La Figura 9 muestra un diagrama de bloques de un decodificador intertramas que emplea un predictor de movimiento compensado que utiliza un decodificador de libro de códigos de filtrado de movimiento de conformidad con una forma de realización.

La Figura 10 muestra una rutina "fuera de línea" para generar un libro de códigos de filtro de movimiento o libros de códigos para ubicaciones de subpíxeles y de píxeles enteros.

La Figura 11 muestra una rutina "online" para buscar en un libro de códigos un conjunto de filtros de movimiento de conformidad con una forma de realización.

La Figura 12 ilustra una sintaxis de flujo binario, a modo de ejemplo, de conformidad con varios codificadores y decodificadores de filtro de libro de códigos de compensación de movimiento y que se describen aquí de diversas formas.

Las Figuras 13a-b ilustran una secuencia de imágenes o tramas dentro de un flujo de vídeo, de conformidad con varias formas de realización.

La Figura 14 ilustra un sistema de codificación/decodificación de vídeo, a modo de ejemplo, según una forma de realización.

La Figura 15 ilustra varios componentes de un dispositivo de codificación, a modo de ejemplo, de conformidad con una forma de realización.

5 La Figura 16 ilustra varios componentes de un dispositivo de decodificación, a modo de ejemplo, de conformidad con una forma de realización.

La Figura 17 ilustra una rutina de codificación de vídeo de filtrado de compensación de movimiento de libro de códigos, tal como puede realizarse mediante un dispositivo de codificación de conformidad con una forma de realización.

10 La Figura 18 ilustra una subrutina para obtener un libro de códigos, de conformidad con una forma de realización.

La Figura 19 ilustra una subrutina para seleccionar un subconjunto de filtros de compensación de movimiento desde el libro de códigos, de conformidad con una forma de realización.

15 La Figura 20 ilustra una subrutina para determinar un filtro de compensación de movimiento de posición entera para una parte de una imagen, de conformidad con una forma de realización.

20 La Figura 21 ilustra una rutina de decodificación de vídeo de filtrado de compensación de movimiento de libro de códigos, tal como puede realizarse mediante un dispositivo de decodificación de conformidad con una forma de realización.

DESCRIPCIÓN DETALLADA DE LA INVENCION

25 La descripción detallada que sigue está representada en gran parte en términos de procesos y representaciones simbólicas de operaciones mediante componentes informáticos convencionales, incluyendo un procesador, dispositivos de almacenamiento de memoria para el procesador, dispositivos de visualización conectados y dispositivos de entrada. Además, estos procesos y operaciones pueden utilizar componentes informáticos convencionales en un entorno informático distribuido heterogéneo, incluyendo servidores de archivos remotos, servidores informáticos y dispositivos de almacenamiento de memoria. El procesador puede acceder a cada uno de
30 estos componentes informáticos distribuidos convencionales a través de una red de comunicación.

Las expresiones "en una forma de realización", "en varias formas de realización", "en algunas formas de realización" y similares se utilizan de manera repetida. Dichas expresiones no se refieren necesariamente a la misma forma de realización. Los términos "que comprende", "que tiene" e "que incluye" son sinónimos, a menos que el contexto indique lo contrario.
35

La presente invención se define en las reivindicaciones independientes. La descripción que permite la invención se encuentra en la forma de realización de las Figuras 10 y 11. Las formas de realización restantes deben entenderse como ejemplos que no describen partes de la presente invención.
40

Visión general

45 Se describen diversas formas de realización de sistemas y métodos para el filtrado de movimiento para la codificación de vídeo intertramas. Algunas de estas formas de realización pueden ofrecer una complejidad informática reducida en comparación con las soluciones existentes, ya que, en lugar de tomar muchas iteraciones para calcular iterativamente los coeficientes de filtro durante la codificación, se busca un conjunto de filtros de movimiento de un libro de códigos precalculado de conjuntos de coeficientes de filtros de movimiento.

50 En algunas formas de realización, la complejidad de la búsqueda puede reducirse aún más dividiendo el libro de códigos completo en un libro de códigos "base" pequeño y un libro de códigos "virtual" más grande, de modo que los cálculos principales para la búsqueda solamente necesitan realizarse en el libro de códigos base. Dichas formas de realización pueden reducir el almacenamiento (a) almacenando solamente el libro de códigos base y un algoritmo para sintetizar el libro de códigos restante; (b) almacenar, para cada conjunto de filtros, un filtro con un número apropiado de tomas de conformidad con la ubicación de filtrado de movimiento subpíxel; y (c) almacenar solamente la precisión de bits necesaria por coeficiente.
55

60 En varias formas de realización, las técnicas de libro de códigos de filtro de movimiento descritas se pueden emplear en imágenes, grupos de cortes, mapas de bloques, mapas de bloques fusionados o particiones adaptativas tales como zonas dependientes del contenido bi-tree, quad-tree o similares. La capacidad de esta técnica para trabajar con particiones dependientes del contenido también reduce la complejidad y mejora la precisión en comparación con otras técnicas que son impulsadas únicamente por la Optimización de Distorsión de Tasa ("RDO"). Se pueden utilizar varias formas de realización para el filtrado de subpíxeles solamente, el filtrado de posición entera solamente, o ambos. Además, el libro de códigos se puede sustituir o ampliar fácilmente para adaptarlo a la naturaleza del contenido.

65 Se describen los detalles del diseño del libro de códigos de filtros y la búsqueda eficiente del libro de códigos. Con el método descrito de diseño de libro de códigos y para el tamaño de libro de códigos sugerido, la técnica puede dar

como resultado un buen rendimiento de filtrado de movimiento, aumentando, de manera potencial, la eficiencia de predicción para un amplio espectro de contenido de vídeo.

En varias formas de realización, las técnicas descritas pueden ser eficientes en bits, ya que solamente puede ser necesario un índice en el libro de códigos para identificar un conjunto de filtros subpíxeles o un filtro de píxeles enteros seleccionados. En muchos casos, codificar un índice puede ser más eficaz que codificar coeficientes reales o enviar coeficientes de filtro codificados diferencialmente. Si bien el número de índices reales enviados depende del número de particiones, en comparación con los métodos existentes que utilizan el mismo tipo de partición, las técnicas descritas basadas en libros de códigos pueden ser más eficientes en tasa binarias.

Se pueden utilizar varias formas de realización con codificación de vídeo típica que utilizan estructuras de imagen tradicionales compuestas por codificación de imagen I, P, B. Se pueden utilizar otras formas de realización con otras estructuras en su lugar o además de las imágenes I y P, tales como imágenes B jerárquicas, imágenes B unidireccionales u otras alternativas de imagen B.

A continuación, se hace referencia en detalle a la descripción de las formas de realización ilustradas en los dibujos.

La Figura 1 muestra un diagrama de bloques general de un codificador 100 intertramas de la técnica anterior que emplea predicción de movimiento compensado que utiliza filtrado de movimiento fijo. Una o más tramas de vídeo (*vidfrms*) se dividen en bloques (*blks*) por el indexador de bloques 101. Los bloques (*blks*) se introducen uno a la vez al diferenciador 133 y se diferencian con los bloques de señal de predicción correspondientes (*pred*), y el residuo (*res*) se reenvía transformado por el transformador 135, lo que da como resultado un bloque de coeficientes (*tcof*) que luego se envía al cuantificador 138, lo que da como resultado un bloque de coeficientes cuantificados (*tcofq*) que se envían tanto al codificador de entropía 118 como al cuantificador inverso 140. El cuantificador inverso 140 envía un bloque de coeficientes des-cuantificados (*tcof'*) al transformador inverso 143. En el sumador 145, el bloque de predicción (*pred*) del bucle de compensación de movimiento se agrega al bloque residual des-cuantificado (*res'*) en la salida del transformador inverso 143. En el sumador 145, el bloque de predicción (*pred*) del bucle de compensación de movimiento se agrega al bloque residual descuantificado (*res'*) en la salida del transformador inverso 743. El sumador 745 genera el bloque decodificado localmente (*rec*), que luego se envía al ensamblador de tramas y al procesador de filtro de desbloqueo 148, lo que reduce el bloqueo. Los bloques (*recd*) en la salida del ensamblador de tramas y el procesador de filtrado de desbloqueo 148 forman una trama que se utiliza como trama de referencia para el estimador de movimiento 128 y el predictor compensado por movimiento 130.

El estimador de movimiento 128 calcula vectores de movimiento (*mv*) entre bloques (*blks*) de la trama original con respecto a la trama reconstruida (*recd*). El proceso de cálculo del vector de movimiento incluye calcular un vector de píxeles enteros por bloque, seguido de un vector de subpíxeles (*sv*) (a, por ejemplo, $\frac{1}{4}$ de resolución de píxel) que se calcula utilizando el interpolador subpíxeles 129. En condiciones normales, el interpolador subpíxeles 129 está contenido en el estimador de movimiento 128 en sí, pero para mayor claridad de la ilustración se muestra y se menciona por separado. El estimador de movimiento 128 busca la mejor coincidencia y, por tanto, el mejor vector de movimiento (*mv*), donde, como predictor de movimiento compensado 130, utiliza el vector de movimiento (*mv*) para generar un bloque de predicción por movimiento compensado (*pred*). El compensador de movimiento 130 también utiliza un interpolador de subpíxeles 131 (por ejemplo, a resolución de $\frac{1}{4}$ píxel) como el interpolador de subpíxeles 129. El conjunto de filtros (*fmf*) utilizado por ambos interpoladores de subpíxeles 129 y 131 está preespecificado y así fijo. El codificador de entropía 118 codifica los coeficientes de transformación cuantificados (no ilustrados), vectores de movimiento diferencial (*dmv*) y datos generales (no ilustrados), generando un flujo binario comprimido 128.

La Figura 2 muestra un decodificador 200 de la técnica anterior, correspondiente al codificador 100 de la técnica anterior, y funciona de manera similar al bucle de decodificación local 155 en el codificador 100. Concretamente, el flujo binario comprimido 201 que se va a decodificar se introduce en el decodificador de entropía 203 que decodifica bloques de coeficientes cuantificados (*tcofq*), vectores de movimiento diferencial (*dmv*) y datos generales (no ilustrados). Los bloques de coeficientes cuantificados (*tcofq*) se cuantifican entonces de forma inversa mediante el cuantificador inverso 203, lo que da como resultado coeficientes des-cuantificados (*tcof'*), que luego se transforman de forma inversa mediante el transformador inverso 215, lo que da como resultado bloques residuales decodificados (*res'*). En el sumador 223, los coeficientes des-cuantificados (*tcof'*) se agregan a los bloques de predicción compensados por movimiento (*pred*) obtenidos utilizando los vectores de movimiento correspondientes (*mv*) en el predictor de movimiento compensado 230, utilizando el vector de subpíxeles (*sv*) (a, por ejemplo, $\frac{1}{4}$ píxeles de resolución), que se calcula utilizando el interpolador de subpíxeles 229. El vídeo decodificado resultante (*rec*) se desbloquea en el ensamblador de tramas y el filtro de desbloqueo 225, y al igual que en el codificador 100, el vídeo desbloqueado (*recd*) se utiliza como una trama de referencia. El predictor de movimiento compensado 230 en el decodificador 200 funciona de manera similar al predictor de movimiento compensado 130 en el codificador 100, tal como el predictor de movimiento compensado 230 utiliza un interpolador de subpíxeles 229 similar con el mismo filtro de movimiento fijo (*fmf*) que el utilizado por el codificador 100.

La Figura 3 muestra una parte 300 de una trama de vídeo organizada como una cuadrícula de píxeles que se divide, además, en una cuadrícula de subpíxeles, en donde se calcula un bloque de predicción por movimiento compensado. Las ubicaciones de píxeles enteros se muestran como A1-A6, B1-B6, C1-C6, D1-D6, E1-E6 y F1-F6. Las ubicaciones

de $\frac{1}{4}$ píxel entre C3-C4 y D3-D4 se muestran mediante las posiciones $\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o\}$ tales que cada bloque puede compensarse en una de estas posiciones de cuarto de píxel. La Figura 3 también muestra varias ubicaciones de $\frac{1}{2}$ píxel en las posiciones $aa, bb, cc, dd, ee, ff, gg, hh, ii$ y jj .

5 El cálculo de los conjuntos de filtros para las posiciones de subpíxeles $\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o\}$ se realiza utilizando filtros de 6 tomas o filtros de 9 tomas. Los coeficientes de filtro se mapean en una matriz de 6x6 que se utiliza para calcular el valor de subpíxeles desde la proximidad de 6x6 de los píxeles circundantes (A1-F6).

Las Figuras 4a-o muestran un mapeo de 6/9 tomas en una matriz de filtro de 6x6 para cada posición de subpíxel.

10 La Figura 4a muestra los coeficientes de filtro y los píxeles sobre los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{a\}$; esta posición requiere 6 coeficientes de filtro.

15 La Figura 4b muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{b\}$; esta posición corresponde a la posición horizontal de la mitad del píxel y requiere solamente 3 coeficientes de filtro debido a la simetría.

20 La Figura 4c muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{c\}$; esta posición requiere 6 coeficientes de filtro.

La Figura 4d muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{d\}$; esta posición requiere 6 coeficientes de filtro.

25 La Figura 4e muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{e\}$; esta posición requiere 6 coeficientes de filtro.

La Figura 4f muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{f\}$; esta posición requiere 9 coeficientes de filtro.

30 La Figura 4g muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{g\}$; esta posición requiere 6 coeficientes de filtro.

35 La Figura 4h muestra los coeficientes de filtro y los píxeles sobre los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{h\}$; esta posición corresponde a la posición vertical de mitad de píxel y debido a la simetría solamente requiere 3 coeficientes.

La Figura 4i muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{i\}$; esta posición requiere 9 coeficientes de filtro.

40 La Figura 4j muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{j\}$; esta posición requiere 9 coeficientes de filtro.

La Figura 4k muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{k\}$; esta posición requiere 9 coeficientes de filtro.

45 La Figura 4l muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{l\}$; esta posición requiere 6 coeficientes de filtro.

50 La Figura 4m muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{m\}$; esta posición requiere 6 coeficientes de filtro.

La Figura 4n muestra los coeficientes de filtro y los píxeles en los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{n\}$; esta posición requiere 9 coeficientes de filtro.

55 La Figura 4o muestra los coeficientes de filtro y los píxeles sobre los que se aplican para calcular directamente el bloque de interpolación filtrado en la posición de subpíxeles $\{o\}$; esta posición requiere 6 coeficientes de filtro.

Filtrado de Wiener adaptativo

60 Antes de describir los detalles de la Figura 5, se describe brevemente la teoría del filtrado óptimo, ya que esa es la base del diseño de filtro adaptativo utilizando el método de filtrado de Wiener.

65 Para cada una de las ubicaciones de los *subpíxeles*, se acumulan matrices de correlación cruzada y autocorrelación, a partir de las cuales se puede calcular el conjunto de filtros Wiener óptimo resolviendo la ecuación de Weiner Hopf tal como sigue.

Sea $x(n)$ la señal de entrada, $y(n)$ la salida y $h(n)$ represente los coeficientes de filtro.

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

Salida de filtro:

5 Señal de error: $e(n)=d(n) - y(n)$

Error cuadrático medio: $J=E[e^2(n)]$

$$\mathbf{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(N-1) \end{bmatrix}$$

En forma vectorial:

10

$$y(n) = \mathbf{h}^T \mathbf{x}(n) = \mathbf{x}(n)^T \mathbf{h}$$

$$\begin{aligned} E[e^2(n)] &= E[(d(n) - y(n))^2] \\ &= E[d^2(n)] - 2E[d(n)\mathbf{x}(n)^T]\mathbf{h} + \mathbf{h}^T E[\mathbf{x}(n)\mathbf{x}(n)^T]\mathbf{h} \\ &= P_d - 2\mathbf{R}_{dx}^T \mathbf{h} + \mathbf{h}^T \mathbf{R}_{xx} \mathbf{h} \end{aligned}$$

15 En donde, P_d es un escalar.

Vector de fila de correlación cruzada: $\mathbf{R}_{dx}=E[d(n)\mathbf{x}(n)^T]$

Matriz de autocorrelación: $\mathbf{R}_{xx}=E[\mathbf{x}(n)\mathbf{x}(n)^T]$

20

Para encontrar el mínimo, se toma la derivada y se pone a cero de la siguiente manera:

$$\frac{\partial E[e^2(n)]}{\partial \mathbf{h}(k)} = -2\mathbf{R}_{dx}^T + 2\mathbf{R}_{xx} \mathbf{h} = 0$$

25 A continuación, despejando $\mathbf{h}$, la ecuación de Wiener Hopf es la siguiente:

$$\mathbf{h} = \mathbf{R}_{xx}^{-1} \mathbf{R}_{dx}$$

30

La ecuación de Wiener Hopf determina los coeficientes de filtro óptimos en el error cuadrático medio, y el filtro resultante se denomina filtro "Wiener".

35

En la ecuación anterior, $\mathbf{h}$ es el vector de coeficientes de filtro (concretamente 6 o 9 coeficientes), $\mathbf{R}_{xx}$ es la matriz de autocorrelación (para datos de bloque de la trama de referencia) y $\mathbf{R}_{dx}$ es una matriz de correlación cruzada/vector de fila (entre las tramas de origen y datos de bloque de trama de referencia). Las matrices se acumulan para cada posición de subpíxeles por separado, por lo que $\mathbf{h}$ puede calcularse resolviendo la ecuación antes mencionada para cada posición de subpíxel, obteniendo así el conjunto de filtros adaptado de 16 filtros (para ubicaciones de $1/4$ de píxel).

40

La Figura 5 muestra un diagrama de bloques de un codificador intertramas 500 de la técnica anterior que emplea un predictor de movimiento compensado que utiliza filtrado de movimiento adaptativo generalizado. El indexador de bloques 501, la transformada directa 535, el cuantificador 538, el cuantificador inverso 540, la transformada inversa 543, el ensamblador de tramas y el procesador de filtrado de desbloqueo 548 y el codificador de entropía 518 funcionan de la misma manera que los componentes correspondientes del codificador 100, tal como se muestra en la Figura 1 y descrito con anterioridad, y sus descripciones no necesitan ahora ser reiteradas. De manera similar, el estimador de movimiento 528 y el compensador de movimiento 530 funcionan de la misma manera que el estimador de movimiento 128 y el compensador de movimiento 130, descritos con anterioridad, excepto por los conjuntos de filtros que utilizan. En este caso, el método es el filtrado de movimiento, y solamente este subsistema se analiza en detalle.

45

50

A diferencia de H.264, que utiliza un conjunto de filtros fijo, el codificador 500 calcula un conjunto de filtros a partir de los datos. Este cálculo tiene lugar en el ordenador 558 del conjunto de filtros iterativos de subpíxeles (Wiener), utilizando las ecuaciones de Wiener Hopf descritas con anterioridad. El proceso comienza con el uso de un conjunto de filtros predeterminado para calcular la posición más baja de Suma de Diferencias Absolutas ("SAD") para cada bloque, que refleja la mejor posición de coincidencia de subpíxeles. En la siguiente iteración, utilizando estas

posiciones de subpíxeles, se calcula un nuevo conjunto de filtros, que cuando se utiliza proporciona ubicaciones de subpíxeles actualizadas. El proceso es iterativo y, por lo tanto, pueden ser necesarias varias iteraciones (4-20) para que el proceso sea convergente. El proceso es intensivo en cálculo debido a la complejidad de la evaluación en cada iteración. En algunos casos, es posible limitar el número de iteraciones, intercambiando complejidad por rendimiento.

El conjunto de filtros de movimiento adaptativo calculado (*amf*) se proporciona al interpolador de subpíxeles 531 en el predictor de movimiento compensado 530. Este conjunto de filtros (*amf*) también necesita codificarse y enviarse a través del flujo binario al decodificador. Puesto que el conjunto de filtros (*amf*) puede utilizar hasta 99 coeficientes de filtro, es importante una codificación eficaz del conjunto de filtros. Las técnicas convencionales varían desde el envío de un conjunto de filtros para las posiciones más significativas, la codificación diferencial de coeficientes (*dmf*) (generada por el diferenciador 560), la supresión de coeficientes que no cambiaron mucho desde la última vez que se enviaron, y similares. Suponiendo que los coeficientes de filtro se puedan codificar de manera eficiente, se pueden enviar múltiples conjuntos de filtros por imagen con el objetivo de elegir entre dos o más conjuntos de filtros para una mayor eficiencia; esta operación se realiza en el particionador espacial y el codificador de particiones 505, que genera un mapa de particiones (*ptmap*). El optimizador de distorsión de frecuencia 508 se utiliza a menudo para asegurar que incluso cuando se utilizan múltiples filtros y la sobrecarga asociada, la ganancia seguirá obteniéndose.

La Figura 6 muestra un diagrama de bloques de un decodificador intertramas 600 de la técnica anterior que emplea un predictor de movimiento compensado que utiliza filtrado de movimiento adaptativo generalizado. El decodificador de entropía 603, la transformada inversa 615, el cuantificador inverso 613 y el ensamblador de tramas y el procesador de filtrado de desbloqueo 625 funcionan de la misma manera que los componentes correspondientes del decodificador 200, tal como se muestra en la Figura 2 y que se describió con anterioridad, y sus descripciones no necesitan ser de nuevo reiteradas. De manera similar, el predictor de movimiento compensado 630 funciona de la misma manera que el predictor de movimiento compensado 230, descrito con anterioridad, excepto por los conjuntos de filtros utilizados. En este caso, el método es el filtrado de movimiento, y solamente este subsistema se analiza en detalle.

El decodificador de entropía 603 decodifica una palabra de código (*dmf*) que decodifica el valor diferencial del filtro de movimiento al que se añade un conjunto de filtros que representa la predicción en el sumador 631, dando como resultado un conjunto de filtros de coeficientes (*amf*). Este conjunto de filtros (*amf*) se proporciona al interpolador de subpíxeles 629 en el predictor de movimiento compensado 630. Además, en caso de que se utilicen varios filtros, se puede decodificar un mapa de partición (*ptmap*) para que el filtro (*amf*) se pueda aplicar a la parte correcta de la imagen.

