

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 540 583**

51 Int. Cl.:

H04N 19/523 (2014.01)

H04N 19/80 (2014.01)

H04N 19/51 (2014.01)

G06T 3/40 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **11.09.2002 E 02758498 (6)**

97 Fecha y número de publicación de la concesión europea: **29.04.2015 EP 1433316**

54 Título: **Método para interpolación de valor de subpíxeles**

30 Prioridad:

17.09.2001 US 954608

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

10.07.2015

73 Titular/es:

**NOKIA TECHNOLOGIES OY (100.0%)
Karaportti 3
02610 Espoo, FI**

72 Inventor/es:

**KARCZEWICZ, MARTA y
HALLAPURO, ANTTI**

74 Agente/Representante:

VALLEJO LÓPEZ, Juan Pedro

ES 2 540 583 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

Método para interpolación de valor de subpíxeles

- 5 La presente invención se refiere a un método para interpolación de valor de subpíxeles en la codificación y decodificación de datos. Se refiere particularmente, pero no exclusivamente, a la codificación y decodificación de vídeo digital.

10 **Antecedentes de la Invención**

- 10 Las secuencias de vídeo digital, como las imágenes en movimiento ordinarias grabadas en película, comprenden una secuencia de imágenes quietas, creándose la ilusión de movimiento por la exhibición de las imágenes una después de la otra a una velocidad de trama relativamente rápida, normalmente de 15 a 30 tramas por segundo. Debido a la velocidad de trama relativamente rápida, las imágenes en las tramas consecutivas tienden a ser muy similares y, por lo tanto, contienen una cantidad considerable de información redundante. Por ejemplo, una escena típica puede comprender algunos elementos estacionarios, tales como el escenario de fondo, y algunas áreas en movimiento, que pueden adoptar muchas formas diferentes, por ejemplo, la cara de una persona que lee un periódico, tránsito en movimiento y así sucesivamente. Alternativamente, la cámara que graba la escena por sí misma puede estar en movimiento, en cuyo caso todos los elementos de la imagen tienen el mismo tipo de movimiento. En muchos casos, esto significa que el cambio global entre una trama de vídeo y la siguiente es más bien pequeño. Desde luego, esto depende de la naturaleza del movimiento. Por ejemplo, mientras más rápido es el movimiento, mayor es el cambio de una trama a la siguiente. De manera similar, si una escena contiene un número de elementos en movimiento, el cambio de una trama a la siguiente es probable que sea mayor que en una escena en donde sólo un elemento está en movimiento.

- 25 Se debe apreciar que cada trama de una secuencia de vídeo digital original, es decir no comprimida, comprende una cantidad muy grande de información de imagen. Cada trama de la secuencia de vídeo digital no comprimida está formada por una disposición de píxeles de imagen. Por ejemplo, en un formato de vídeo digital comúnmente usado, conocido como el formato de intercambio común de un cuarto (QCIF), una trama comprende una disposición de 176 x 144 píxeles, en cuyo caso cada trama tiene 25.344 píxeles. A su vez, cada píxel está representado por cierto número de bits, que llevan información acerca del contenido de luminancia y/o color de la región de la imagen correspondiente al píxel. Comúnmente, un denominado modelo de color YUV se usa para representar el contenido de luminancia y crominancia de la imagen. El componente de luminancia, o Y, representa la intensidad (brillo) de la imagen, mientras que el contenido de color de la imagen está representado por dos componentes de crominancia, marcados como U y V.

- 40 Los modelos de color basados en una representación de imagen/crominancia del contenido de imagen proporcionan ciertas ventajas comparadas con modelos de color que se basan en una representación que implica colores primarios (es decir, rojo, verde y azul, RGB). El sistema visual humano es más sensible a la variación de intensidad que a las variaciones de color; los modelos de color YUV explotan esta propiedad al usar una resolución espacial más baja para los componentes de crominancia (U, V) que para el componente de luminancia (Y). De esta manera, la cantidad de información necesaria para codificar la información de color en una imagen puede reducirse con una reducción aceptable en calidad de imagen.

- 45 La resolución espacial más baja de los componentes de crominancia generalmente se logra por submuestreo. Normalmente, un bloque de 16 x 16 píxeles de imagen está representado por un bloque de 16 x 16 píxeles que comprenden información de luminancia, y los componentes de crominancia correspondientes está representados cada uno por un bloque de 8 x 8 píxeles que representan un área de la imagen equivalente a los 16 x 16 píxeles del componente de luminancia. Los componentes de crominancia, por lo tanto, son submuestreados espacialmente por un factor de 2 en las direcciones x e y. El conjunto resultante de un bloque de luminancia de 16 x 16 píxeles y dos bloques de crominancia de 8 x 8 píxeles comúnmente se refiere como un macrobloque de YUV o macrobloque, para brevedad.