Las Figuras 7a-f muestran varios modos de partición para dividir una imagen. Por ejemplo, la Figura 7a representa una partición basada en imágenes donde los códigos de filtro solamente se cambian imagen por imagen. La Figura 7b muestra el particionamiento basado en cortes o grupos de cortes de manera que se aplica un primer conjunto de filtros al grupo de cortes 'A', y se aplica un segundo conjunto de filtros al grupo de cortes 'B'. La Figura 7c representa un mapa de bloques que identifica qué filtro utilizar para el tipo de bloque 'A' y qué filtro utilizar para el tipo de bloque 'B'. La Figura 7d muestra la división de una imagen en mosaicos y la división adicional de mosaicos en particiones de bi-tree de modo que varios cortes de bi-tree basados en la optimización de la distorsión de la tasa ("RDO") se etiquetan como 'A' o 'B', lo que indica que pueden utilizar correspondientemente un primer o un segundo conjunto de filtros. Asimismo, la Figura 7e muestra la división de una imagen en mosaicos y la división adicional de un mosaico en dos tipos de particiones de quad-tree etiquetadas como 'A' o 'B', de modo que puedan utilizar correspondientemente un primer o un segundo conjunto de filtros. Por último, la Figura 7f muestra las zonas que resultan de la fusión de cortes, la fusión de bloques, la fusión de bi-tree o la fusión de particiones de quad-tree, o simplemente por segmentación espacial.

La Figura 8 muestra un diagrama de bloques de un codificador intertramas 800 que emplea un predictor de movimiento compensado que utiliza un codificador de libro de códigos de filtrado de movimiento de conformidad con una forma de realización. El indexador de bloques 801, la transformada directa 835, el cuantificador 838, el cuantificador inverso 840, la transformada inversa 843, el ensamblador de tramas y el procesador de filtrado de desbloqueo 848 y el codificador de entropía 818 funcionan de la misma manera que los componentes correspondientes del codificador 100, tal como se muestra en la Figura 1 y descrito con anterioridad, y sus descripciones no necesitan de nuevo ser reiteradas. De manera similar, el estimador de movimiento 828 y el compensador de movimiento 830 funcionan de la misma manera que el estimador de movimiento 128 y el compensador de movimiento 130 descritos con anterioridad, excepto por los conjuntos de filtros que utilizan. En este caso, el método es el filtrado de movimiento, y solamente este subsistema se analiza en detalle.

En comparación con los codificadores de filtrado de movimiento adaptativo existentes, el codificador de filtrado de movimiento basado en libro de códigos 800 selecciona un conjunto de filtros de un libro de códigos existente en lugar de calcular el filtro de movimiento explícitamente. En una forma de realización, se utilizan dos tipos de libros de códigos, uno que almacena conjuntos de filtros de subpíxeles 860 y otro que almacena filtros de posición entera 863. El procesador de búsqueda de libro de códigos de conjunto de filtros MC 858 permite una búsqueda eficiente de los respectivos libros de códigos produciendo índices de libros de códigos de posición entera y subpíxeles, (*mcfsi*) y (*mcfsii*), que se refieren correspondientemente al conjunto de filtros de subpíxeles (*mcfs*) y al filtro de posición entera (*mcfi*).

En varias formas de realización, el conjunto de filtros de subpíxeles (*mcfs*) y el filtro de posición entera (*mcfi*) se pueden enviar en una imagen (una sola vez) o en una partición espacial (varias veces por imagen). En varias formas de realización, se puede utilizar cualquier tipo de particiones espaciales, tal como, por ejemplo, corte, tamaño de bloque variable, bloque, bi-tree, quad-tree, zona, mosaico y similares (véase, por ejemplo, las Figuras 7a-f). Esta partición se realiza mediante el particionador espacial y el codificador de particiones 805. Los detalles de la partición, tal como el número de particiones que se enviarán, se deciden en función del análisis de distorsión de la tasa mediante el analizador de compensación de tasa de bits/distorsión 808, así como el analizador de contenido de escena 803. La mejor coincidencia del conjunto de filtros de subpíxeles y del conjunto de filtros de píxeles enteros se envían en el flujo binario al decodificador. El conjunto de filtros de subpíxeles (*mcfs*) y el filtro de posición entera (*mcfi*) se proporcionan al procesador de filtrado MC 865 para generar predicción de movimiento compensado en el estimador de movimiento/predicador de movimiento compensado 828. En algunas formas de realización, el proceso también puede ser eficiente en la búsqueda, así como en el coste de bits para la representación de filtros/conjuntos de filtros por índices.

La Figura 9 muestra un diagrama de bloques de un decodificador intertramas 900 que emplea un predictor de movimiento compensado que utiliza un decodificador de libro de códigos de filtrado de movimiento de conformidad con una forma de realización. El decodificador de entropía 903, la transformada inversa 915, el cuantificador inverso 913 y el ensamblador de tramas y el procesador de filtrado de desbloqueo 925 funcionan de la misma manera que los componentes correspondientes del decodificador 200, tal como se muestra en la Figura 2 y que se describió con anterioridad, y sus descripciones no necesitan de nuevo reiterarse. De manera similar, el predictor de movimiento compensado 930 funciona de la misma manera que el predictor de movimiento compensado 230, descrito con anterioridad, excepto por los conjuntos de filtros utilizados. En este caso, el método es el filtrado de movimiento, y solamente este subsistema se analiza en detalle.

El decodificador de entropía 903 decodifica la entropía del flujo binario 901 para obtener el índice de filtro de movimiento de subpíxeles del libro de códigos (*mcfsi*) y/o del índice de filtro de movimiento de posición entera (*mcfii*). Estos índices (*mcfsi*) y (*mcfii*) se utilizan para acceder al conjunto de filtros de subpíxeles deseados (*mcfs*) y/o al filtro de posición entera (*mcfi*) del libro de códigos de subpíxeles 933 y/o del libro de códigos de posición entera 935. El conjunto de filtros de subpíxeles resultante *mcfs* y/o el filtro de inter-píxeles *mcfi* se proporcionan al procesador de filtrado MC 929 que realiza la interpolación de subpíxeles y/o el filtrado de píxeles enteros para generar predicción de movimiento compensado (*pred*). En algunas formas de realización, los indicadores (*filt_sent*) y (*intfilt_sent*) pueden decodificarse de manera adicional mediante el decodificador de entropía 903, indicando si el flujo binario 901 incluye índice de filtro de movimiento de subpíxeles (*mcfsi*) e índice de filtro de movimiento de posición entera (*mcfii*), respectivamente.

La Tabla 1 muestra dos conjuntos de filtros de muestra (id=0 e id=1) de un libro de códigos "base" de subpíxeles a modo de ejemplo. En una forma de realización, el libro de códigos de subpíxeles "base" puede incluir 16 conjuntos de filtros de subpíxeles "base" (cada uno incluye 15 o 16 filtros de posición de subpíxeles en el caso de filtrado de 1/4 píxel). Los filtros de posición de subpíxeles individuales de los 16 conjuntos de filtros de subpíxeles "base" se pueden utilizar en diferentes combinaciones predeterminadas para crear un libro de códigos "virtual". En una forma de realización, el libro de códigos "virtual" puede incluir 256 conjuntos de filtros de subpíxeles. En otras formas de realización, los libros de códigos "base" y "virtual" pueden incluir más o menos conjuntos de filtros.

Tabla 1

Subconjunto del libro de códigos base de posiciones a-o de subpíxeles (1/4 píxel)									
id del conjunto de filtros = 0									
<i>a₀</i>	8	-31	217	83	-33	12			
<i>b₀</i>	14	-39	153	153	-39	14			
<i>c₀</i>	15	-30	68	216	-15	2			
<i>d₀</i>	-12	20	187	83	-25	3			
<i>e₀</i>	10	-41	202	112	-34	7			
<i>f₀</i>	23	-123	89	-278	1004	-60	421	1	-53
<i>g₀</i>	15	-56	186	128	-24	7			
<i>h₀</i>	-1	-19	148	148	-19	-1			
<i>i₀</i>	-13	21	-73	-8	6	-141	890	469	-127
<i>j₀</i>	19	-20	16	-55	48	-130	103	-214	745
<i>k₀</i>	41	3	-169	-61	-1	-115	954	466	-94
<i>l₀</i>	-2	-6	77	210	-24	1			
<i>m₀</i>	6	-39	202	103	-25	9			

Subconjunto del libro de códigos base de posiciones a-o de subpíxeles (¼ píxel)									
id del conjunto de filtros = 0									
n_0	-11	-74	118	-369	1071	66	332	-15	-94
o_0	19	-53	162	169	-56	15			
id del conjunto de filtros = 1									
a_1	39	25	59	72	26	35			
b_1	38	27	63	63	27	38			
c_1	-58	-3	27	112	57	121			
d_1	-121	82	294	91	-61	-29			
e_1	-23	69	42	147	-2	23			
f_1	-71	73	131	226	470	212	300	-172	-145
g_1	18	203	237	-46	246	-402			
h_1	-430	338	220	220	338	-430			
i_1	-525	-42	196	101	-92	218	244	461	463
j_1	-884	-102	-53	378	105	240	270	138	420
k_1	128	4230	-1742	1603	-3906	766	-1538	1578	-93
l_1	-232	176	76	109	233	-106			
m_1	37	47	85	121	72	-106			
n_1	14	99	300	166	355	211	331	-206	-246
o_1	49	16	55	61	8	67			
id del conjunto de filtros = 2									
$a_{3..}$	:	:	:	:	:	:	:	:	:

De los conjuntos de filtros a modo de ejemplo ilustrados en la Tabla 1, el id de conjunto de filtros = 0 corresponde a una escena nítida, y el id de conjunto de filtros = 1 corresponde a una escena más borrosa.

- 5 En algunas formas de realización, un libro de códigos de subpíxeles tal como el que se muestra (en parte) en la Tabla 1 se puede caracterizar de manera equivalente por incluir una multiplicidad de filtros de compensación de movimiento agrupados en una pluralidad de grupos de posiciones de subpíxeles que corresponden, respectivamente, a la pluralidad de posiciones de subpíxeles, comprendiendo cada uno de entre la pluralidad de grupos de posiciones de subpíxeles una pluralidad de filtros de compensación de movimiento adecuados para interpolar bloques de una imagen en una correspondiente de entre la pluralidad de posiciones de subpíxeles. Por ejemplo, el libro de códigos que se muestra en la Tabla 1 se puede caracterizar por incluir 15 o 16 grupos de filtros de compensación de movimiento correspondientes a las posiciones de subpíxeles {a-o}, por ejemplo:

- grupo {a} de posiciones de subpíxeles: {a₀, a₁, a₂..a₁₅};
- grupo {b} de posiciones de subpíxeles: {b₀, b₁, b₂..b₁₅};
- grupo {o} de posiciones de subpíxeles: {o₀, o₁, o₂..o₁₅}.

- 20 El libro de códigos se puede caracterizar, además, por incluir varios (p. ej., 256) subconjuntos predeterminados de filtros de compensación de movimiento, uno de cada grupo de posiciones de subpíxeles, p. ej.:

- id de subconjunto=0: {a₀, b₀, c₀, d₀, e₀, f₀, g₀, h₀, i₀, j₀, k₀, l₀, m₀, n₀, o₀};
- id de subconjunto=1: {a₁, b₁, c₁, d₁, e₁, f₁, g₁, h₁, i₁, j₁, k₁, l₁, m₁, n₁, o₁};
- id de subconjunto=15: {a₁₅, b₁₅, c₁₅, d₁₅, e₁₅, f₁₅, g₁₅, h₁₅, i₁₅, j₁₅, k₁₅, l₁₅, m₁₅, n₁₅, o₁₅};
- id de subconjunto=16: {a₀, b₅, c₇, d₉, e₄, f₄, g₁₂, h₆, i₇, j₂, k₄, l₁₁, m₈, n₀, o₁₀};
- id de subconjunto=255: {a₁₀, b₆, c₁₂, d₉, e₂, f₂, g₃, h₁₀, i₃, j₅, k₁, l₈, m₁₁, n₆, o₁₀}.

- 35 En algunas formas de realización, el libro de códigos puede incluir, además, un grupo de filtros de posición entera, por ejemplo, grupo de posición entera {0}: {0₀, 0₁, 0₂..0₁₅}. En dichas formas de realización, los subconjuntos predeterminados de filtros de compensación de movimiento también pueden incluir un filtro de posición entera, por

ejemplo: id de subconjunto=0: {0₀, a₀, b₀, c₀, d₀, e₀, f₀, g₀, h₀, i₀, j₀, k₀, l₀, m₀, n₀, o₀}; id de subconjunto=1: {0₁, a₁, b₁, c₁, d₁, e₁, f₁, g₁, h₁, i₁, j₁, k₁, l₁, m₁, n₁, o₁}; y así sucesivamente.

En algunas formas de realización, los filtros de posición entera pueden incluirse en un libro de códigos de posición entera. La Tabla 2 muestra dos filtros de posición entera de muestra (id=0 e id=1) de un libro de códigos de posición entera a modo de ejemplo. En una forma de realización, el libro de códigos de posición entera puede incluir 48 filtros de posición entera. En otras formas de realización, el libro de códigos de posición entera puede incluir más o menos filtros. En algunas formas de realización, los filtros de posición entera también se pueden incluir dentro de los conjuntos de filtros de subpíxeles del libro de códigos de filtros de subpíxeles, en lugar de, o además del, libro de códigos de posición entera.

Tabla 2

Subconjunto del libro de códigos de posición entera									
id filtro =0									
O	-58	162	48	51	309	103	15	252	142
id filtro =1									
O	-32	10	54	196	519	280	65	31	-99
id filtro =2									
:	:	:	:	:	:	:	:	:	:

La Figura 10 muestra una rutina 1000 "fuera de línea" para generar un libro de códigos de filtro de movimiento o libros de códigos para ubicaciones de subpíxeles y de píxeles enteros. En el bloque 1005, la rutina 1000 obtiene un conjunto de prueba que incluye una o más secuencias de vídeo de muestra. En algunas formas de realización, las secuencias de vídeo se seleccionan para tener combinaciones de diferentes grados de movimiento y una diversidad de textura casi uniforme. En una forma de realización, las secuencias de prueba se eligen de manera que contengan zonas similares a texturas, que en condiciones normales necesiten un filtrado más preciso para lograr la ganancia de codificación deseada. En una forma de realización, se pueden emplear varias secuencias CIF estándar tal como secuencias de vídeo de muestra.

Comenzando en el bloque de bucle inicial 1010, la rutina 1000 procesa cada secuencia de vídeo en el conjunto de prueba. Comenzando en el bloque de bucle inicial 1015, la rutina 1000 procesa una o más imágenes o tramas individuales de la secuencia de vídeo actual. En el bloque 1020, la rutina 1000 aplica el filtrado de movimiento adaptativo basado en filtros de Wiener para calcular un conjunto de filtros totalmente adaptado a la imagen o trama actual. En algunas formas de realización, para obtener mejores conjuntos de filtros, se pueden realizar dos o más iteraciones, lo que significa que el cálculo de los coeficientes de filtro y la actualización de las ubicaciones de los *subpíxeles* se pueden realizar de forma iterativa dos o más veces (la primera vez que las colaciones de *subpíxeles* se establecen en resultados de filtro estándar, y en cada iteración subsiguiente las posiciones se establecen en las obtenidas con el conjunto de filtros previamente calculado). En las formas de realización de ¼ píxel, un conjunto de filtros así generado consta de 16 filtros, puesto que se calcula un filtro por cada ubicación de ¼ píxel. En una forma de realización, cada filtro individual en el conjunto de filtros generado tiene 6 o 9 tomas dependiendo de la ubicación de ¼ píxel.

Al finalizar el bloque de bucle 1025, la rutina 1000 itera de nuevo al bloque 1015 para procesar otra imagen o trama individual de la secuencia de vídeo actual (si la hubiere). En el bloque 1030, la rutina 1000 selecciona un conjunto de filtros representativo para la secuencia de vídeo actual a partir de los conjuntos de filtros generados. En una forma de realización, seleccionar un conjunto de filtros representativo puede incluir aplicar todos los conjuntos de filtros calculados a cada imagen o trama de la secuencia de vídeo y seleccionar un conjunto de filtros que tenga una ganancia de codificación alta (o la más alta) (es decir, que tenga un SAD bajo o el SAD más bajo). En algunas formas de realización, el conjunto representativo puede seleccionarse de conformidad con un subconjunto de imágenes o tramas de la secuencia de vídeo actual (por ejemplo, las primeras *N* tramas).

Al finalizar el bloque de bucle 1035, la rutina 1000 itera de nuevo al bloque 1010 para procesar la siguiente secuencia de vídeo en el conjunto de prueba (si lo hubiere). En una forma de realización, después de que se hayan procesado todas las secuencias de vídeo, se pueden haber generado y seleccionado hasta 100 o más conjuntos de filtros representativos. En el bloque 1038, la rutina 1000 refina, de manera opcional, el grupo recogido de conjunto de filtros representativos. Por ejemplo, en una forma de realización, se puede emplear un algoritmo de recorrido aleatorio (o un algoritmo de optimización similar) para probar combinaciones aleatorias de filtros en algunas o todas las secuencias de vídeo para mejorar aún más la ganancia de codificación general proporcionada por el grupo recogido de conjunto de filtros refinados en el transcurso de horas o incluso días.

En el bloque 1040, la rutina 1000 selecciona un grupo "base" de conjuntos de filtros de subpíxeles que tienen una ganancia de codificación alta o la más alta entre el grupo, de manera opcional, refinado de conjuntos de filtros. En el bloque 1045, la rutina 1000 almacena el grupo "base" seleccionado de conjuntos de filtros en un libro de códigos de subpíxeles.

En el bloque 1050, la rutina 1000 deriva un libro de códigos "virtual" a partir de los filtros individuales del grupo "base" seleccionado de conjuntos de filtros. En una forma de realización, se puede emplear un algoritmo de recorrido aleatorio (o un algoritmo de optimización similar) para probar varias combinaciones de filtros individuales del grupo "base" seleccionado de conjuntos de filtros. En una forma de realización, se pueden derivar 256 combinaciones diferentes de filtros del grupo "base" seleccionado de conjuntos de filtros. Las combinaciones derivadas se almacenan para crear un libro de códigos "virtual" que puede derivarse de los filtros del grupo "base" seleccionado de conjuntos de filtros.

En el bloque 1055, se selecciona un grupo de filtros de posición entera de entre el grupo recogido de conjuntos de filtros opcionalmente refinados. En una forma de realización, los filtros de posición entera se pueden seleccionar para que tengan una alta ganancia de codificación y difieran entre sí. En una forma de realización, se pueden seleccionar 48 filtros de posición entera para el grupo. En el bloque 1060, el grupo seleccionado de filtros de posición entera se almacena en un libro de códigos de posición entera. Habiendo generado y almacenado un libro de códigos de subpíxeles y un libro de códigos de posición entera, la rutina 1000 finaliza en el bloque 1099.

La Figura 11 muestra una rutina 1100 "online" para buscar en un libro de códigos un conjunto de filtros de movimiento de conformidad con una forma de realización. En el bloque 1103, la rutina 1100 obtiene una imagen o trama de una secuencia de vídeo. Comenzando con la apertura del bloque de bucle 1105, la rutina 1100 procesa cada bloque de contenido de imagen de la imagen.

En el bloque 1108, la rutina 1100 determina una estimación de movimiento de posición entera para el bloque actual de contenido de imagen. En una forma de realización, un vector de movimiento de posición entera se calcula mediante búsqueda completa o reducida utilizando un tamaño de búsqueda de movimiento dado. En el bloque 1110, la rutina 1100 calcula una ganancia de codificación (por ejemplo, determina SAD) para la estimación de movimiento de posición entera determinada.

En el bloque 1113, la rutina 1100 determina una estimación de movimiento de subpíxeles utilizando un conjunto de filtros predeterminado (por ejemplo, el conjunto de filtros fijo H.264, o cualquier otro conjunto de filtros para su uso como referencia). Se buscan ubicaciones de subpíxeles interpolados para ubicar una con la ganancia de codificación más alta (SAD más baja). Dicho de otro modo, el conjunto de filtros predeterminado se aplica al bloque de contenido de imagen y se elige la ubicación que produce el SAD más pequeño. La ubicación a la que corresponde ese conjunto de filtros se utiliza como el vector de movimiento de subpíxeles estimado. En el bloque 1115, la rutina 1100 guarda (al menos temporalmente) la ganancia de codificación general para el filtro predeterminado.

En el bloque 1116, la rutina 1100 carga un libro de códigos de subpíxeles "base", que incluye, por ejemplo, 16 conjuntos de filtros, incluyendo cada conjunto de filtros, por ejemplo, 15 filtros de posición de subpíxeles (uno para cada posición de subpíxel). A partir del bloque 1118, la rutina 1100 procesa cada conjunto de filtros. En el bloque 1120, la rutina 1100 calcula una ganancia de codificación para el conjunto de filtros actual. Usando la estimación del movimiento de la posición entera como un desplazamiento, la rutina 1100 determina un SAD para cada posición de subpíxeles utilizando el filtro de posiciones de subpíxeles correspondiente del conjunto de filtros actual. Al finalizar el bloque de bucle 1023, la rutina 1100 itera de nuevo al bloque 1118 para procesar el siguiente conjunto de filtros (si lo hubiere).

En el bloque 1023, la rutina 1100 ensambla ganancias de codificación para N (p. ej., 256) conjuntos de filtros "virtuales" utilizando las ganancias de codificación calculadas para filtros de posiciones de subpíxeles individuales en cada uno de los conjuntos de filtros. En una forma de realización, los conjuntos de filtros "virtuales" incluyen varias combinaciones de filtros de posiciones de subpíxeles individuales de los conjuntos de filtros del libro de códigos de subpíxeles "base". Por tanto, las ganancias de codificación para cada conjunto de filtros "virtual" puede ensamblarse utilizando las ganancias de codificación de filtros de posición de subpíxeles individuales ya calculadas (en el bloque 1120).

En el bloque 1128, la rutina 1100 itera de nuevo al bloque 1105 para procesar el siguiente bloque de contenido de imagen (si lo hubiere) de la imagen.

En el bloque 1133, la rutina 1100 establece un modo de partición espacial. Dependiendo del modo de funcionamiento, se puede utilizar una división de la imagen basada en tramas, basada en cortes, basada en bloques, basada en segmentos o en una división similar. Comenzando en el bloque de bucle de apertura 1135, la rutina 1100 procesa cada partición espacial de la imagen. En el bloque 1138, la rutina 1100 selecciona entre los N conjuntos de filtros de subpíxeles "virtuales" un conjunto de filtros de subpíxeles que tiene una alta ganancia de codificación para la partición actual. En una forma de realización, el mejor conjunto de filtros se elige con respecto a la SAD más pequeña/la ganancia de codificación más grande. En algunas formas de realización, el modo de funcionamiento más bajo utiliza una división basada en tramas. Los siguientes dos modos utilizan cortes. El modo de corte inferior divide una trama

en cuatro bandas/cortes horizontales iguales (donde el último corte podría tener un tamaño más pequeño dependiendo del tamaño de la trama). Por otro lado, el modo de corte superior utiliza cortes de bloque de 12x4 16x16 (de nuevo, los cortes en la parte inferior y en el borde derecho pueden ser más pequeños debido al tamaño de la trama). El resto de los modos utilizan los segmentos/zonas que se derivan de la fusión de bloques en grupos de bloques donde el mismo conjunto de filtros produce un SAD bajo. Estas zonas, en cierto sentido, corresponden a particiones de movimiento. Hay dos umbrales que se utilizan en el proceso, y cada modo tiene una selección diferente de valores de umbral que pueden producir menos o más segmentos/zonas. La trama se divide primero en bloques/supermosaicos. El primer umbral se utiliza para determinar cómo dividir un supermosaico de 64x64 en mosaicos de 32x32, mientras que el segundo se utiliza para determinar cuándo fusionar los bloques de mosaicos a través de los límites del supermosaico. En ambos casos, la decisión de fusión o división se toma si la diferencia de SAD de los dos bloques/objetos en cuestión está por debajo o por encima del umbral dado. La división de un supermosaico en mosaicos se codifica con patrones 2x2. Los diferentes modos de funcionamiento admiten patrones básicos 3 o 7 ampliados. Los 3 patrones básicos son sin división, con división horizontal y con división vertical, mientras que el conjunto extendido también incluye las cuatro divisiones de 3 vías. A continuación, la siguiente etapa del proceso es fusionar mosaicos que tocan los bordes del supermosaico con los próximos si hay un conjunto de filtros común que resulte en un SAD lo suficientemente pequeño (basado en la aplicación del segundo umbral). Los bits de fusión se codifican en el flujo binario además de los códigos de patrón de longitud variable ("VLC") de modo que el decodificador pueda recuperar completamente los mismos segmentos espaciales.

Habiendo seleccionado un conjunto de filtros de subpíxeles que tiene una ganancia de codificación alta para la partición actual, en el bloque 1140, la rutina 1100 codifica en un flujo binario un índice u otro código (por ejemplo, `subpelfilt_code`) correspondiente al conjunto de filtros de subpíxeles seleccionado. En varias formas de realización, se puede utilizar una tabla VLC fija, una tabla de codificación de Huffman adaptativa o similar para codificar el índice u otro código. Si se utiliza Huffman adaptativa, los códigos se calculan, en una forma de realización, en función de las frecuencias de los índices de las tramas anteriores, ya que se observó en experimentos que existe una redundancia temporal, y los mismos conjuntos de filtros tienden a aparecer en las tramas próximas.

En el bloque 1141, la rutina 1100 carga un libro de códigos de posición entera que incluye un número de (por ejemplo, 48) filtros de posición entera. En el bloque de decisión 1143, la rutina 1100 determina si la estimación del movimiento de enteros (determinada en el bloque 1108) proporciona una ganancia de codificación más alta (menor que SAD) que el conjunto de filtros de subpíxeles seleccionado. Si no es así, entonces la rutina 1100 procede a finalizar el bloque de bucle 1153, donde la rutina 1100 itera de nuevo al bloque 1135 para procesar la siguiente partición espacial (si la hubiere).

Sin embargo, si la estimación del movimiento de enteros (determinada en el bloque 1108) proporciona una ganancia de codificación más alta (SAD más bajo) que el conjunto de filtros de subpíxeles seleccionado, entonces en el bloque 1145, la rutina 1100 selecciona desde el libro de códigos de posición entera un filtro de posición entera que tiene una ganancia de codificación alta (o la más alta) para la partición espacial actual. En el bloque de decisión 1148, la rutina 1100 determina si el filtro de posición entera seleccionado proporciona la ganancia de codificación más alta (SAD más pequeña). Si es así, entonces en el bloque 1150, la rutina 1100 codifica al flujo binario un índice u otro código (por ejemplo, `intpelfilt_code`) correspondiente al filtro de posición entera seleccionado. En algunas formas de realización, un indicador u otra cabecera de 1 bit también se codifica en el flujo binario para especificar que el filtro de posición entera seleccionado del libro de códigos de posición entera produce una ganancia de codificación más alta que la del filtro de posición entera estándar.

Al finalizar el bloque de bucle 1153, la rutina 1100 itera de nuevo al bloque 1135 para procesar la siguiente partición espacial (si la hubiere). Una vez que se procesan todas las particiones espaciales, la rutina 1100 termina en el bloque 1199. En varias formas de realización, el método descrito puede aplicarse por imagen y un índice a un conjunto de filtros enviado para una imagen completa, una zona o un segmento, un grupo de cortes, para cada una de las particiones quad-tree o bi-tree de una trama o de un mosaico, en bloques de grupo de una trama, en mosaicos o en fragmentos de un mosaico.

La Figura 12 ilustra una sintaxis de flujo binario 1200 a modo de ejemplo, de conformidad con varios codificadores y decodificadores de filtro de libro de códigos de compensación de movimiento y que se describen aquí de diversas formas. La sintaxis de flujo binario 1200 comienza con la cabecera de 3 bits `mfc_hdr` 1201. El indicador 1202 de imagen de un único bit indica si el filtrado basado en imágenes tendrá lugar o no. En caso afirmativo, se omite el siguiente elemento de sintaxis opcional `numpt` 1203; si la respuesta es negativa, se envía el código numérico de 3 bits 1203. El campo numérico opcional de 3 bits 1203 indica un número de particiones (hasta ocho).

El indicador de envío de mapas de 1 bit opcional 1204 indica si se envía un mapa de partición para indicar dónde se aplica el filtro o no, o de forma alternativa, para cambiar entre múltiples filtros. El campo de mapa de puntos de n bits 1205 opcional incluye el mapa de partición (si lo hubiere).

El indicador de envío de `ninot` filtrado 1 bit 1206 indica si se incluye un código de filtro de subpíxeles en el flujo binario. Si el indicador 1206 está establecido, entonces se envía `subpelfilt_code` 1208 de 8 bits para identificar el conjunto de filtros de subpíxeles que se utilizará; de lo contrario, no se envía `subpelfilt_code` 1208.

El indicador `intfilt_sent` 1210 indica si un código de filtro de posición entera está incluido en el flujo binario. Cuando se establece en '1', `intfilt_sent` 1210 indica que se envía `intpelfilt_code` de 6 bits 1211 para identificar el filtro de posición entera que se utilizará. De lo contrario, no se envía `intpelfilt_code` 1211.

La secuencia de códigos `filt_sent` 1206, `subpelfilt_code` 1208, `intfilt_sent` 1210 e `intpelfilt_code` 1211 ocurre una sola vez si se utiliza solamente el modo de 1 partición (imagen solamente); de lo contrario, la secuencia se repite `numpt` veces 1203.

La Figura 13a ilustra una secuencia de imágenes o tramas 1301-1308 dentro de un flujo de vídeo, que incluye I-tramas 1301, 1308 (imágenes codificadas de manera independiente); P-tramas 1304, 1306 (imágenes codificadas predichas unidireccionalmente); y B-tramas 1302-1303, 1305, 1307 (imágenes codificadas predichas bidireccionalmente), como es normal en la codificación de vídeo según cualquiera de las normas MPEG. La estructura de la imagen ilustrada también muestra un número variable de B-tramas, aunque ese hecho en sí mismo no es una clave para nuestra descripción sobre el filtrado de movimiento. Más pertinente, mientras que I-imagen 1301, 1308 no necesita información de conjunto de filtros de movimiento, P-imagen 1304, 1306 necesita uno o más conjuntos de filtros, mientras que B-imagen 1302-1303, 1305, 1307 necesitan dos (debido a referencias bidireccionales) o más conjuntos de filtros.

La Tabla 3 indica un coste de bits de codificación a modo de ejemplo para codificar filtros de libro de códigos de compensación de movimiento de conformidad con varias combinaciones de tipo de imagen y número de particiones (`numpt`), de conformidad con varias formas de realización, incluyendo los modos de filtrado de movimiento (por ejemplo, 'subpíxel', y 'int y subpíxel'), para los tipos de imagen P y B. Para reducir el coste de bits de las imágenes B, se introducen 2 modos adicionales (por ejemplo, 'imagen subpíxel' y 'imagen int subpíxel'). En general, se puede observar que para las P-imágenes, el conjunto de filtros de señalización de subpíxeles solamente toma entre 13 y 43 bits (para hasta 4 particiones), un conjunto de filtros de señalización tanto int como subpíxeles solamente toma entre 19 y 67 bits, y B- imágenes para subpíxeles entre 22 y 79 bits, e B-imágenes para int y subpíxeles entre 34 y 127 bits. (Los costes de bits de codificación a modo de ejemplo que se muestran en la Tabla 3 excluyen los costes de bits para mapas de particiones espaciales tales como zonas, segmentos, mosaicos o fragmentos de mosaicos, imagen bi-tree o quad-tree, mosaico bi-tree o quad-tree, y/o bloques fusionados).

Tabla 3

Tipo de imagen (y modos de filtro de movimiento)	Número de conjuntos de filtros y coste de bits de codificación							
	numpt=1		numpt=2		numpt=3		numpt=4	
	Número conjuntos filtros	Coste aproximado de bits *	Número conjuntos filtros	Coste aproximado de bits *	Número conjuntos filtros	Coste aproximado de bits *	Número conjuntos filtros	Coste aproximado de bits *
I	-	-	-	-	-	-	-	-
P: subpíxel	1	13	2	25	3	34	4	43
P: int y subpíxel	1	19	2	37	3	52	4	67
B: imagen, subpíxel	2	22	2	22	2	22	2	22
B: imagen, int y subpíxel	2	34	2	34	2	34	2	34
B: subpíxel	2	22	4	43	6	61	8	79
B: int y subpíxel	2	34	4	67	6	97	8	127

La Figura 13b ilustra una secuencia de imágenes o tramas 1301-1308 dentro de un flujo de vídeo, incluyendo las I-tramas 1301, 1308, que están divididas en tres particiones a modo de ejemplo (`numpt`=3); P-tramas 1304, 1306, que están divididas en tres particiones a modo de ejemplo (`numpt`=3); y B-tramas 1302-1303, 1305, 1307, que están divididas en dos particiones a modo de ejemplo (`numpt`=2). Tal como se ilustra en la Figura 13b, las particiones para el filtrado de movimiento pueden ser totalmente adaptables y, en base a una imagen, pueden operar en cualquiera de los modos (imagen, zonas, cortes, mosaicos, fragmentos fusionados de un mosaico, imagen bi-tree o mosaico fusionado bi-tree, imagen quad-tree o mosaico fusionado quad-tree, bloques fusionados o similares). Por ejemplo, la Figura 13b muestra que la primera P-imagen 1304 consta de 3 cortes, mientras que la segunda B-imagen 1303 está codificada en bi-tree y consta de 2 particiones. Además, la segunda P-imagen 1306 está dividida en una matriz de 3x3 mosaicos grandes, mientras que la tercera B-trama 1305 está codificada con una división de bi-tree. Los siguientes ejemplos de la Figura 13b, para el caso del filtrado de movimiento de libro de códigos, pueden calcularse los recuentos de bits del filtro de movimiento para cada imagen en la estructura de imagen mostrada.

La Figura 14 ilustra un sistema de codificación/decodificación de vídeo 1400 a modo de ejemplo de conformidad con una forma de realización, el sistema 1400 incluye una fuente de vídeo no codificado 1415, un dispositivo de codificación

1500 (véase Figura 15, descrito a continuación), un dispositivo de decodificación 1600 (véase Figura 16, descrito a continuación) con una pantalla externa opcional 1425 y una fuente de vídeo codificado opcional 1420. En algunas formas de realización, uno o ambos del dispositivo de codificación 1500 y/o del dispositivo de decodificación 1600 pueden estar conectados a una red 1450. En algunas formas de realización, el dispositivo de decodificación 1600 y el dispositivo de codificación 1500 pueden comprender un único dispositivo. En algunas formas de realización, puede estar presente más de un dispositivo de codificación 1500; por ejemplo, un dispositivo de codificación puede realizar procesos de generación de libros de códigos "fuera de línea", mientras que otro dispositivo de codificación puede realizar procesos de codificación "online", tal como se describe a lo largo de esta descripción.

En algunas formas de realización, también pueden estar presentes otros servidores y/o dispositivos (no ilustrados). Por ejemplo, en algunas formas de realización, un servidor de distribución (no ilustrado) puede distribuir vídeo codificado al dispositivo de decodificación 1600 a través de la red 1450.

En algunas formas de realización, el dispositivo de codificación 1500 puede comunicarse con la fuente de vídeo no codificado 1415 a través de la red 1450, una red de área de almacenamiento ("SAN"), un bus serie de alta velocidad y/o mediante otra tecnología de comunicación adecuada. De forma similar, en algunas formas de realización, el dispositivo de decodificación 1600 puede comunicarse con la fuente de vídeo codificado 1420 a través de la red 1450, de una red de área de almacenamiento ("SAN"), de un bus serie de alta velocidad y/o mediante otra tecnología de comunicación adecuada. En algunas formas de realización, el dispositivo de codificación 1500, el dispositivo de decodificación 1600, la fuente de vídeo codificado 1420 y/o la fuente de vídeo no codificado 1415 pueden comprender uno o más dispositivos físicos o lógicos replicados y/o distribuidos.

En varias formas de realización, la red 1450 puede incluir Internet, una red de área local ("LAN"), una red de área amplia ("WAN"), una red de datos celular y/u otra red de datos. En muchas formas de realización, puede haber más dispositivos de codificación 1500, dispositivos de decodificación 1600, fuentes de vídeo codificado 1420 y/o fuentes de vídeo no codificado 1415 de las que se ilustran.

La Figura 15 ilustra varios componentes de un dispositivo de codificación 1500 a modo de ejemplo, de conformidad con una forma de realización. La Figura 15 ilustra varios componentes de un dispositivo de codificación 1500 a modo de ejemplo. En algunas formas de realización, el dispositivo de codificación 1500 puede incluir muchos más componentes que los ilustrados en la Figura 15. Sin embargo, no es necesario que todos estos componentes, por lo general convencionales, se muestren para revelar una forma de realización ilustrativa. Tal como se muestra en la Figura 15, el dispositivo de codificación 1500 incluye una interfaz de red opcional 1530 para conectarse, de manera opcional, a la red 1450.

El dispositivo de codificación 1500 también incluye al menos una unidad de procesamiento 1510 (por ejemplo, una unidad de procesamiento central de uso general, una unidad de procesamiento integrada, una unidad de procesamiento de vídeo de uso especial, o similar), una memoria 1550, un dispositivo de salida opcional y/o una pantalla 1540, y un dispositivo de entrada opcional 1570 (por ejemplo, un teclado, un ratón, un control remoto, un joystick y similares), todos interconectados junto con la interfaz de red opcional 1530 a través de un bus 1520. La memoria 1550 por lo general comprende una memoria de acceso aleatorio ("RAM"), una memoria de solo lectura ("ROM") y un dispositivo de almacenamiento masivo permanente, tal como una unidad de disco. La memoria 1550 almacena un codificador de vídeo 800 con filtrado de compensación de movimiento de libro de códigos (véase la Figura 8, descrita con anterioridad); uno o ambos de un libro de códigos de conjunto de filtros de subpíxeles de compensación de movimiento 1565 y un libro de códigos de filtro de posición entera de compensación de movimiento 1570; una rutina de generación de libro de códigos "fuera de línea" 1000 opcional (véase Figura 10, descrita con anterioridad); una rutina de búsqueda de libro de códigos de filtro de compensación de movimiento 1100 (véase Figura 11, descrita con anterioridad); y una rutina de codificación de vídeo de filtrado de compensación de movimiento de libro de códigos (véase Figura 17, que se describe a continuación). Además, la memoria 1550 también almacena un sistema operativo 1555. En algunas formas de realización, estos componentes de software pueden cargarse desde un medio de almacenamiento legible por ordenador 1595 en la memoria 1550 del dispositivo de codificación 1500 utilizando un mecanismo de accionamiento (no ilustrado) asociado con un medio de almacenamiento transitorio legible por ordenador 1595, tal como un disquete, una cinta, una unidad de DVD/CD-ROM, una tarjeta de memoria o similar. En algunas formas de realización, los componentes de software también pueden cargarse a través de la interfaz de red opcional 1530, en lugar de a través de un medio de almacenamiento legible por ordenador 1595.

En algunas formas de realización, el dispositivo de codificación 1500 puede comprender, además, una interfaz especializada 1535 para comunicarse con la fuente de vídeo no codificado 1415, tal como un bus serie de alta velocidad, o similar. En otras formas de realización, el dispositivo de codificación 1500 puede comunicarse con la fuente de vídeo no codificado 1415 a través de la interfaz de red 1530. En otras formas de realización, la fuente de vídeo no codificado 1415 puede residir en la memoria 1550.

Aunque se ha descrito un dispositivo de codificación 1500, a modo de ejemplo, que por lo general se ajusta a los dispositivos informáticos de uso general convencionales, un dispositivo de codificación 1500 puede ser cualquiera de entre un gran número de dispositivos capaces de codificar vídeo, por ejemplo, un dispositivo de grabación de vídeo,

un dispositivo de coprocesador y/o acelerador, un ordenador personal, una consola de juegos, un decodificador, un ordenador portátil, un teléfono móvil o cualquier otro dispositivo adecuado.

La Figura 16 ilustra varios componentes de un dispositivo de decodificación 1600, a modo de ejemplo, de conformidad con una forma de realización. En algunas formas de realización, el dispositivo de decodificación 1600 puede incluir muchos más componentes que los ilustrados en la Figura 16. Sin embargo, no es necesario que todos estos componentes, por lo general convencionales, se muestren con el fin de revelar una forma de realización ilustrativa. Tal como se muestra en la Figura 16, el dispositivo de decodificación 1600 incluye una interfaz de red opcional 1630 para conectarse, de manera opcional, a la red 1450.

El dispositivo de decodificación 1600 también incluye al menos una unidad de procesamiento 1610 (por ejemplo, una unidad de procesamiento central de uso general, una unidad de procesamiento integrada, una unidad de procesamiento de vídeo de uso especial, o similar), una memoria 1650, un dispositivo de salida opcional y/o una pantalla 1640, y un dispositivo de entrada opcional 1670 (por ejemplo, un teclado, un ratón, un control remoto, un joystick y similares), todos interconectados junto con la interfaz de red opcional 1630 a través de un bus 1620. La memoria 1650 por lo general comprende una memoria de acceso aleatorio ("RAM"), una memoria de solo lectura ("ROM") y un dispositivo de almacenamiento masivo permanente, tal como una unidad de disco. La memoria 1650 almacena códigos de programa para un decodificador de vídeo 900 con filtrado de compensación de movimiento de libro de códigos (véase Figura 9, descrita con anterioridad); uno o ambos de entre un libro de códigos de conjunto de filtros de subpíxeles de compensación de movimiento 1665 y un libro de códigos de filtro de posición entera de compensación de movimiento 1670; y una rutina de decodificación de vídeo de filtrado de compensación de movimiento de libro de códigos (véase Figura 21, que se describe a continuación). Además, la memoria 1650 también almacena un sistema operativo 1655. En algunas formas de realización, estos componentes de software pueden cargarse desde un medio de almacenamiento legible por ordenador 1695 en la memoria 1650 del dispositivo de decodificación 1600 utilizando un mecanismo de accionamiento (no ilustrado) asociado con un medio de almacenamiento no transitorio legible por ordenador 1695, tal como un disquete, una cinta, una unidad de DVD/CD-ROM, una tarjeta de memoria o similar. En algunas formas de realización, los componentes de software también se pueden cargar a través de la interfaz de red opcional 1630, en lugar de a través de un medio de almacenamiento 1695 legible por ordenador.

En algunas formas de realización, el dispositivo de decodificación 1600 puede comprender, además, una interfaz especializada 1635 para comunicarse con la fuente de vídeo codificado 1420, tal como un bus en serie de alta velocidad o similar. En algunas formas de realización, el dispositivo de decodificación 1600 puede comunicarse con la fuente de vídeo codificado 1420 a través de la interfaz de red 1630. En otras formas de realización, la fuente de vídeo codificado 1415 puede residir en la memoria 1650.

Aunque se ha descrito un dispositivo de decodificación 1600, a modo de ejemplo, que por lo general se ajusta a los dispositivos informáticos de uso general convencionales, un dispositivo de decodificación 1600 puede ser cualquiera de entre un gran número de dispositivos capaces de decodificar vídeo, por ejemplo, un dispositivo de reproducción de vídeo, un ordenador personal, una consola de juegos, un decodificador, un ordenador portátil, un teléfono móvil o cualquier otro dispositivo adecuado.

La Figura 17 ilustra una rutina 1700 de codificación de vídeo de filtrado de compensación de movimiento de libro de códigos, tal como se puede realizar mediante el dispositivo de codificación 1500 de conformidad con una forma de realización. En el bloque de subrutina 1800 (véase Figura 18, descrito a continuación), la rutina 1700 obtiene un libro de códigos que incluye una multiplicidad de filtros de compensación de movimiento agrupados en una pluralidad de grupos de posiciones de subpíxeles que corresponden, respectivamente, a una pluralidad de posiciones de subpíxeles. Comprendiendo, cada uno de entre la pluralidad de grupos de posiciones de subpíxeles, una pluralidad de filtros de compensación de movimiento adecuados para interpolar bloques de una imagen en una correspondiente de entre la pluralidad de posiciones de subpíxeles. Por ejemplo, tal como se describió con anterioridad en referencia a la Tabla 1, un libro de códigos puede incluir una multiplicidad de filtros de compensación de movimiento (por ejemplo, $a_0..15$, $b_0..15... 0_0..15$) agrupados en grupos de posiciones de subpíxeles, tales como las siguientes:

- grupo de posiciones de subpíxeles $\{a\}$: $\{a_0, a_1, a_2..15\}$;
- grupo de posiciones de subpíxeles $\{b\}$: $\{b_0, b_1, b_2..15\}$;
- grupo de posiciones de subpíxeles $\{o\}$: $\{o_0, o_1, o_2..15\}$.

En el bloque 1705, la rutina 1700 obtiene una trama o imagen de vídeo sin codificar para codificar. En algunos casos, se puede determinar que la trama o la imagen de vídeo no codificados incluyen varias partes diferentes (véanse, por ejemplo, las Figuras 7a-f, que muestran varios modos de partición para dividir una imagen en partes). En otros casos, toda la trama sin codificar o la imagen de vídeo puede tratarse como una sola parte.