- 55 Una imagen de QCIF comprende 11 x 9 macrobloques. Si los bloques de luminancia y los bloques de crominancia están representados con una resolución de 8 bits (es decir por números en el intervalo de 0 a 255), el número total de bits requeridos para el macrobloque es $(16 \times 16 \times 8) + 2 \times (8 \times 8 \times 8) = 3072$ bits. El número de bits necesarios para representar una trama de vídeo en formato QCIF es, por lo tanto, de $99 \times 3072 = 304.128$ bits. Esto significa que la cantidad de datos requeridos para transmitir/grabar/exhibir una secuencia de vídeo en formato QCIF, representado mediante el uso de un modelo de color YUV, a una velocidad de 30 tramas por segundo, es más de 9 Mbps (millones de bits por segundo). Esta es una velocidad de datos extremadamente alta y no es práctica para usarse en aplicaciones de grabación, transmisión y exhibición de vídeo debido a la capacidad de almacenamiento muy grande, la capacidad de canal de transmisión y el rendimiento de hardware requerido.

- 65 Si los datos de vídeo han de ser transmitidos en tiempo real en una red de línea fija tal como una ISDN (Red Digital de Servicios Integrados) o una PSTN convencional (Red de Telefonía de Servicios Públicos), la amplitud de banda de transmisión de datos disponible es normalmente del orden de 64kbits/s. En videotelefonía móvil, donde la

transmisión tiene lugar por lo menos en parte sobre un enlace de comunicaciones de radio, la amplitud de banda disponible puede ser tan baja como de 20kbits/s. Esto significa que una reducción significativa en la cantidad de información usada para representar datos de video se debe lograr para permitir la transmisión de secuencias de video digital en redes de comunicación de baja amplitud de banda. Por esta razón, se han desarrollado técnicas de compresión de video que reducen la cantidad de información transmitida mientras retienen una calidad de imagen aceptable.

Los métodos de compresión de video se basan en la reducción de las partes redundantes y perceptualmente irrelevantes de secuencias de video. La redundancia en secuencias de video se puede categorizar en redundancia espacial, temporal y espectral. La "redundancia espacial" es el término usado para describir la correlación entre píxeles vecinos dentro de una trama. El término "redundancia temporal" expresa el hecho de que los objetos que aparecen en una trama de una secuencia probablemente aparezcan en tramas posteriores, mientras que la "redundancia espectral" se refiere a la correlación entre diferentes componentes de color de la misma imagen.

La compresión suficientemente eficiente generalmente no se puede lograr reduciendo simplemente las formas diversas de redundancia en una secuencia de imágenes dada. Por lo tanto, la mayoría de los codificadores de video actuales reducen la calidad de aquellas partes de la frecuencia de video que son subjetivamente las menos importantes. Además, la redundancia de la corriente de bits de video comprimido es por sí misma reducida mediante codificación de menos pérdida eficiente. Normalmente, esto se logra mediante el uso de una técnica conocida como "codificación de longitud variable" (VLC).

Estándares de compresión de video modernos, tales como las recomendaciones H.261, H.263 (+) (++), H.26L de la ITU-T y la recomendación MPEG-4 del Grupo de Expertos de Imágenes en Movimiento hacen uso de la "predicción temporal compensada en movimiento". Esta es una forma de reducción de redundancia temporal en la cual el contenido de algunas tramas (con frecuencia muchas) en una secuencia de video se "predice" a partir de otras tramas en la secuencia al rastrear el movimiento de objetos o regiones de imágenes entre tramas.

Las imágenes comprimidas que no hacen uso de reducción de redundancia temporal generalmente se denominan tramas INTRA-codificadas o tramas-I, mientras que las imágenes temporalmente predichas se denominan tramas INTER-codificadas o tramas-P. En el caso de las tramas intercodificadas, la imagen predicha (compensada en cuanto a movimiento) es rara vez lo suficientemente precisa para representar el contenido de imagen con cantidad suficiente y, por lo tanto, un error de predicción (PE) espacialmente comprimido está también asociado con cada trama intercodificada. Muchos esquemas de compresión de video también pueden hacer uso de tramas bidireccionalmente predichas, que comúnmente son denominadas como imágenes-B o tramas-B. Las imágenes-B son insertadas entre pares de imágenes de referencia o denominadas "de anclaje" (imágenes I o P) y son predichas a partir de cualquiera o ambas de las imágenes de anclaje. Las imágenes-B no son usadas por sí mismas como imágenes de anclaje, es decir no predicen otras tramas a partir de las mismas, y por lo tanto pueden ser descartadas de la secuencia de video sin causar un deterioro en la calidad de imágenes futuras.