Durante la codificación de la trama o imagen no codificadas, la rutina 1700 procesa cada una de las una o más partes de la trama o imagen no codificadas comenzando en el bloque de bucle de inicio 1710. En el bloque de subrutina 1900

(véase Figura 19, descrita a continuación), la rutina 1700 selecciona un subconjunto de entre la multiplicidad de filtros de compensación de movimiento del libro de códigos por estar bien adaptado para la parte actual de la trama de vídeo o imagen, incluyendo el subconjunto seleccionado uno de cada uno de entre la pluralidad de grupos de posiciones de subpíxeles. Por ejemplo, tal como se describió con anterioridad con referencia a la Tabla 1, un libro de códigos puede caracterizarse, además, por incluir un número (por ejemplo, 256) de subconjuntos predeterminados de filtros de compensación de movimiento, uno de cada grupo de posiciones de subpíxeles, por ejemplo:

- id subconjunto=0: {a₀, b₀, c₀, d₀, e₀, f₀, g₀, h₀, i₀, j₀, k₀, l₀, m₀, n₀, o₀};
- id subconjunto=1: {a₁, b₁, c₁, d₁, e₁, f₁, g₁, h₁, i₁, j₁, k₁, l₁, m₁, n₁, o₁};
- id subconjunto=15: {a₁₅, b₁₅, c₁₅, d₁₅, e₁₅, f₁₅, g₁₅, h₁₅, i₁₅, j₁₅, k₁₅, l₁₅, m₁₅, n₁₅, o₁₅};
- id subconjunto=16: {a₀, b₅, c₇, d₉, e₄, f₄, g₁₂, h₆, i₇, j₂, k₄, l₁₁, m₈, n₀, o₁₀};
- id subconjunto=255: {a₁₀, b₆, c₁₂, d₉, e₂, f₂, g₃, h₁₀, i₃, j₅, k₁, l₈, m₁₁, n₆, o₁₀}.

El bloque de subrutina 1900 devuelve un indicador de conjunto de filtros enviado y (si el indicador de conjunto de filtros enviado=1) un código de conjunto de filtros de subpíxeles que identifica el subconjunto seleccionado de entre la pluralidad de filtros de compensación de movimiento dentro del libro de códigos. Por ejemplo, en una forma de realización, el código del conjunto de filtros de subpíxeles puede ser (o derivarse de) un número de "id subconjunto" u otro índice en el libro de códigos. En el bloque 1720, la rutina 1700 escribe el indicador enviado del conjunto de filtros devuelto y el código del conjunto de filtros de subpíxeles (si lo indica el indicador enviado del conjunto de filtros) en un flujo binario codificado generado por la rutina 1700.

En el bloque de subrutina 2000, la rutina 1700 determina, de manera opcional, un filtro de compensación de movimiento de posición entera (véase Figura 20 descrita a continuación) para la parte actual de la trama o imagen no codificada. Al finalizar el bloque de bucle 1730, la rutina 1700 itera de nuevo al bloque 1710 para procesar la siguiente parte de la trama o imagen no codificada (si existe). Habiendo codificado así al menos un conjunto de filtro de compensación de movimiento establecido en el flujo binario, la rutina 1700 finaliza en el bloque 1799. En muchos casos, la rutina 1700 se puede realizar en serie para una serie de tramas de un vídeo, siendo la rutina 1700 parte de un proceso de codificación de vídeo más grande.

La Figura 18 ilustra una subrutina 1800 para obtener un libro de códigos, de conformidad con una forma de realización. En algunas formas de realización, el proceso de generar un libro de códigos puede ser bastante largo (tarda horas o incluso días), por lo que el libro de códigos puede generarse solamente una vez (o al menos con poca frecuencia) y almacenarse para su reutilización. En el bloque de decisión 1805, la subrutina 1800 determina si un libro de códigos generado previamente se ha almacenado para su reutilización. Si es así, en el bloque 1810, la subrutina 1800 efectúa la lectura del libro de códigos almacenado y lo devuelve al final del bloque 1899.

Sin embargo, si un libro de códigos generado previamente no se ha almacenado para su reutilización, entonces la subrutina 1800 comienza el proceso de generar un libro de códigos. En algunas formas de realización, los bloques 1815-1865 se pueden realizar de forma independiente, tal como una rutina de generación de libro de códigos "fuera de línea". Los bloques 1815-1865 pueden entenderse como una vista alternativa o una caracterización alternativa similar a los procesos ilustrados en los bloques 1005-1050 de la Figura 10, que se describe en detalle con anterioridad. En aras de la brevedad, el tema descrito con anterioridad en relación con la Figura 10 por lo general no se reiterará en la descripción siguiente de la Figura 18.

Comenzando en el bloque de bucle inicial 1815, la subrutina 1800 procesa cada una de entre una serie de posiciones de subpíxeles (por ejemplo, a-o, tal como se ilustra en la Figura 3, descrita con anterioridad).

En el bloque 1820, la subrutina 1800 determina un filtro de compensación de movimiento predeterminado para la posición de subpíxeles actual. Por ejemplo, en una forma de realización, la subrutina 1800 puede utilizar un filtro de compensación de movimiento de subpíxeles fijo, tal como el del conjunto de filtros fijos H.264 o cualquier otro conjunto de filtros predeterminado adecuado.

En el bloque 1825, la subrutina 1800 refina iterativamente el filtro de compensación de movimiento predeterminado para generar una multiplicidad de filtros de compensación de movimiento refinados para la posición de subpíxeles actual. Por ejemplo, en una forma de realización, la subrutina 1800 puede emplear un algoritmo de recorrido aleatorio (o un algoritmo de optimización similar) para refinar iterativamente el filtro de compensación de movimiento predeterminado para generar una multiplicidad de filtros de compensación de movimiento refinados.

Comenzando en el bloque de bucle inicial 1830, la subrutina 1800 procesa cada uno de entre la multiplicidad refinada de filtros de compensación de movimiento refinados. En el bloque 1835, la subrutina 1800 determina una ganancia de codificación (por ejemplo, una medida de SAD) para el filtro de compensación de movimiento refinado actual. En el

bloque 1840, la subrutina 1800 itera de retorno al bloque 1830 para procesar el siguiente filtro de compensación de movimiento refinado (si lo hubiere).

En el bloque 1845, la subrutina 1800 utiliza las ganancias de codificación determinadas para seleccionar un grupo de filtros de compensación de movimiento refinados de buen rendimiento para la posición actual de subpíxeles (por ejemplo, un grupo de filtros con ganancia de codificación alta/SAD baja). En el bloque 1850, la subrutina 1800 almacena el grupo seleccionado de filtros de compensación de movimiento refinados de buen rendimiento para la posición de subpíxeles actual. Al finalizar el bloque de bucle 1855, la subrutina 1800 itera de nuevo al bloque 1815 para procesar la siguiente posición de subpíxeles (si la hubiere). En algunas formas de realización, el grupo seleccionado de filtros de compensación de movimiento refinados de buen rendimiento se puede considerar como filtros "base" (tal como se describió con anterioridad) para la posición de subpíxeles actual, que junto con los filtros de "base" para otras posiciones de subpíxeles, forman un libro de códigos "base" o grupo "base" de conjuntos de filtros.

En el bloque 1860, la subrutina 1800 utiliza los grupos de filtros de compensación de movimiento refinados de buen rendimiento (un grupo para cada posición de subpíxeles) para determinar iterativamente una pluralidad de conjuntos de filtros, cada conjunto de filtros incluye una combinación distinta de filtros, un filtro para cada posición de subpíxeles. (En algunos contextos de esta descripción, la expresión "conjuntos de filtros" también se denomina como "subconjuntos" de filtros de compensación de movimiento, por ejemplo, $\{a_0, b_5, c_7, d_9, e_4, f_4, g_{12}, h_6, i_7, j_2, k_4, l_{11}, m_8, n_0, o_{10}\}$). En algunas formas de realización, la subrutina 1800 puede determinar iterativamente la pluralidad de conjuntos/subconjuntos de filtros utilizando un algoritmo de recorrido aleatorio (o un algoritmo de optimización similar). En el bloque 1865, la subrutina 1800 indica los conjuntos de filtros/subconjuntos determinados en el libro de códigos, de modo que cada uno de los conjuntos de filtros/subconjuntos determinados de filtros pueda direccionarse, seleccionarse, indexarse o identificarse de otro modo dentro del libro de códigos. Por ejemplo, en una forma de realización, la subrutina 1800 puede asignar a cada uno de los conjuntos de filtros/subconjuntos determinados un número "id" o índice.

En el bloque 1870, la subrutina 1800 almacena el libro de códigos así generado para su posterior reutilización. En el bloque 1899, la subrutina 1800 finaliza y devuelve el libro de códigos.

La Figura 19 ilustra una subrutina 1900 para seleccionar un subconjunto de filtros de compensación de movimiento desde el libro de códigos, de conformidad con una forma de realización. La Figura 19 puede entenderse como una vista alternativa o una caracterización alternativa similar a los procesos ilustrados en la Figura 11, que se describieron con anterioridad en detalle. En aras de la brevedad, la materia descrita con anterioridad en relación con la Figura 11 por lo general no se reiterará en la descripción siguiente de la Figura 19.

En el bloque 1903, la subrutina 1900 obtiene al menos una parte de una imagen (véanse, por ejemplo, las Figuras 7a-f, que muestran varios modos de partición para dividir una imagen en partes), la parte de una imagen que incluye varios bloques de contenido de la imagen. Comenzando en el bloque de bucle inicial 1905, la subrutina 1900 procesa cada bloque de contenido de imagen de la parte de imagen. Comenzando en el bloque de bucle inicial 1908, la subrutina 1900 procesa cada una de varias posiciones de subpíxeles (por ejemplo, a-o, tal como se ilustra en la Figura 3, descrita con anterioridad).

En el bloque 1910, la subrutina 1900 obtiene una multiplicidad de filtros de compensación de movimiento "base" para la posición actual de subpíxeles (por ejemplo, $a_0, a_1, a_2 \dots a_{15}$). Comenzando en el bloque de bucle inicial 1913, la subrutina 1900 procesa cada uno de los filtros de compensación de movimiento "base" para la posición actual de subpíxel. En el bloque 1915, la subrutina 1900 determina una ganancia de codificación individual (por ejemplo, calculando un SAD) para el filtro de compensación de movimiento "base" actual para la posición actual de subpíxeles (por ejemplo, a_0). Al finalizar el bloque de bucle 1918, la subrutina 1900 itera de nuevo al bloque 1913 para procesar el siguiente filtro de compensación de movimiento "base" (si lo hubiere) para la posición de subpíxeles actual.

Habiendo determinado las ganancias de codificación para cada uno de los filtros de compensación de movimiento "base" para la posición de subpíxeles actual, en el bloque 1920, la subrutina 1900 determinó una ganancia de codificación para un filtro de compensación de movimiento "predeterminado", que es parte de un filtro predeterminado de conjunto de filtros de compensación de movimiento. Por ejemplo, en una forma de realización, la subrutina 1900 puede determinar una ganancia de codificación para un filtro de compensación de movimiento fijo para la posición actual de subpíxeles, tal como uno de los filtros del conjunto de filtro de compensación de movimiento fijo utilizado en H.264 u otro conjunto adecuado de filtro de compensación de movimiento "predeterminado".

Al finalizar el bloque de bucle 1923, la subrutina 1900 itera de nuevo al bloque 1908 para procesar la siguiente posición de subpíxeles (si la hubiere). Al finalizar el bloque de bucle 1925, la subrutina 1900 itera de nuevo al bloque 1905 para procesar el siguiente bloque de contenido de imagen de la parte de imagen (si la hubiere).

En el bloque 1928, la subrutina 1900 obtiene una pluralidad de conjuntos de filtros adaptativos, comprendiendo cada conjunto de filtros uno de los filtros de compensación de movimiento "base" para cada posición de subpíxel. Por ejemplo, en una forma de realización, la subrutina 1900 puede obtener un libro de códigos que incluye una pluralidad de conjuntos de filtros adaptativos tales como los siguientes:

- id conjunto=0: {a₀, b₀, c₀, d₀, e₀, f₀, g₀, h₀, i₀, j₀, k₀, l₀, m₀, n₀, o₀};
- id conjunto=1: {a₁, b₁, c₁, d₁, e₁, f₁, g₁, h₁, i₁, j₁, k₁, l₁, m₁, n₁, o₁};
- id conjunto=15: {a₁₅, b₁₅, c₁₅, d₁₅, e₁₅, f₁₅, g₁₅, h₁₅, i₁₅, j₁₅, k₁₅, l₁₅, m₁₅, n₁₅, o₁₅};
- id conjunto=16: {a₀, b₅, c₇, d₉, e₄, f₄, g₁₂, h₆, i₇, j₂, k₄, l₁₁, m₈, n₀, o₁₀};
- id conjunto=255: {a₁₀, b₆, c₁₂, d₉, e₂, f₂, g₃, h₁₀, i₃, j₅, k₁, l₈, m₁₁, n₆, o₁₀}.

Comenzando en el bloque de bucle inicial 1930, la subrutina 1900 procesa cada uno de los conjuntos de filtros adaptativos obtenidos. En el bloque 1933, la subrutina 1900 determina una ganancia de codificación agregada para el conjunto de filtros adaptativos actual. Por ejemplo, en una forma de realización, la determinación de una ganancia de codificación agregada puede incluir la suma de las ganancias de codificación individuales (tal como se determina en el bloque 1915) para los filtros constituyentes del conjunto de filtros adaptativos actual. Al finalizar el bloque de bucle 1935, la subrutina 1900 itera de retorno al bloque 1930 para procesar el siguiente conjunto de filtros adaptativos (si lo hubiere).

Habiendo determinado las ganancias de codificación agregadas para cada uno de los conjuntos de filtros adaptativos, en el bloque 1938, la subrutina 1900 determina una ganancia de codificación agregada "predeterminada" para el conjunto de filtros "predeterminado" (por ejemplo, el conjunto de filtros fijos de H .264 u otro conjunto adecuado), utilizando las ganancias de codificación "predeterminadas" individuales determinadas en el bloque 1920, descrito con anterioridad.

En el bloque 1940, utilizando las ganancias de codificación agregadas determinadas en el bloque 1933, la subrutina 1900 selecciona un conjunto de filtros adaptativos con una ganancia de codificación agregada favorable para la parte de una imagen obtenida en el bloque 1903. En algunas formas de realización, el conjunto de filtros adaptativos seleccionado puede ser el conjunto de filtros adaptativos con la más alta ganancia de codificación agregada determinada.

En el bloque de decisión 1943, la subrutina 1900 compara la ganancia de codificación agregada "predeterminada" determinada en el bloque 1938 con la ganancia de codificación agregada del conjunto de filtros adaptativos favorables seleccionado. Si el conjunto de filtros "predeterminado" proporciona una ganancia de codificación más alta (menor SAD), entonces en el bloque 1950, la subrutina 1900 establece un indicador de filteret_sent en '0' o 'NO' y devuelve el indicador de filteret_sent al finalizar el bloque 1999.

Sin embargo, si el conjunto de filtros adaptativos proporciona una ganancia de codificación más alta (menor SAD), entonces en el bloque 1945, la subrutina 1900 establece un indicador de envío del conjunto de filtros en '1' o 'SÍ', y en el bloque 1948, la subrutina 1900 determina un código del conjunto de filtro de subpíxeles que identifica el conjunto de filtros adaptativos seleccionado. Por ejemplo, en una forma de realización, el código del conjunto de filtros de subpíxeles puede ser (o derivarse de) un número de "ID de subconjunto" u otro índice en un libro de códigos. La subrutina 1900 retorna el indicador de conjunto de filtros enviado y el código de conjunto de filtros de subpíxeles determinado al finalizar el bloque 1999.

La Figura 20 ilustra una subrutina 2000 para determinar un filtro de compensación de movimiento de posición entera para una parte de una imagen, de conformidad con una forma de realización. En el bloque 2005, la subrutina 2000 obtiene al menos una parte de una imagen (véanse, por ejemplo, las Figuras 7a-f, que muestran varios modos de partición para dividir una imagen en partes), la parte de una imagen incluye varios bloques de contenido de imagen. Comenzando en el bloque de bucle inicial 2010, la subrutina 2000 procesa cada bloque de contenido de imagen de la parte de imagen.

En el bloque 2015, la subrutina 2000 obtiene una pluralidad de filtros de compensación de movimiento de posición entera, por ejemplo, 0₀, 0₁, 0₂...0₄₇. Por ejemplo, en una forma de realización, la subrutina 2000 puede obtener un libro de códigos de posición entera, tal como el que se muestra parcialmente en la Tabla 2 anterior. En otras formas de realización, la pluralidad de filtros de compensación de movimiento de posición entera puede incluirse dentro de un libro de códigos que también incluye filtros de posiciones de subpíxeles.

Comenzando en el bloque de bucle de inicio 2020, la subrutina 2000 procesa cada uno de entre la pluralidad de filtros de compensación de movimiento de posición entera. En el bloque 2025, la subrutina 2000 determina una ganancia de codificación individual (por ejemplo, calculando un SAD) para el filtro de compensación de movimiento de posición entera actual. Al finalizar el bloque de bucle 2030, la subrutina 2000 itera de nuevo al bloque 2020 para procesar el siguiente filtro de compensación de movimiento de posición entera (si lo hubiere). Al finalizar el bloque de bucle 2035, la subrutina 2000 itera de nuevo al bloque 2010 para procesar el siguiente bloque de contenido de imagen (si lo hubiere) de la parte de imagen.

Una vez que se han determinado las ganancias de codificación para cada filtro de compensación de movimiento de posición entera y cada bloque de contenido de imagen de la parte de imagen, en el bloque 2040, la subrutina 2000 selecciona uno de entre la pluralidad de filtros de compensación de movimiento de posición entera como que tiene una ganancia de codificación favorable (SAD baja) para la parte de imagen actual.

En el bloque de decisión 2045, la subrutina 2000 compara la ganancia de codificación del filtro de compensación de movimiento de posición entera favorable seleccionado con una ganancia de codificación de un conjunto de filtros de subpíxeles también seleccionados para la misma parte de imagen (véase, por ejemplo, la descripción del bloque 2040 en la Figura 20 anterior). Si el conjunto de filtros de subpíxeles proporciona una ganancia de codificación más alta (SAD más bajo), entonces, en el bloque 2060, la subrutina 2000 establece un indicador `intfilt_sent` en el flujo binario a '0' o 'NO' y retorna al finalizar al bloque 2099.

Sin embargo, si el filtro de compensación de movimiento de posición entera favorable proporciona una ganancia de codificación más alta (SAD más bajo), entonces en el bloque 2050, la subrutina 2000 establece un indicador `intfilt_sent` en '1' o 'SÍ', y en el bloque 2055, la subrutina 2000 escribe en el flujo binario un código de filtro de posición entera que identifica el filtro de compensación de movimiento de posición entera favorable seleccionado. Por ejemplo, en una forma de realización, el código de filtro de posición entera puede ser (o derivarse de) un número de "id" u otro índice en un libro de códigos. La subrutina 2000 retorna al finalizar al bloque 2099.

La Figura 21 ilustra una rutina 2100 de decodificación de vídeo de filtrado de compensación de movimiento de libro de códigos, tal como puede realizarse mediante el dispositivo de decodificación 1600 de conformidad con una forma de realización. En el bloque 2105, la rutina 2100 obtiene un flujo binario codificado de una fuente de vídeo codificada, incluyendo el flujo binario codificado una o más imágenes o tramas de vídeo. Comenzando en el bloque de bucle inicial 2110, la rutina 2100 procesa cada una de las imágenes o tramas de vídeo codificadas.

En el bloque 2115, la rutina 2100 efectúa la lectura del flujo binario de uno o más códigos de parte de imagen que identifican una o más partes de imagen (véanse, por ejemplo, las Figuras 7a-f, que muestran varios modos de partición para dividir una imagen en partes) de la imagen o trama codificada actual. Por ejemplo, en una forma de realización, la rutina 2100 puede efectuar la lectura de algunos o todos los códigos tales como `pict_only` 1202, `numpt` 1203, `map_sent` 1204 y/o `ptmap` 1205, tal como se muestra en la Figura 12 y se describe con anterioridad.

Comenzando en el bloque de bucle inicial 2120, la rutina 2100 procesa cada una de las una o más partes de imagen de la imagen codificada actual o trama de vídeo.

En el bloque de decisión 2125, la rutina 2100 determina si el flujo binario codificado incluye un código de filtro de posición entera para la parte de imagen actual de la imagen o trama de vídeo codificada actual. Por ejemplo, en una forma de realización, la rutina 2100 puede determinar si el flujo binario codificado incluye un indicador `intfilt_sent` 1210, tal como se muestra en la Figura 12 y se describe con anterioridad.

Si la rutina 2100 determina que el flujo binario codificado incluye dicho código de filtro de posición entera, entonces en el bloque 2130, la rutina 2100 efectúa la lectura de un código de filtro de posición entera del flujo binario. Por ejemplo, en una forma de realización, la rutina 2100 puede efectuar la lectura de un código `intpelfilt_code` 1211 del flujo binario, tal como se muestra en la Figura 12 y se describe con anterioridad.

En el bloque 2135, la rutina 2100 obtiene una pluralidad de filtros de compensación de movimiento de posición entera, por ejemplo, $0_0, 0_1, 0_2 \dots 0_{47}$. Por ejemplo, en una forma de realización, la subrutina 2000 puede obtener un libro de códigos de posición entera, tal como el que se muestra parcialmente en la Tabla 2 anterior. En otras formas de realización, la pluralidad de filtros de compensación de movimiento de posición entera puede incluirse dentro de un libro de códigos que también incluye filtros de posiciones de subpíxeles. En condiciones normales, la rutina 2100 efectuaría la lectura de una copia almacenada previamente de entre la pluralidad de filtros de compensación de movimiento de posición entera desde una memoria accesible a la rutina 2100.

En el bloque 2140, la rutina 2100 selecciona uno de entre la pluralidad de filtros de compensación de movimiento de posición entera tal como lo indica el código de filtro de posición entera (p. ej., código de filtro interno 1211) objeto de lectura del flujo binario en el bloque 2130. La subrutina 2100 utiliza entonces el filtro de compensación de movimiento de posición entera seleccionado en el bloque 2170 para predecir bloques de la parte de imagen actual de la imagen o trama de vídeo codificada actual.