Los diferentes tipos de trama que se producen en una secuencia de video comprimida típica se ilustran en la figura 3 de los dibujos adjuntos. Como se puede ver a partir de la figura, la secuencia empieza con la trama intracodificada o trama I, 30. En la figura 3, las flechas 33 indican el proceso de predicción "hacia adelante", por el cual se forman las tramas-P (marcadas 34). El proceso de predicción bidireccional por el cual se forman las tramas B (36) se indica mediante las flechas 31a y 31b, respectivamente.

Un diagrama esquemático de un sistema de codificación de video de ejemplo que usa predicción compensada en cuanto a movimiento se muestra en las figuras 1 y 2. La figura 1 ilustra un codificador 10 que emplea compensación de movimiento y la figura 2 ilustra un decodificador 20 correspondiente. El codificador 10 mostrado en la figura 1 comprende un bloque de estimación de campo de movimiento 11, un bloque de codificación de campo de movimiento 12, un bloque de predicción compensada en cuanto a movimiento 13, un bloque de codificación de error de predicción 14, un bloque de codificación de error de predicción 15, un bloque de multiplexión 16, una memoria de trama 17 y un agregador 19. El decodificador 20 comprende un bloque de predicción compensada en cuanto a movimiento 21. Un bloque de codificación de error de predicción 22, un bloque de desmultiplexión 23 y una memoria de trama 24.

El principio de operación de los codificadores de video que usan compensación de movimiento es reducir al mínimo la cantidad de información en una trama de error de predicción $E_n(x,y)$, que es la diferencia entre una trama actual $I_n(x,y)$ que se codificada y una trama de predicción $P_n(x,y)$. La trama de error de predicción es, por lo tanto:

$$E_n(x, y) = I_n(x, y) - P_n(x, y). \quad (1)$$

La trama de predicción $P_n(x,y)$ se construye mediante el uso de valores de píxeles de una trama de referencia $R_n(x,y)$, que es generalmente una de las tramas previamente codificadas y transmitidas, por ejemplo, la trama que precede inmediatamente a la trama actual y está disponible de la memoria de trama 17 del codificador 10. Más específicamente, la trama de predicción $P_n(x,y)$ se construye al encontrar los llamados "píxeles de predicción" en la

trama de referencia $R_n(x,y)$ que corresponden sustancialmente con píxeles en la trama actual. La información de movimiento, que describe la relación (por ejemplo, localización relativa, rotación, escala, etc.) entre píxeles en la trama actual y sus píxeles de predicción correspondientes en la trama de referencia se deriva y la trama de predicción se construye al mover los píxeles de predicción de acuerdo con la información de movimiento. De esta manera, la trama de predicción se construye como una representación aproximada de la trama actual, mediante el uso de valores de píxeles en la trama de referencia. La trama de error de predicción referida anteriormente, por lo tanto, representa la diferencia entre la representación aproximada de la trama actual proporcionada por la trama de predicción y la trama misma actual. La ventaja básica proporcionada por los codificadores de video que usan predicción compensada en cuanto a movimiento surge del hecho de que la descripción comparativamente compacta de la trama actual se puede obtener al representarlo en términos de la información de movimiento requerida para formar su predicción junto con la información de error de predicción asociada en la trama de error de predicción.

Sin embargo, debido al número muy grande de píxeles en una trama, generalmente no es eficiente transmitir información de movimiento separada para cada píxel al decodificador. En lugar de ello, en la mayoría de los esquemas de codificación de video, la trama actual se divide en grandes segmentos de imagen S_k y la información de movimiento relacionada con los segmentos es transmitida al decodificador. Por ejemplo, la información de movimiento se proporciona normalmente mediante cada monobloque de una trama y la misma información de movimiento se usa después para todos los píxeles dentro del macrobloque. En algunos estándares de codificación de video, tales como H.26L, un macrobloque se puede dividir en bloques más pequeños, cada bloque más pequeño estando provisto de su propia información de movimiento.

La información de movimiento generalmente adopta la forma de vectores de movimiento $[\Delta x(x,y), \Delta y(x,y)]$. El par de números $\Delta x(x,y)$ y $\Delta y(x,y)$ representan los desplazamientos horizontal y vertical de un píxel en el sitio (x,y) en la trama actual $I_n(x,y)$ con respecto a un píxel en la trama de referencia $R_n(x,y)$. Los vectores de movimiento $[\Delta x(x,y), \Delta y(x,y)]$ se calculan en el bloque de estimación de campo de movimiento 11 y el conjunto de vectores de movimiento de la trama actual $[\Delta x(\cdot), \Delta y(\cdot)]$ se refiere como el campo de vector de movimiento.