Sin embargo, si en el bloque de decisión 2125, la rutina 2100 determina que el flujo binario codificado no incluye un código de filtro de posición entera, entonces, en el bloque de decisión 2145, la rutina 2100 determina si el flujo binario codificado incluye un código de conjunto de filtros de subpíxeles para la parte de imagen actual de la imagen codificada actual o trama de vídeo. Por ejemplo, en una forma de realización, la rutina 2100 puede determinar si el flujo binario codificado incluye un indicador `1206` de `filt_sent`, tal como se muestra en la Figura 12 y se describió con anterioridad. Si no es así, en el bloque 2165, la rutina 2100 seleccionó un conjunto de filtros de compensación de movimiento de subpíxeles predeterminado (por ejemplo, el conjunto de filtros de subpíxeles fijos utilizados en H.264, u otro conjunto

adecuado de filtros predeterminados). La subrutina 2100 utiliza entonces el filtro de compensación de movimiento de subpíxeles predeterminado establecido en el bloque 2170 para predecir bloques de la parte de imagen actual de la imagen o trama de vídeo codificada actual.

- 5 Sin embargo, si la rutina 2100 determina en el bloque de decisión 2145 que el flujo binario codificado incluye un código de conjunto de filtros de subpíxeles para la parte de imagen actual de la imagen codificada actual o trama de vídeo, entonces en el bloque 2150, la rutina 2100 efectúa la lectura de un código del conjunto de filtros de subpíxeles desde el flujo binario. Por ejemplo, en una forma de realización, la rutina 2100 puede efectuar la lectura de un código subpelfilt_code 1208 desde el flujo binario, tal como se muestra en la Figura 12 y se describió con anterioridad.
- 10 En el bloque 2155, la rutina 2100 obtiene un libro de códigos de compensación de movimiento de subpíxeles, tal como el que se muestra parcialmente en la Tabla 1, descrita con anterioridad. En condiciones normales, la rutina 2100 efectuaría la lectura de una copia almacenada previamente del libro de códigos de compensación de movimiento de subpíxeles desde una memoria accesible para la rutina 2100.
- 15 En el bloque 2140, la rutina 2100 selecciona uno de entre la pluralidad de conjuntos de filtros de compensación de movimiento de subpíxeles tal como se indica por el código de conjunto de filtros de subpíxeles (por ejemplo, código subpelfilt 1208) objeto de lectura desde el flujo binario en el bloque 2150. La subrutina 2100 utiliza entonces el conjunto de filtro de compensación de movimiento de subpíxeles seleccionado en el bloque 2170 para predecir bloques de la parte de imagen actual de la imagen o trama de vídeo codificada actual.
- 20 Al finalizar el bloque de bucle 2175, la rutina 2100 itera de nuevo al bloque 2120 para procesar la siguiente parte de imagen (si la hubiere) de la imagen o trama de vídeo codificada actual.
- 25 En el bloque 2180, la rutina 2100 genera una imagen decodificada de conformidad con los bloques predichos de una o más partes de imagen predichas en varias iteraciones del bloque 2170. En el bloque 2185, la rutina 2100 representa la imagen generada en una pantalla asociada con el dispositivo que realiza la rutina 2100.
- 30 Al finalizar el bloque de bucle 2190, la rutina 2100 itera de nuevo al bloque 2110 para procesar la siguiente imagen codificada o trama de vídeo (si lo hubiere). La rutina 2100 finaliza en el bloque 2199.

REIVINDICACIONES

1. Un método puesto en práctica por un dispositivo codificador de vídeo para codificar un conjunto de filtros de compensación de movimiento adaptativo para una pluralidad de posiciones de subpíxeles para predecir bloques en una trama de vídeo codificada, comprendiendo el método:

obtener, mediante el dispositivo codificador de vídeo, un libro de códigos que comprende una multiplicidad de filtros de compensación de movimiento agrupados en una pluralidad de grupos de posiciones de subpíxeles que corresponden respectivamente a la pluralidad de posiciones de subpíxeles, comprendiendo cada uno de entre la pluralidad de grupos de posiciones de subpíxeles una pluralidad de filtros de compensación de movimiento adecuados para interpolar bloques de una imagen en una posición correspondiente de entre la pluralidad de posiciones de subpíxeles;

obtener, mediante el dispositivo codificador de vídeo, una trama de vídeo no codificada para codificar mediante el dispositivo codificador de vídeo;

codificar, mediante el dispositivo codificador de vídeo, al menos una parte de la trama de vídeo en un flujo binario, comprendiendo la parte de la trama de vídeo una pluralidad de bloques de contenido de imagen; y

durante la codificación de al menos la parte de la trama de vídeo, el dispositivo codificador de vídeo:

selecciona desde el libro de códigos un subconjunto de entre la pluralidad de filtros de compensación de movimiento como estando bien adaptados para predecir la pluralidad de bloques de contenido de imagen, comprendiendo el subconjunto seleccionado uno de cada uno de entre la pluralidad de grupos de posiciones de subpíxeles;

incluyendo en el flujo binario un código de filtro de subpíxeles que identifica el subconjunto seleccionado de entre la pluralidad de filtros de compensación de movimiento dentro del libro de códigos;

caracterizado porque el método comprende, además:

dividir el libro de códigos en un libro de códigos base pequeño y un libro de códigos virtual más grande, comprendiendo el libro de códigos virtual de conjuntos de filtros, combinaciones de filtros seleccionados desde el libro de códigos base de conjuntos de filtros;

almacenar solamente el libro de códigos base y un algoritmo para sintetizar el libro de códigos virtual.

2. El método según la reivindicación 1, en donde seleccionar el subconjunto de entre la pluralidad de filtros de compensación de movimiento comprende seleccionar uno de entre una pluralidad de subconjuntos predeterminados de entre la pluralidad de filtros de compensación de movimiento.

3. El método según la reivindicación 2, en donde el código incluyendo en el flujo binario comprende un índice que identifica el subconjunto seleccionado de entre la pluralidad de subconjuntos predeterminados.

4. El método según la reivindicación 2, en donde seleccionar uno de entre la pluralidad de subconjuntos predeterminados de entre la pluralidad de filtros de compensación de movimiento comprende:

para cada uno de entre la pluralidad de bloques de contenido de imagen, determinar, respectivamente, una multiplicidad de ganancias de codificación individuales para la multiplicidad de filtros de compensación de movimiento; utilizar las multiplicidades determinadas de ganancias de codificación individuales, determinar una ganancia de codificación agregada para cada uno de entre la pluralidad de subconjuntos predeterminados de entre la pluralidad de filtros de compensación de movimiento aplicados a la pluralidad de bloques de contenido de imagen; y

seleccionar uno de entre la pluralidad de subconjuntos predeterminados como teniendo una ganancia de codificación agregada favorable para la pluralidad de bloques de contenido de imagen, en particular que comprende, además:

determinar una ganancia de codificación agregada predeterminado para un conjunto de filtros predeterminado aplicado a la pluralidad de bloques de contenido de imagen, comprendiendo el conjunto de imágenes predeterminado una pluralidad de filtros de compensación de movimiento de subpíxeles predeterminado correspondientes, respectivamente, a la pluralidad de posiciones de subpíxeles; y determinar que la ganancia de codificación agregada favorable es más favorable que la ganancia de codificación agregada predeterminado.

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

obtener una pluralidad de filtros de compensación de movimiento de posición entera; y

durante la codificación de al menos la parte de la trama de vídeo, el dispositivo codificador de vídeo selecciona uno de entre la pluralidad de filtros de compensación de movimiento de posición entera como estando bien adaptado para predecir la pluralidad de bloques de contenido de imagen.

5 6. El método según la reivindicación 5, que, además, comprende:

determinar que el seleccionado de entre la pluralidad de filtros de compensación de movimiento de posición entera proporciona una mejor ganancia de codificación que el subconjunto seleccionado de entre la pluralidad de filtros de compensación de movimiento; y

10 incluir en el flujo binario generado por el dispositivo codificador de vídeo:

un código de filtro de enteros que identifica el seleccionado de entre la pluralidad de filtros de compensación de movimiento de posición entera; y

15 un indicador de filtro de enteros que indica que el código de filtro de enteros está incluyendo en el flujo binario para al menos la parte de la trama de vídeo.

20 7. El método según la reivindicación 5, en donde el libro de códigos comprende, además, la pluralidad de filtros de compensación de movimiento de posición entera o en donde obtener la pluralidad de filtros de compensación de movimiento de posición entera comprende obtener un libro de códigos de posición entera.

8. El método según la reivindicación 1, en donde obtener el libro de códigos comprende:

25 determinar una pluralidad de filtros de compensación de movimiento predeterminado que corresponden, respectivamente, a la pluralidad de posiciones de subpíxeles;

30 generar una pluralidad de grupos de posiciones de subpíxeles correspondientes, respectivamente, a la pluralidad de posiciones de subpíxeles realizando al menos las etapas (i), (ii) y (iii) para cada una de entre la pluralidad de posiciones de subpíxeles:

35 (i) refinar iterativamente el correspondiente de entre la pluralidad de filtros de compensación de movimiento predeterminados para generar una multiplicidad de filtros de compensación de movimiento refinados correspondientes a la posición de subpíxeles actual utilizando las ecuaciones de Wiener Hopf;

(ii) determinar una multiplicidad de ganancias de codificación correspondientes, respectivamente, a la multiplicidad de filtros de compensación de movimiento refinados;

40 (iii) de conformidad con la multiplicidad determinada de ganancias de codificación, seleccionar desde la multiplicidad generada de filtros de compensación de movimiento refinados, un grupo de posiciones de subpíxeles de filtros de compensación de movimiento correspondiente a la posición de subpíxeles actual, en particular, que comprende, además, seleccionar una pluralidad de subconjuntos de filtros de compensación de movimiento, comprendiendo cada subconjunto un filtro de compensación de movimiento seleccionado desde cada uno de entre la pluralidad de grupos de posiciones de subpíxeles.

45 9. Un método puesto en práctica por un dispositivo de decodificador de vídeo para decodificar un conjunto de filtros de compensación de movimiento adaptativo para una pluralidad de posiciones de subpíxeles para predecir bloques de una trama de vídeo codificada, comprendiendo el método:

50 obtener, mediante el dispositivo decodificador de vídeo, un libro de códigos que comprende una multiplicidad de filtros de compensación de movimiento agrupados en una pluralidad de grupos de posiciones de subpíxeles que corresponden, respectivamente, a la pluralidad de posiciones de subpíxeles, comprendiendo cada uno de entre la pluralidad de grupos de posiciones de subpíxeles una pluralidad de filtros de compensación de movimiento adecuados para interpolar bloques de una imagen en una posición correspondiente de entre la pluralidad de posiciones de subpíxeles;

55 obtener un flujo binario codificado que comprende la trama de vídeo codificada y un código de filtro de subpíxeles que identifica el conjunto de filtros de compensación de movimiento adaptativo dentro del libro de códigos, comprendiendo el conjunto de filtros de compensación de movimiento adaptativo una pluralidad de filtros de compensación de movimiento correspondientes, respectivamente, a la pluralidad de posiciones de subpíxeles;

60 durante la decodificación de la trama de vídeo codificada, el dispositivo decodificador de vídeo selecciona desde el libro de códigos el conjunto de filtros de compensación de movimiento adaptativo identificado por el código de filtro de subpíxeles;

65

utilizando el conjunto de filtros de compensación de movimiento adaptativo, predecir una pluralidad de bloques de contenido de imagen, correspondiendo la pluralidad de bloques a al menos una parte de la trama de vídeo codificada;

5 generar una imagen decodificada de conformidad con al menos la pluralidad prevista de bloques de contenido de imagen; y

efectuar la presentación de la imagen decodificada generada a una pantalla asociada con el dispositivo decodificador de vídeo;

10 caracterizado porque el método comprende, además:

dividir el libro de códigos en un libro de códigos base pequeño y en un libro de códigos virtual más grande, comprendiendo el libro de códigos virtual de conjuntos de filtros combinaciones de filtros seleccionados desde el libro de códigos base de conjuntos de filtros;

15 almacenar solamente el libro de códigos base y un algoritmo para sintetizar el libro de códigos virtual.

10. El método según la reivindicación 9, en donde el flujo binario codificado, incluye, además:

20 un segundo código de filtro de subpíxeles, que identifica un segundo filtro de compensación de movimiento adaptativo establecido dentro del libro de códigos; y

al menos un código de identificación de parte que permite al decodificador de vídeo identificar la parte de la trama de imagen codificada y una segunda parte de la trama de imagen codificada, que en particular comprende, además:

25 durante la decodificación de la trama de vídeo codificada, el dispositivo decodificador de vídeo selecciona desde el libro de códigos el segundo conjunto de filtros de compensación de movimiento adaptativo identificado por el segundo código de filtro de subpíxeles;

30 utilizando el segundo conjunto de filtros de compensación de movimiento adaptativo, predecir una segunda pluralidad de bloques de contenido de imagen, correspondiendo la segunda pluralidad de bloques a al menos una segunda parte de la trama de imagen codificada; y

35 en donde la imagen decodificada se genera, además, de conformidad con la segunda pluralidad prevista de bloques de contenido de imagen.

11. El método según la reivindicación 9, que comprende, además, obtener una pluralidad de filtros de compensación de movimiento de posición entera.

40 12. El método según la reivindicación 11, en donde el flujo binario codificado comprende, además:

al menos un código de identificación de parte que permite al decodificador de vídeo identificar la parte de la trama de imagen codificada y una segunda parte de la trama de imagen codificada; y

45 un código de filtro de enteros que identifica uno de entre la pluralidad de filtros de compensación de movimiento de posición entera; y

un indicador de filtro de enteros que indica que el código de filtro de enteros está incluyendo en el flujo binario para al menos la segunda parte de la trama de vídeo, en particular que comprende, además, durante la decodificación de la trama de vídeo codificada:

50 la lectura del indicador de filtro de enteros y la selección, en respuesta, de entre la pluralidad de filtros de compensación de movimiento de posición entera, el filtro de compensación de movimiento de posición entera identificado por el código de filtro de enteros;

55 utilizando el filtro de compensación de movimiento de posición entera seleccionado, predecir una segunda pluralidad de bloques de contenido de imagen, correspondiendo la segunda pluralidad de bloques a al menos la segunda parte de la trama de imagen codificada; y

60 en donde la imagen decodificada se genera, además, de conformidad con la segunda pluralidad prevista de bloques de contenido de imagen.

13. El método según la reivindicación 11, en donde el libro de códigos comprende, además, la pluralidad de filtros de compensación de movimiento de posición entera o en donde obtener la pluralidad de filtros de compensación de movimiento de posición entera comprende obtener un libro de códigos de posición entera.

65

14. Un medio de almacenamiento no transitorio legible por ordenador que comprende instrucciones que, cuando son ejecutadas por un ordenador, hacen que el ordenador lleve a cabo el método según la reivindicación 1 o 9.

- 5 15. Un aparato de codificación de vídeo que comprende un procesador y un medio de almacenamiento que almacena instrucciones que, cuando son ejecutadas por el procesador, hacen que el aparato de codificación de vídeo lleve a cabo el método según la reivindicación 1 o un aparato de decodificación de vídeo que comprende un procesador y un medio de almacenamiento que almacena instrucciones que, cuando son ejecutadas por el procesador, hacen que el aparato de decodificación de vídeo lleve a cabo el método según la reivindicación 9.

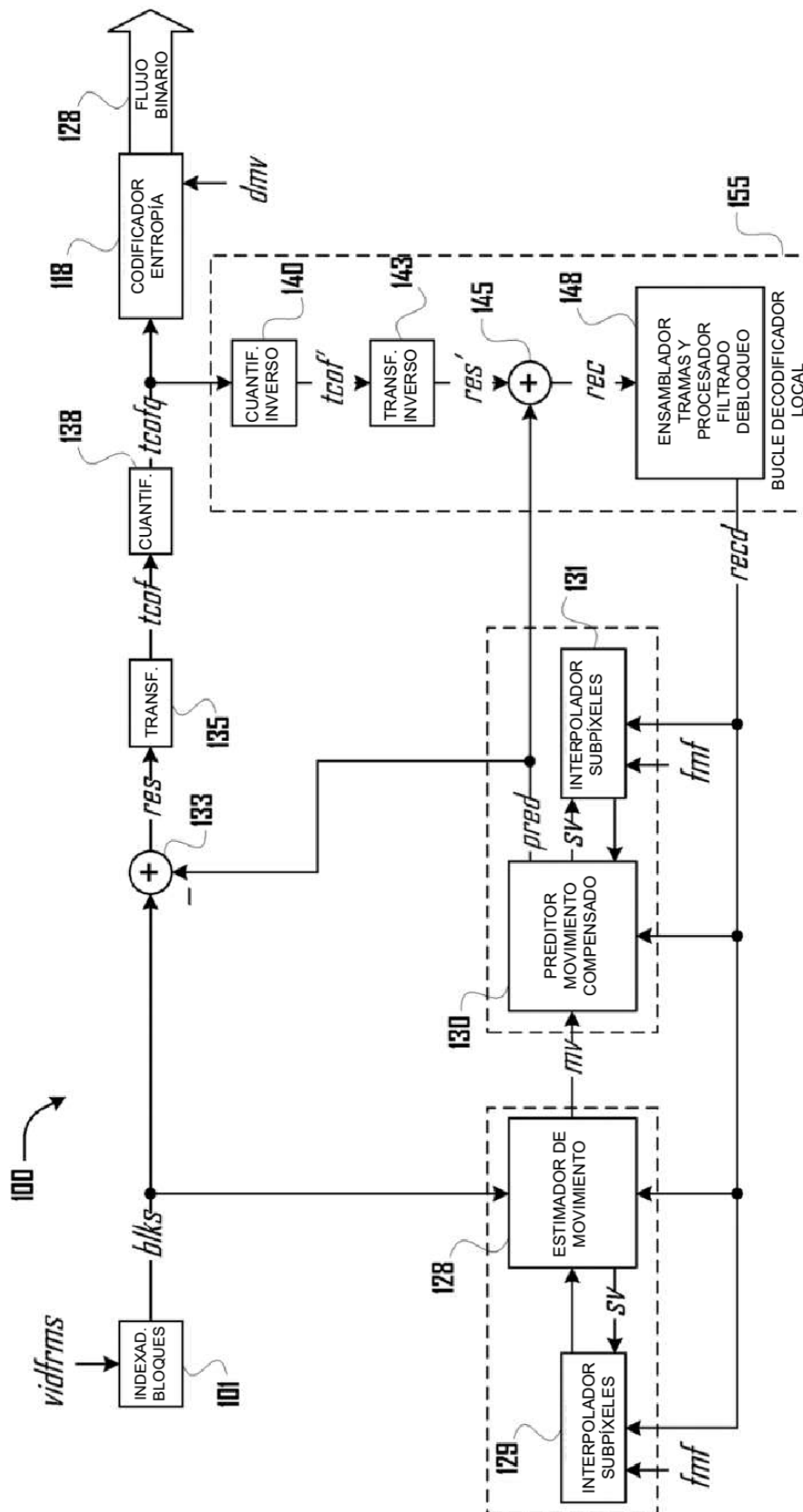


Fig.1 (técnica anterior)

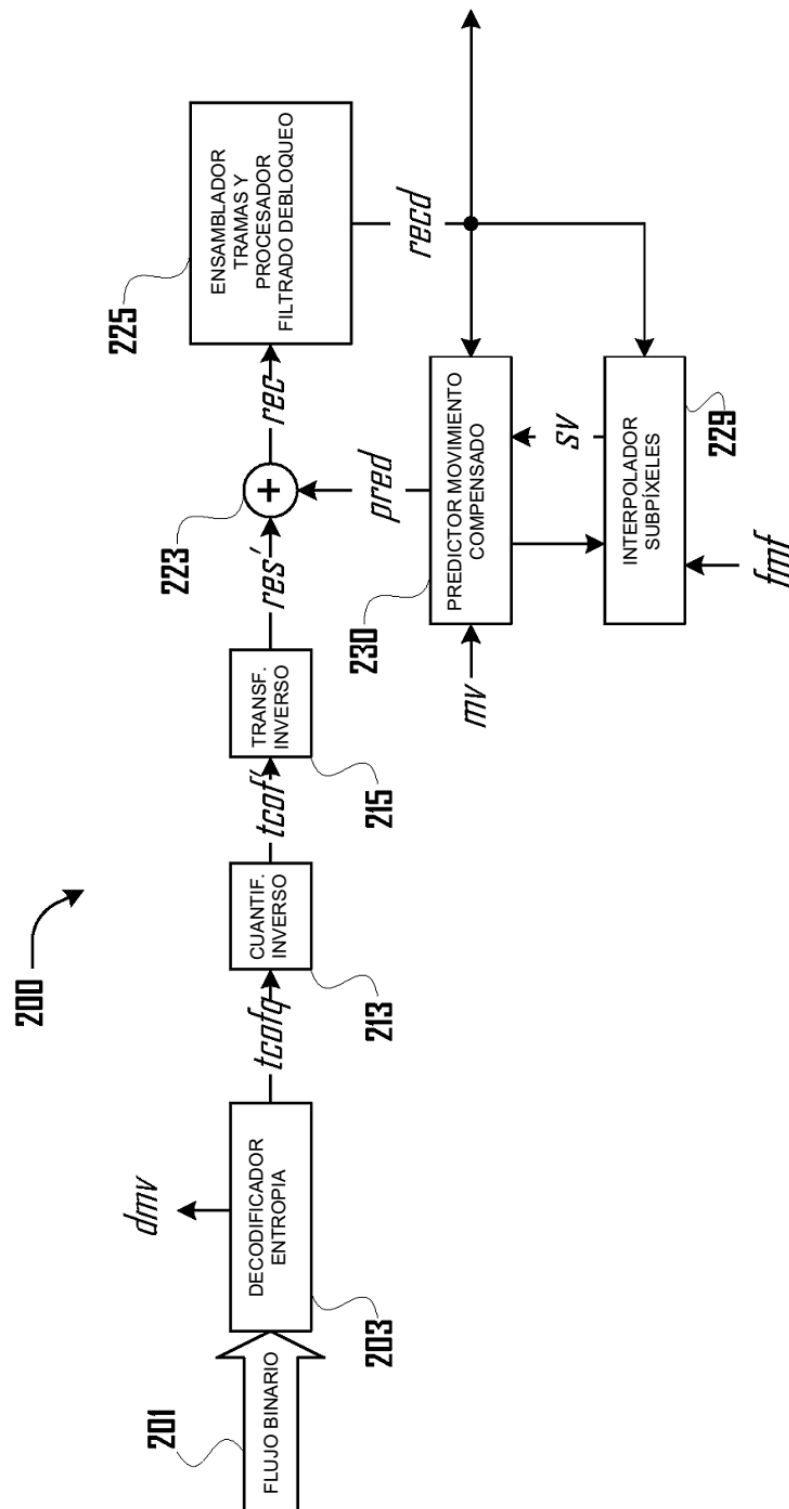


Fig. 2 (técnica anterior)

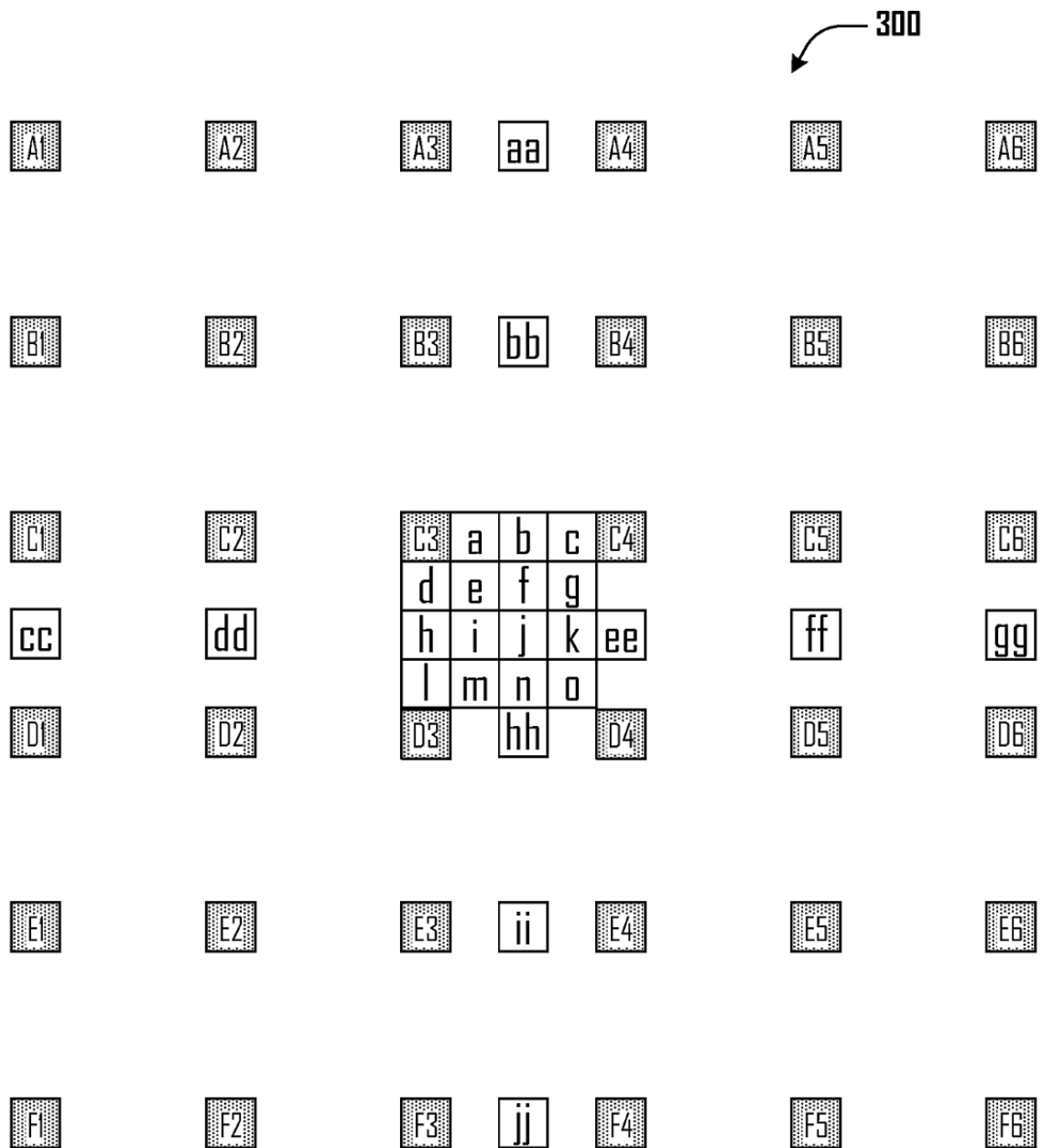


Fig. 3

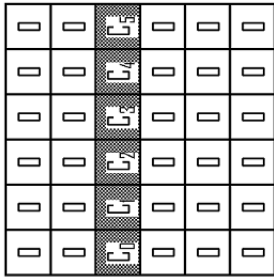


Fig. 4a

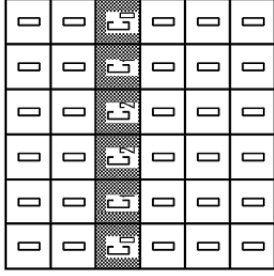


Fig. 4b

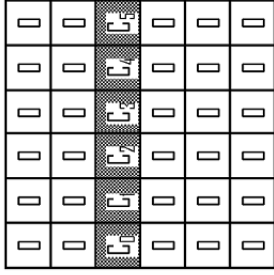


Fig. 4c

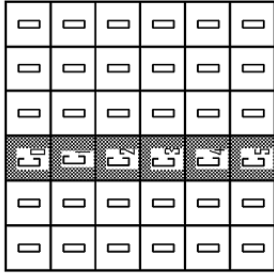


Fig. 4d

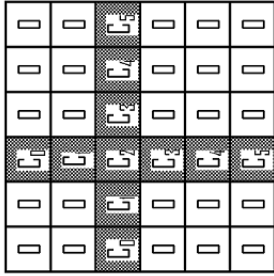


Fig. 4e

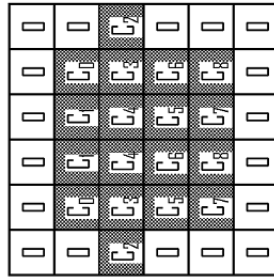


Fig. 4f

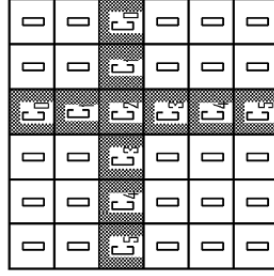


Fig. 4g

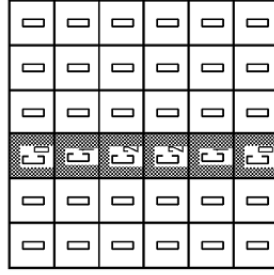


Fig. 4h

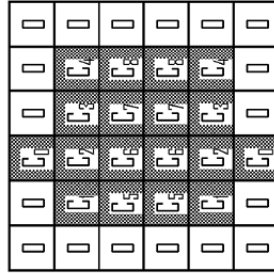


Fig. 4i

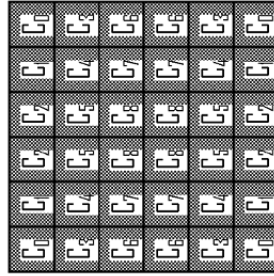


Fig. 4j

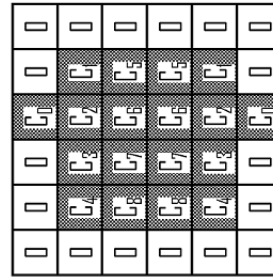


Fig. 4k

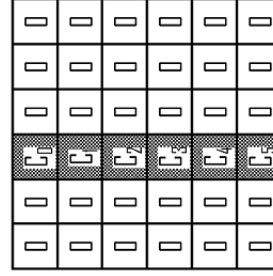


Fig. 4l

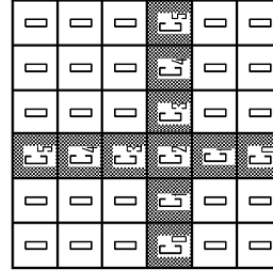


Fig. 4m

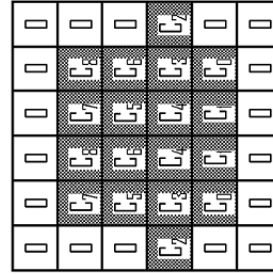


Fig. 4n

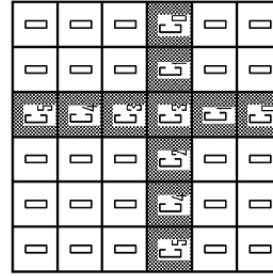


Fig. 4o

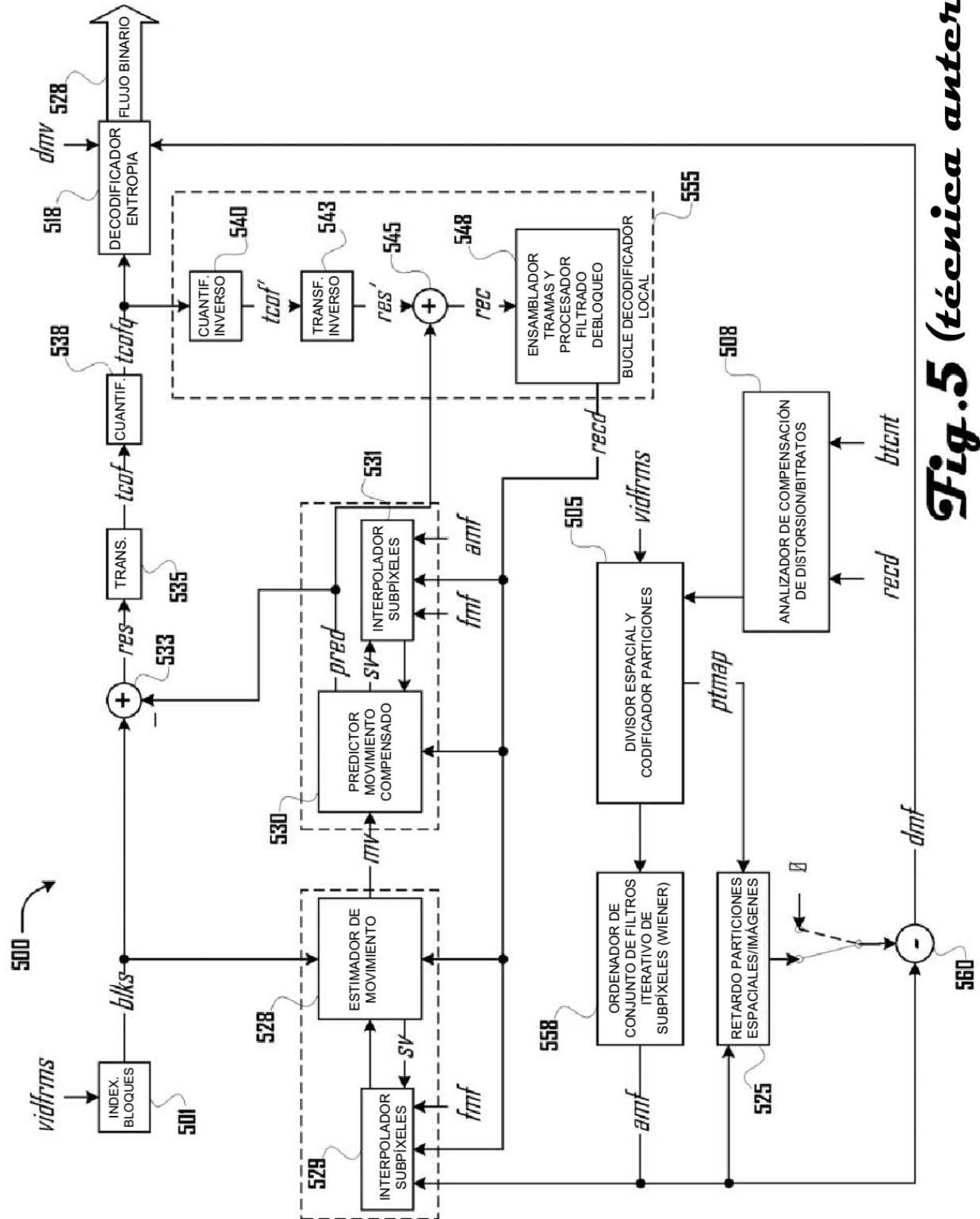


Fig. 5 (técnica anterior)

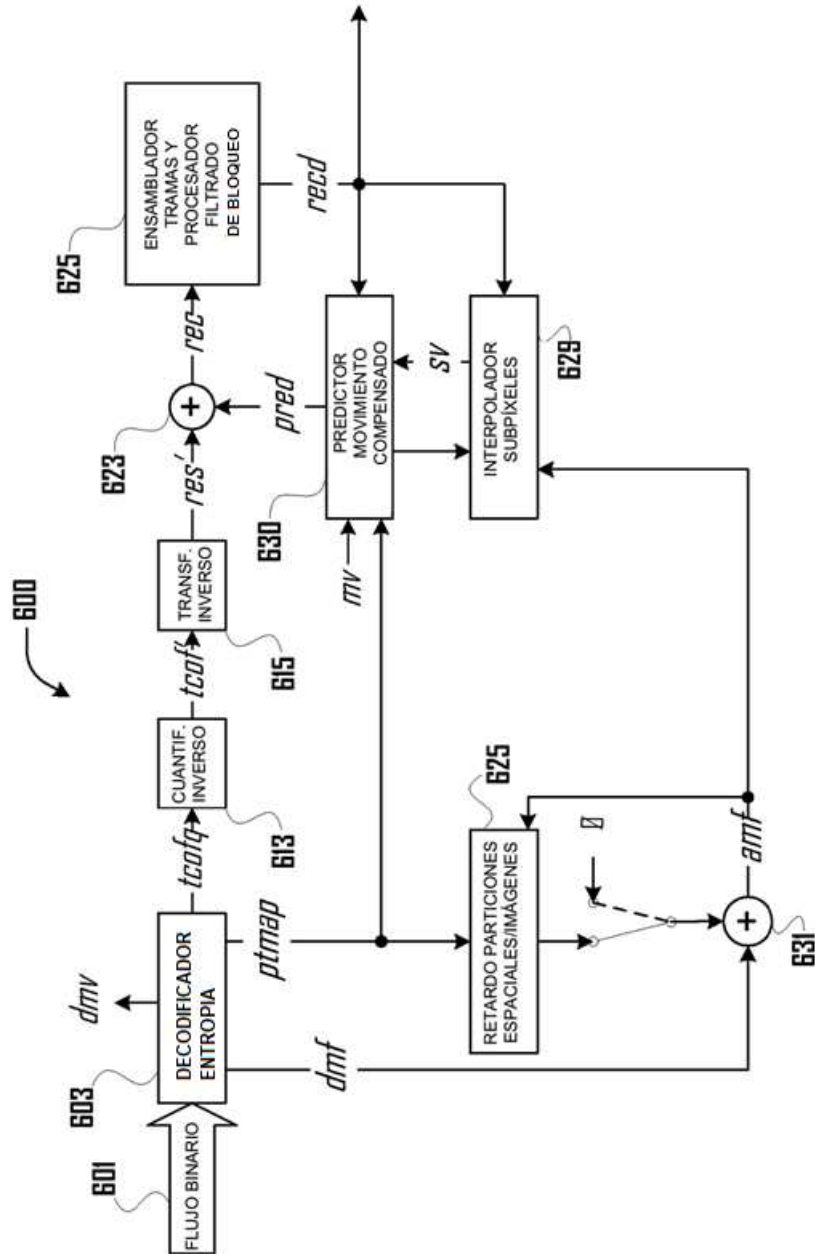


Fig. 6 (técnica anterior)



Fig. 7a

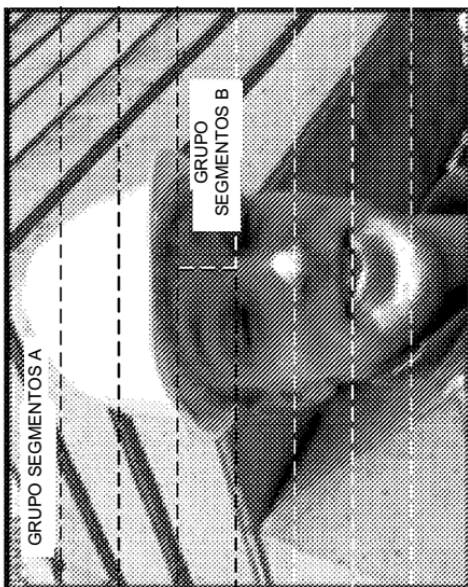


Fig. 7b-



Fig. 7c



Fig. 7c



Fig. 7b



Fig. 7d

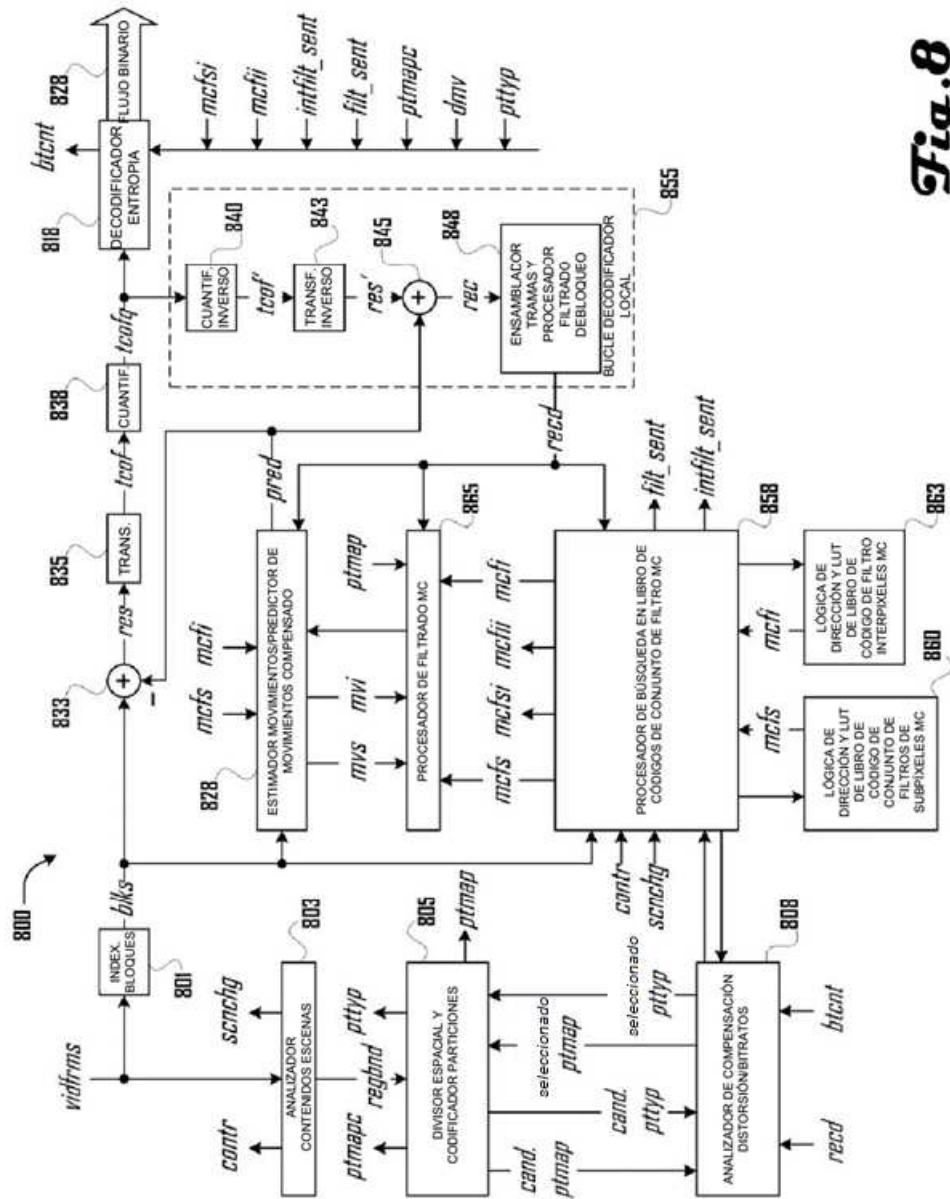


Fig. 8

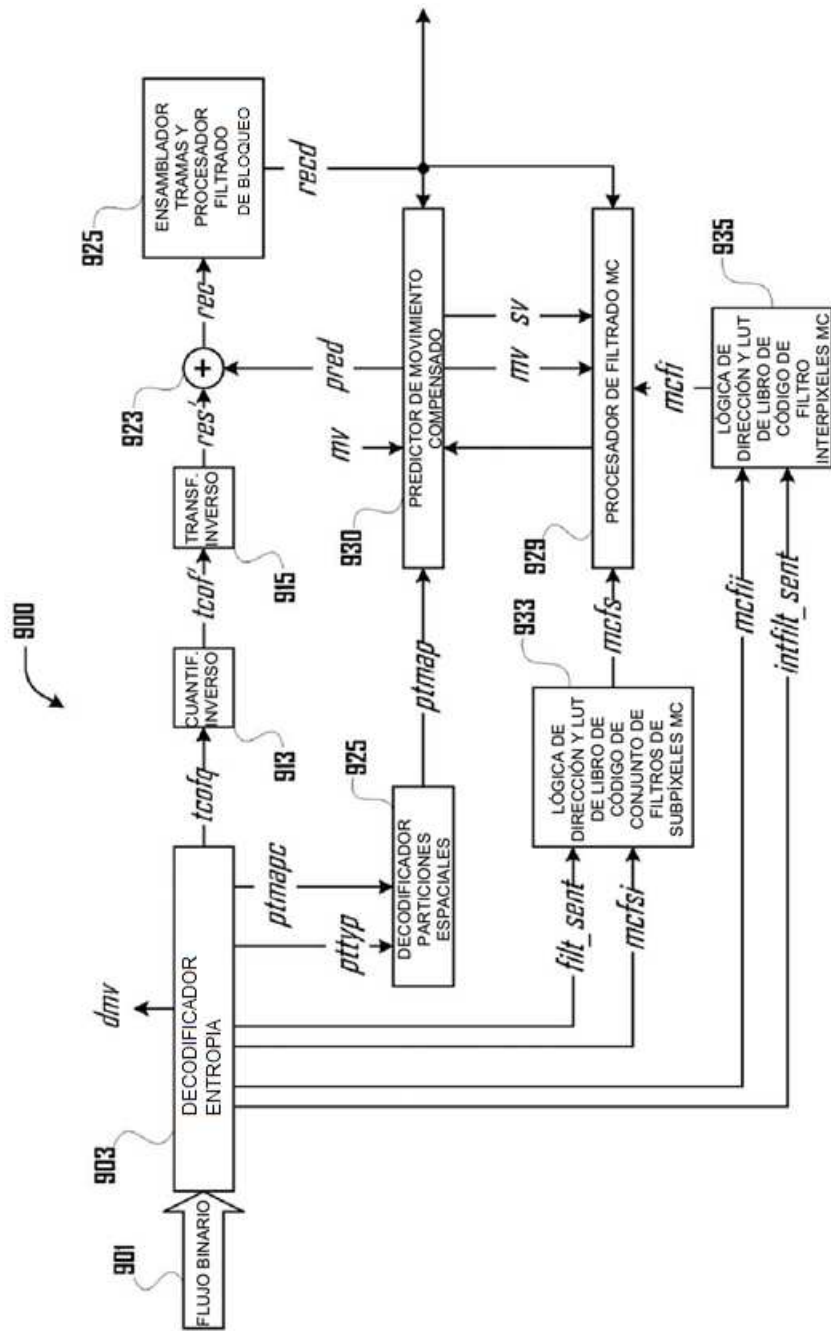


Fig. 9

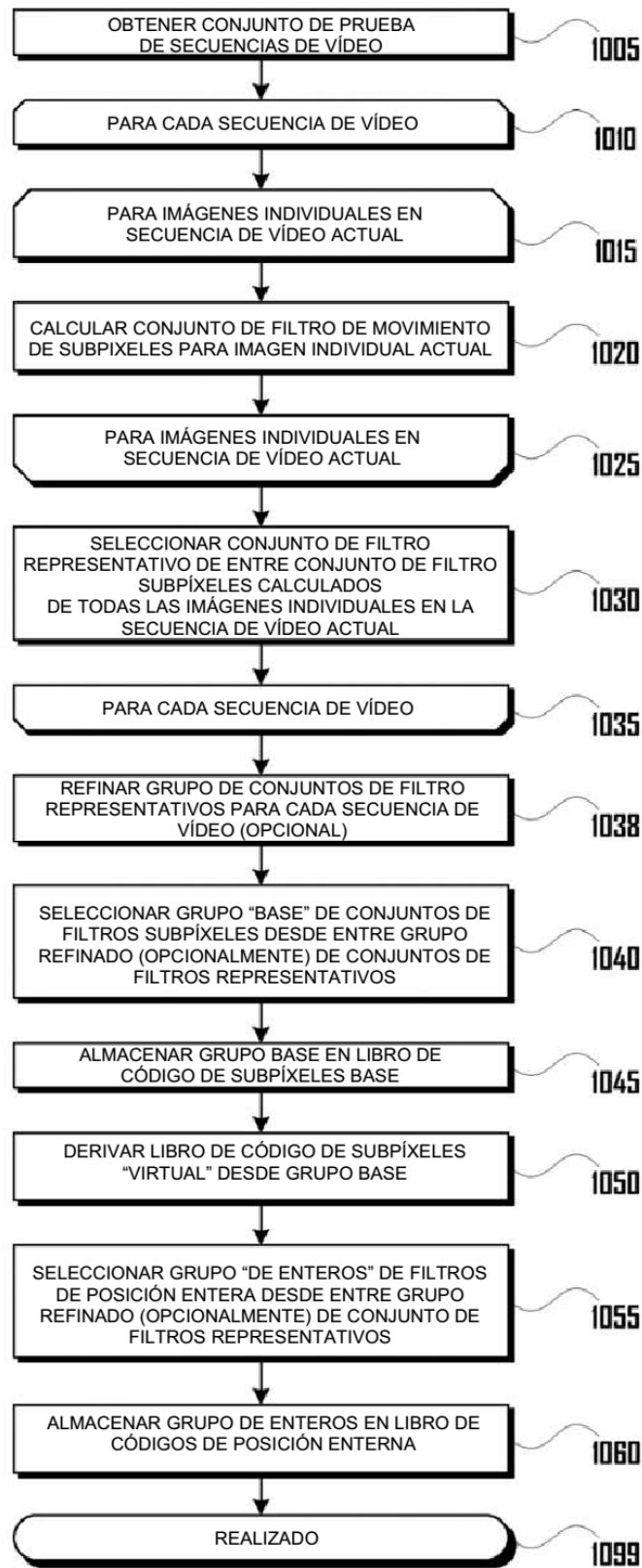


Fig.10

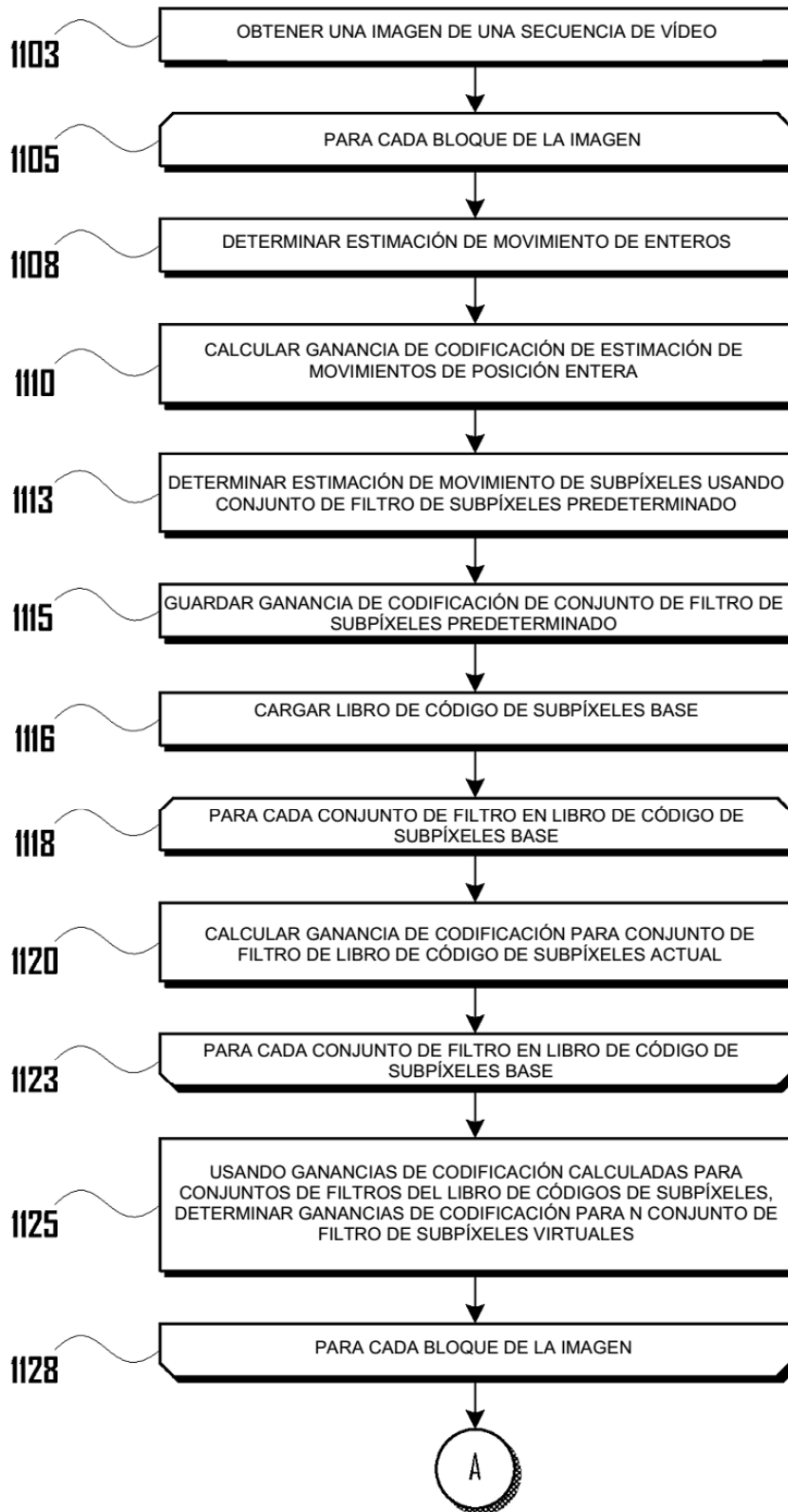


Fig.11

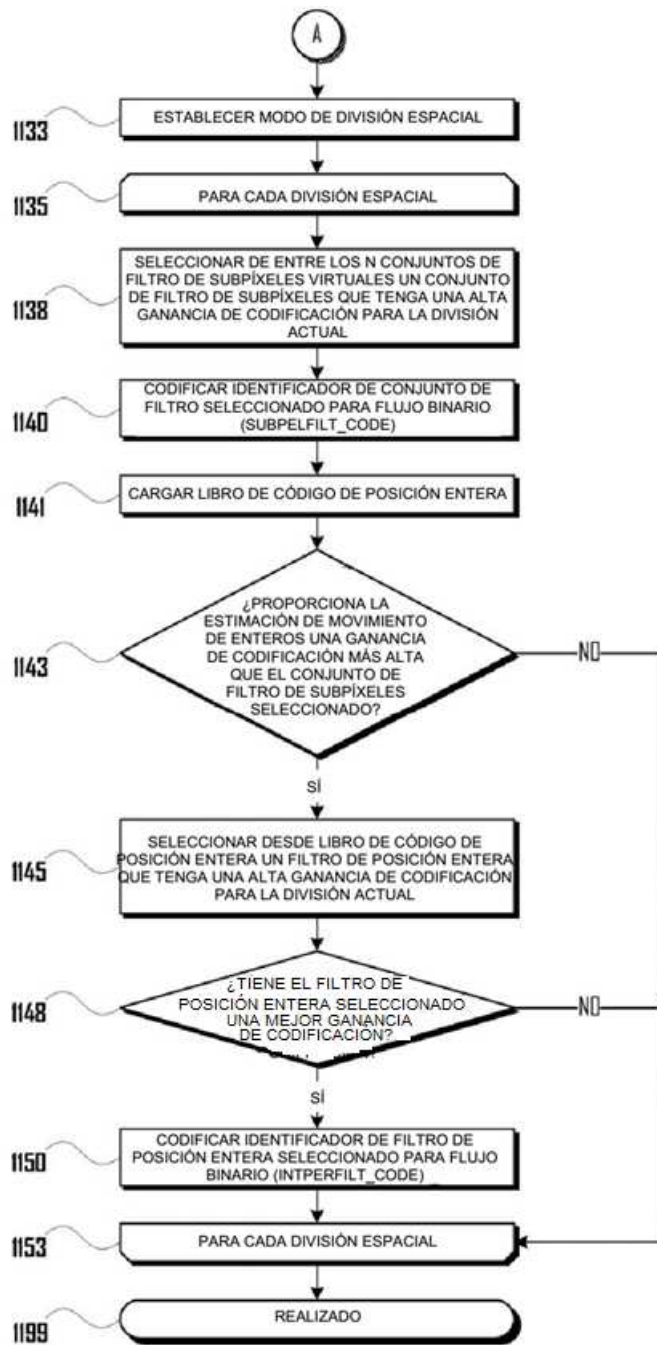


Fig.11 (cont.)

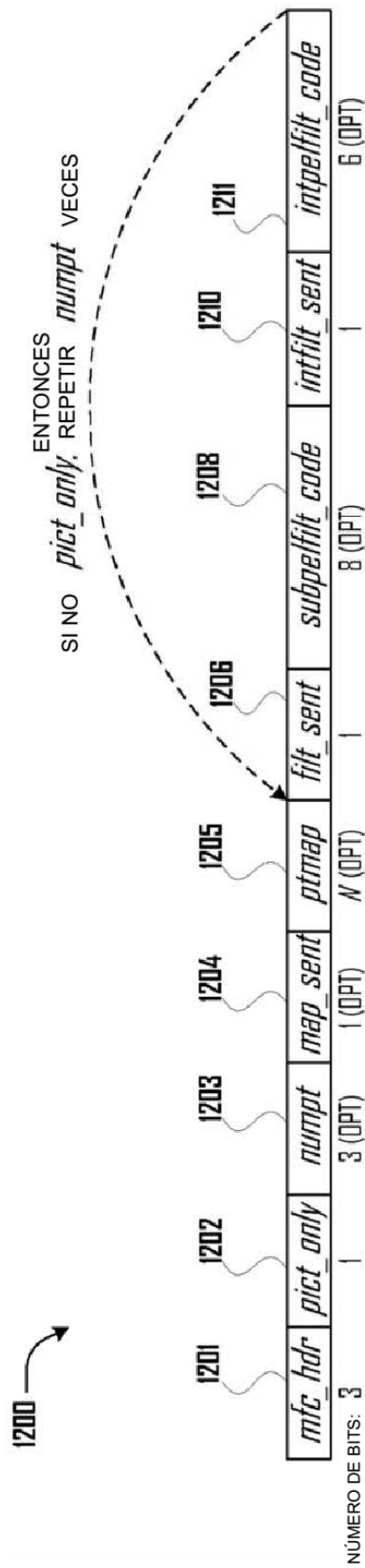


Fig.12

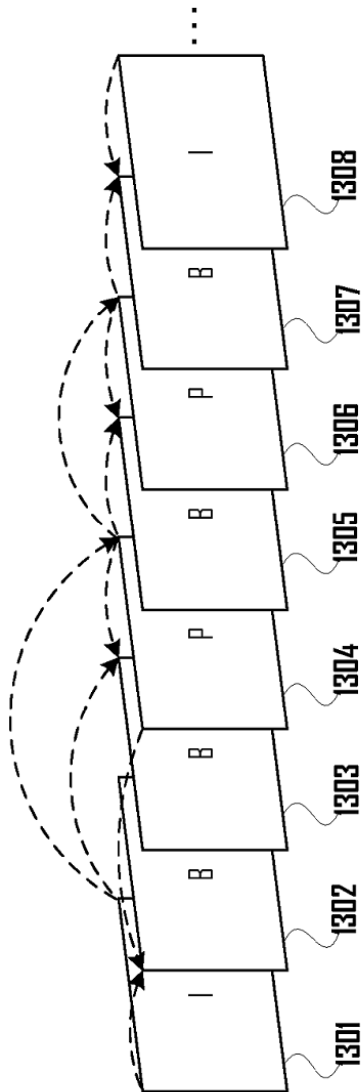


Fig. 13a

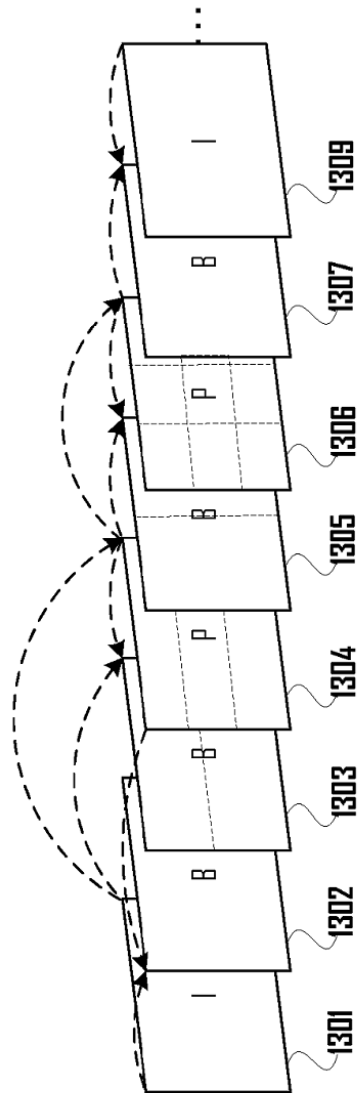


Fig. 13b

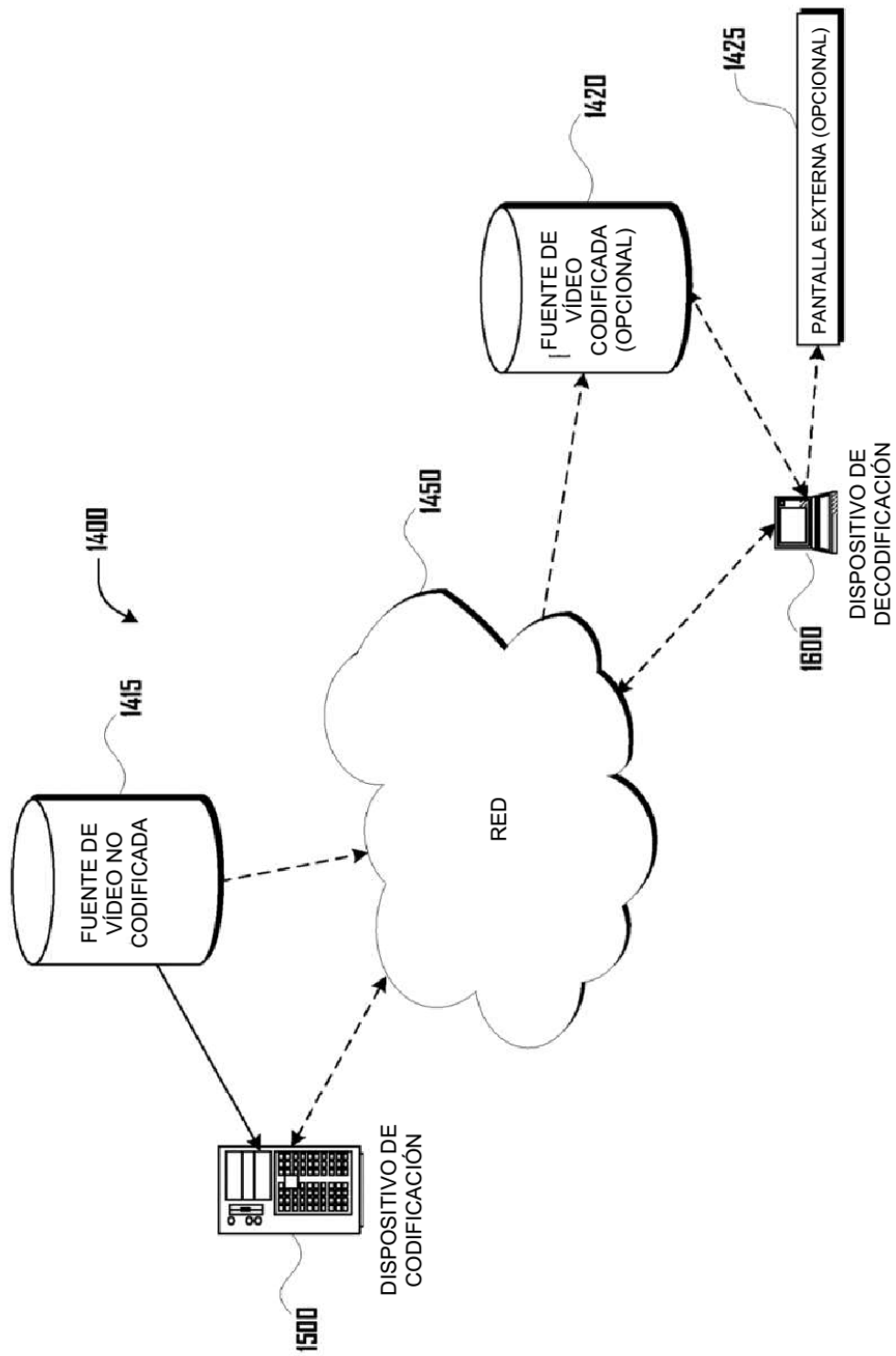


Fig. 14

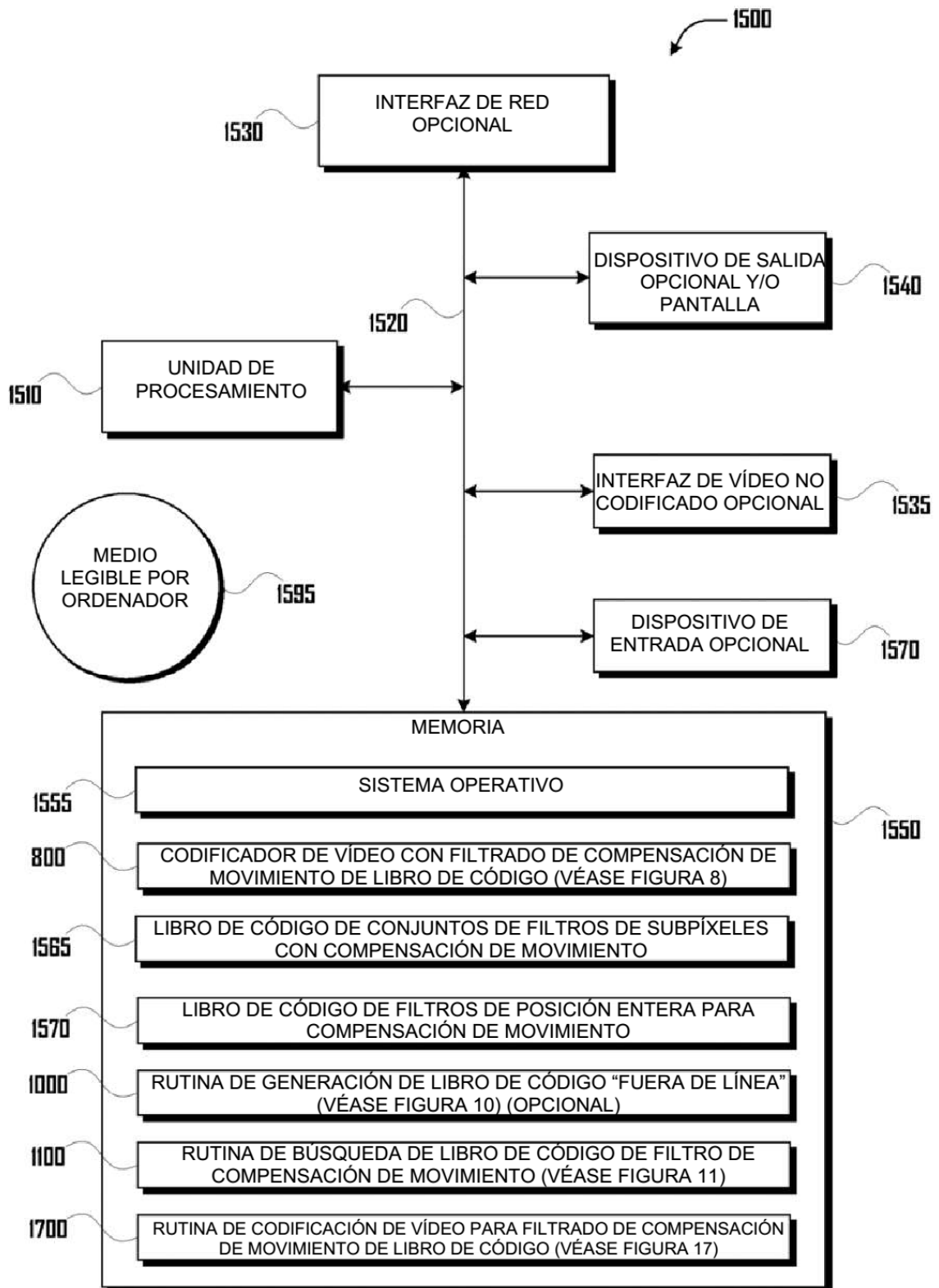


Fig.15

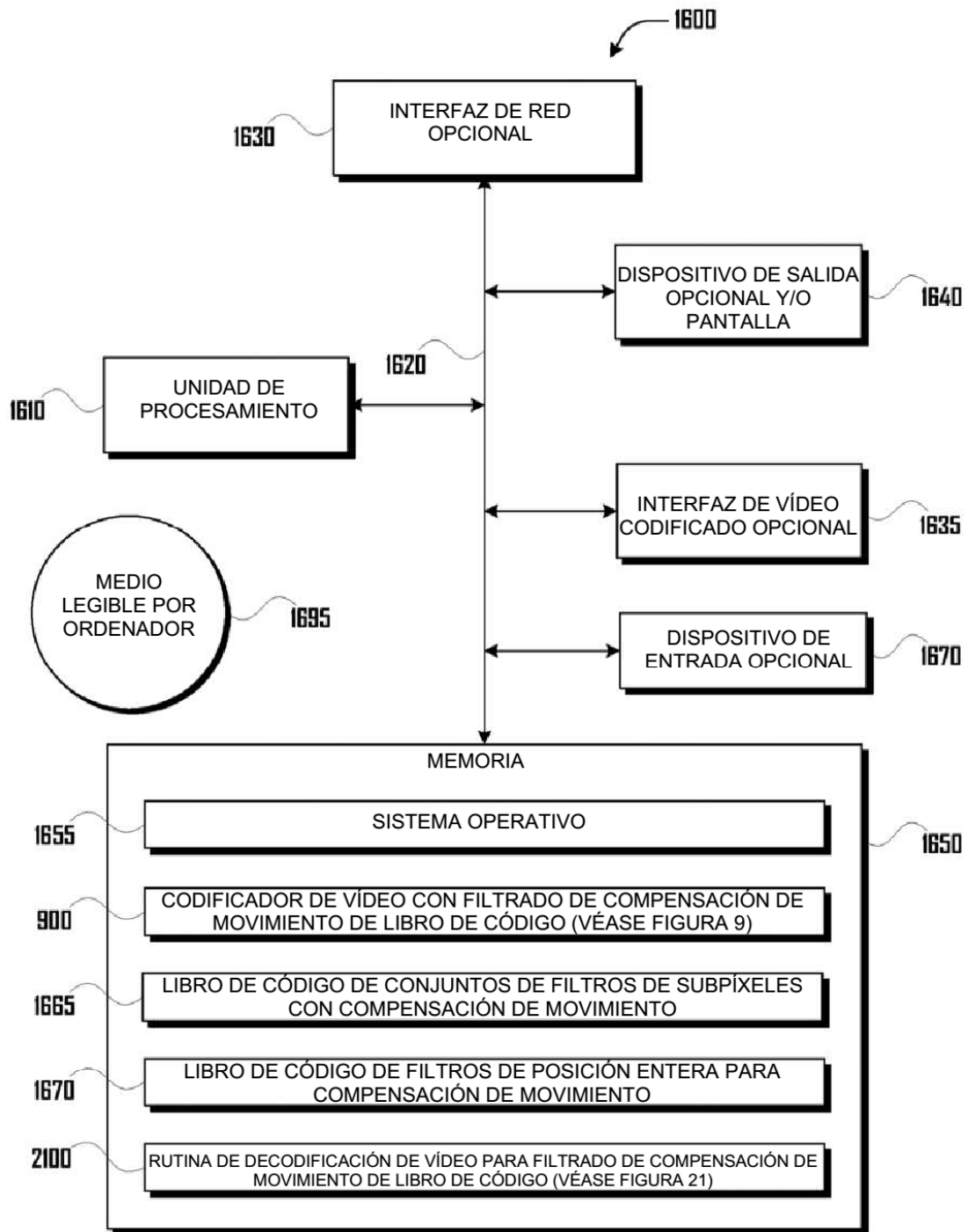


Fig.16

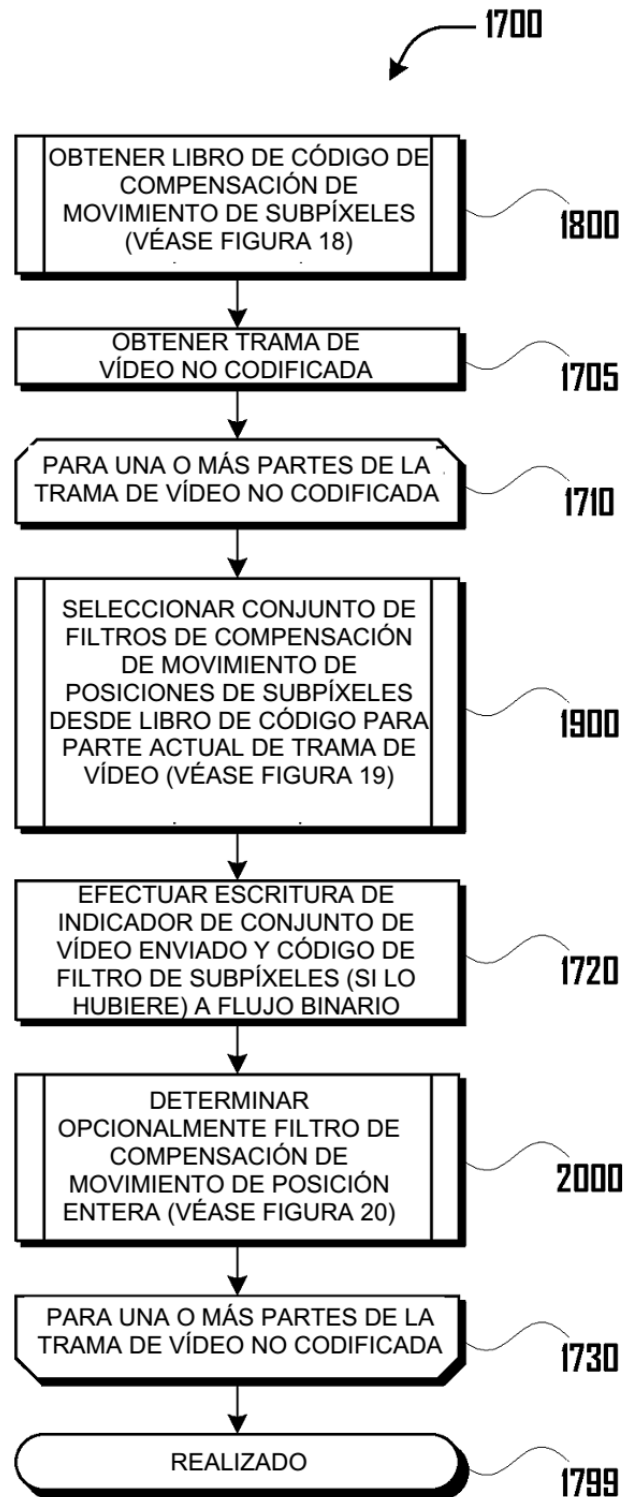


Fig.17

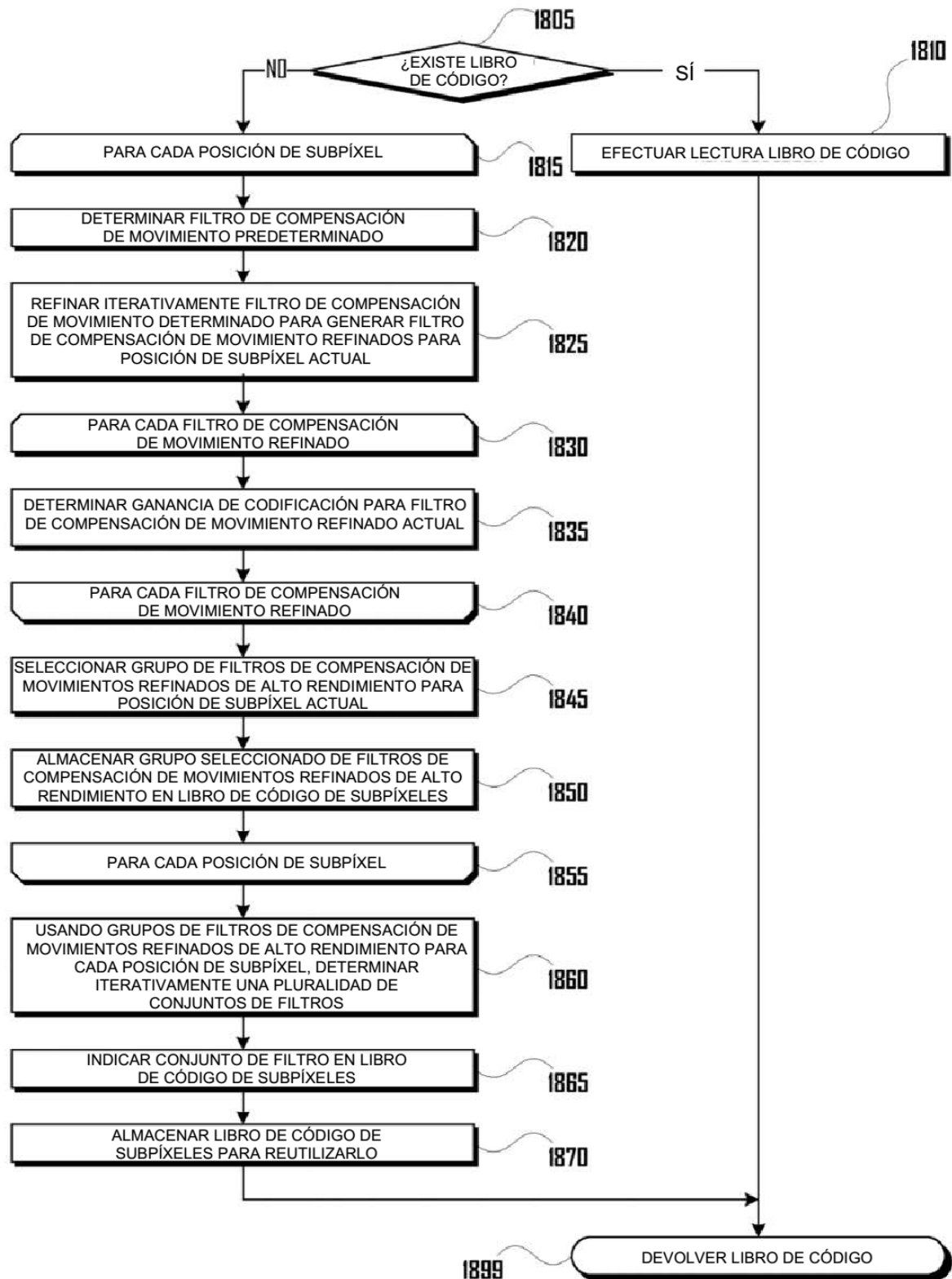
**Fig.18**



Fig.19

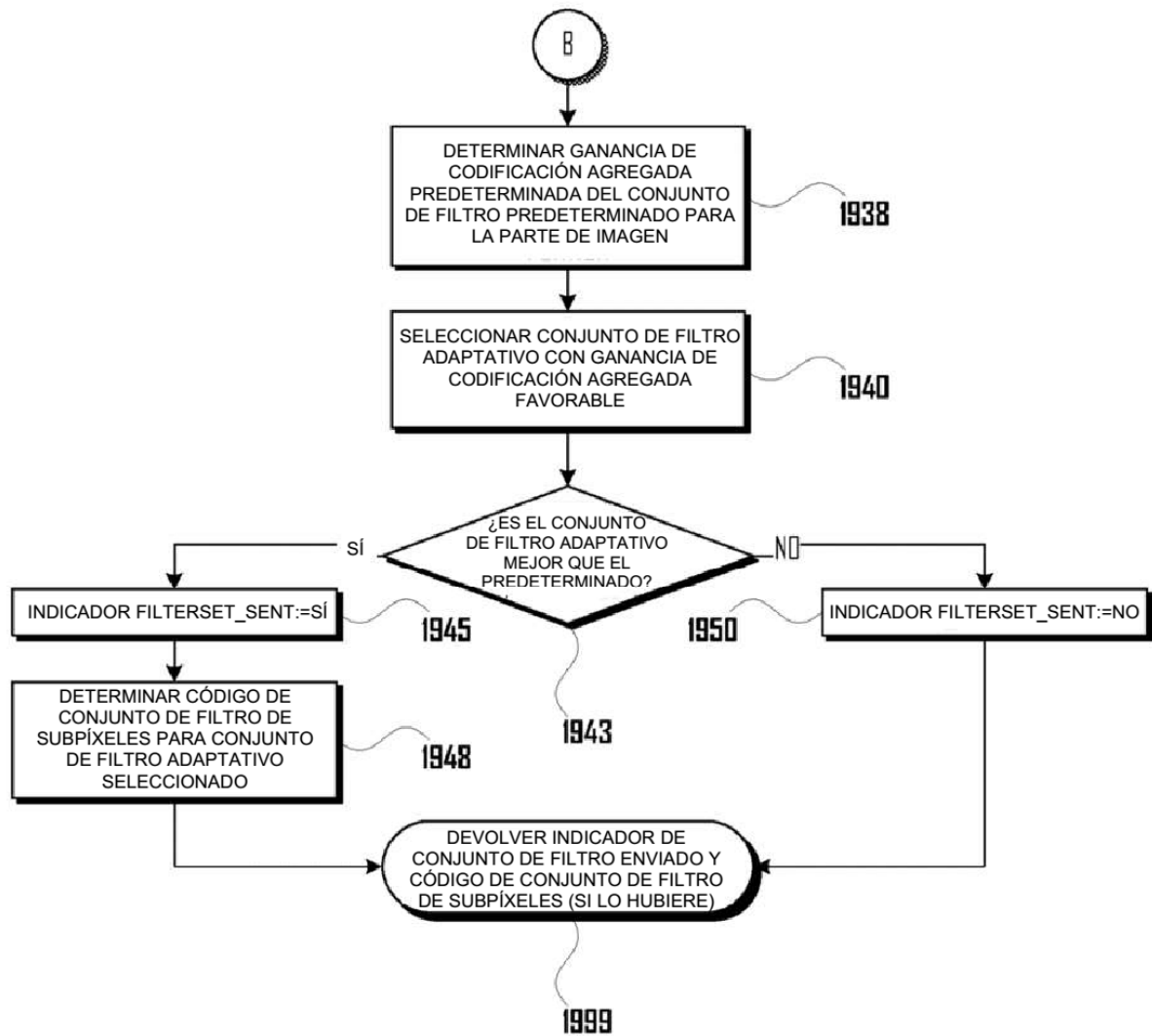


Fig.19 (cont.)

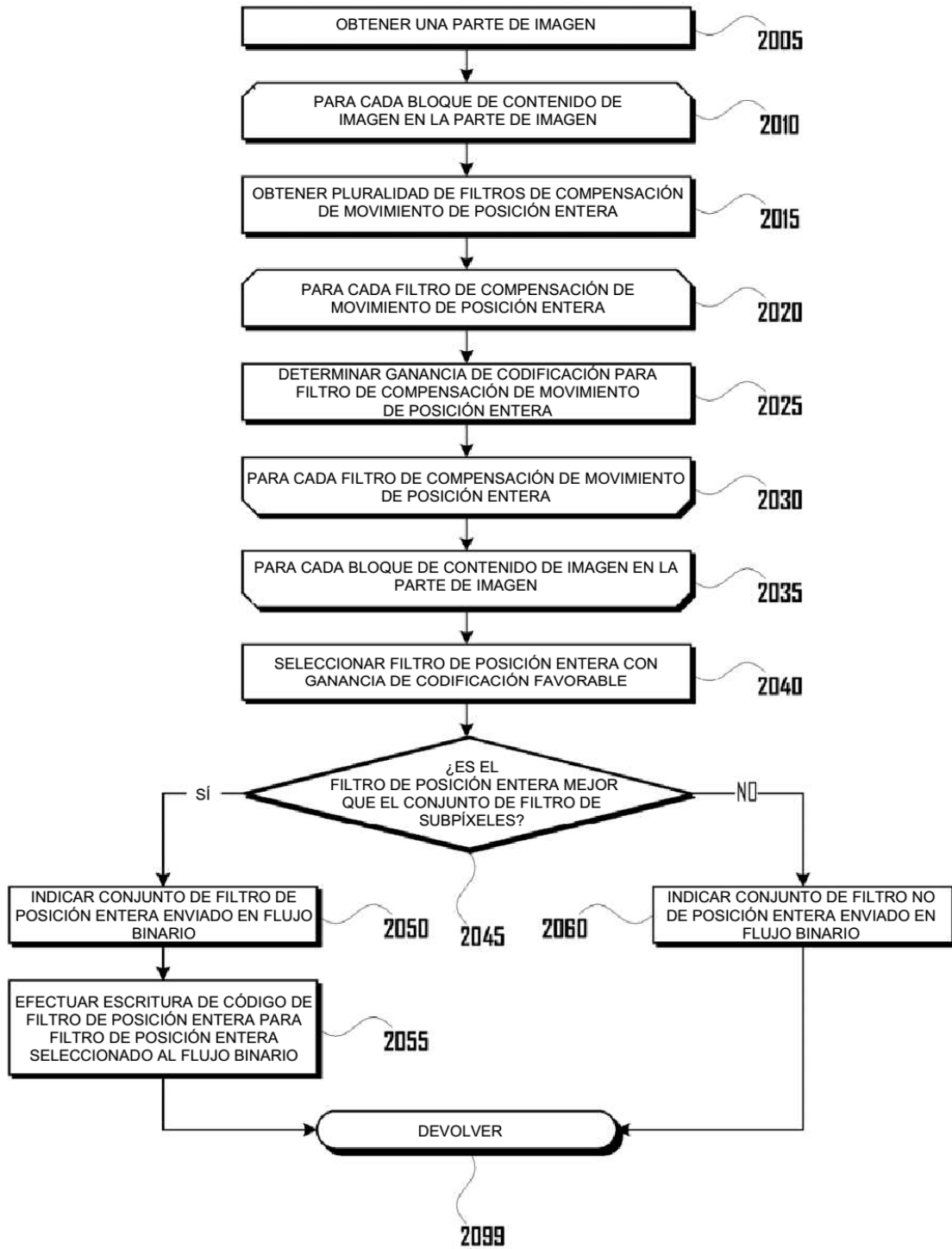


Fig.20

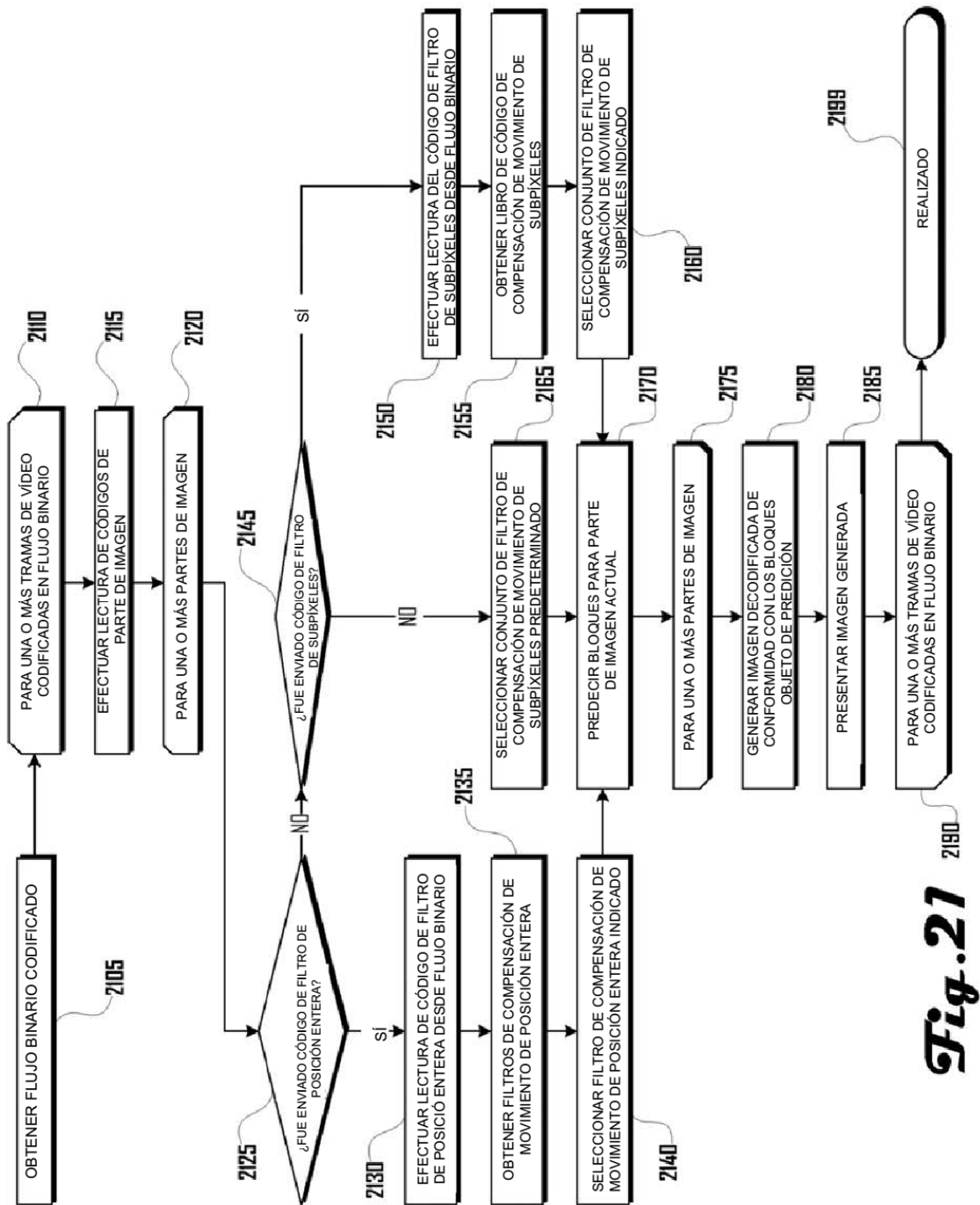


Fig. 27