Normalmente, la posición de un macrobloque en una trama de video actual se especifica mediante la coordenada (x,y) de su esquina izquierda superior. Por lo tanto, en un esquema de codificación de video en el cual la información de movimiento está asociada con cada macrobloque de una trama, cada vector de movimiento describe el desplazamiento horizontal y vertical $\Delta x(x,y)$ y $\Delta y(x,y)$ de un píxel que representa la esquina superior izquierda de un macrobloque en la trama actual $I_n(x,y)$ con respecto a un píxel en la esquina superior izquierda de un bloque sustancialmente correspondiente de píxeles de predicción en la trama de referencia $R_n(x,y)$ (como se muestra en la figura 4b).

La estimación de movimiento es una tarea computacionalmente intensiva. Dada una trama de referencia $R_n(x,y)$ y, por ejemplo, un macrobloque cuadrado que comprende $N \times N$ píxeles en una trama actual (como se muestra en la figura 4a), el objetivo de estimación de movimiento es encontrar un bloque de $N \times N$ píxeles en la trama de referencia que coincida con las características del macrobloque en la imagen actual de acuerdo con cierto criterio. El criterio puede ser, por ejemplo, una suma de diferencias absolutas (SAD) entre los píxeles del macrobloque en la trama actual y el bloque de píxeles en la trama de referencia con la que se compara. Este proceso es generalmente conocido como "coincidencia de bloques". Debe indicarse que, en general, la geometría del bloque que ha de coincidir y la de la trama de referencia no tienen que ser la misma, ya que los objetos del mundo real pueden sufrir cambios de escala, así como rotación y trama. Sin embargo, en estándares de codificación de video internacionales actuales, sólo se usa un modelo de movimiento de translación (véase más adelante) y, por lo tanto, la geometría rectangular fija es suficiente.

De manera ideal, para lograr la mejor probabilidad de encontrar una coincidencia, se debe buscar la trama de referencia en conjunto. Sin embargo, esto no es práctico, ya que impone una carga computacional demasiado alta sobre el codificador de video. En lugar de ello, la región de búsqueda se restringe a la región $[-p,p]$ alrededor de la posición original del macrobloque en la trama actual, como se muestra en la figura 4c.

Para reducir la cantidad de información de movimiento que ha de ser transmitida desde el codificador 10 al codificador 20, el campo de vector de movimiento es codificado en el bloque de codificación de campo de movimiento 12 del codificador 10, representándolo con un modelo de movimiento. En este proceso, los vectores de movimiento de segmentos de imagen son expresados nuevamente mediante el uso de ciertas funciones predeterminadas o, en otras palabras, el campo de vector de movimiento representado con un modelo. Casi todos los modelos de campo de vector de movimiento actualmente usados son modelos de movimiento aditivos, que cumplen con la siguiente fórmula general:

$$\Delta x(x, y) = \sum_{i=0}^{N-1} a_i f_i(x, y) \quad (2)$$

$$\Delta y(x, y) = \sum_{i=0}^{M-1} b_i g_i(x, y) \quad (3)$$

donde los coeficientes a_i y b_i se denominan coeficientes de movimiento. Los coeficientes de movimiento son transmitidos al decodificador 20 (corriente de información 2 en las figuras 1 y 2). Las funciones f_i y g_i se denominan funciones de base de campo de movimiento, y se conocen tanto para el codificador como para el decodificador. Un

campo de vector de movimiento aproximado $(\tilde{\Delta x}(x, y), \tilde{\Delta y}(x, y))$ se puede construir mediante el uso de coeficientes y las funciones de base. Puesto que las funciones de base se conocen (es decir, almacenadas en) para el codificador 10 y el decodificador 20, sólo los coeficientes de movimiento necesitan ser transmitidos al codificador, lo que reduce así la cantidad de información requerida para representar la información de movimiento de la trama.

El modelo de movimiento más simple es el modelo de movimiento de traslación que requiere sólo dos coeficientes para describir los vectores de movimiento de cada segmento. Los valores de vectores de movimiento están dados por:

$$\begin{aligned} \Delta x(x, y) &= a_0 \\ \Delta y(x, y) &= b_0 \end{aligned} \quad (4)$$

Este modelo se usa ampliamente en varios estándares internacionales (ISO MPEG-1, MPEG-2, MPEG-4, recomendaciones H.261 y H.263) de ITU-T para describir los bloques de 16 x 16 y 8 x 8 píxeles. Los sistemas que usan un modelo de movimiento de traslación normalmente realizan una estimación de movimiento a una resolución de píxeles completa o alguna fracción integral de resolución de píxeles completa, por ejemplo, a la mitad o a un cuarto de la resolución de píxeles.

La trama de predicción $P_n(x, y)$ se construye en el bloque de predicción compensado en cuanto a movimiento 13 en el codificador 10 y está dada por:

$$P_n(x, y) = R_n [x + \tilde{\Delta x}(x, y), y + \tilde{\Delta y}(x, y)] \quad (5)$$

En el bloque de codificación de error de predicción 14, la trama de error de predicción $E_n(x, y)$ es normalmente comprimida al representarla como una serie finita (transformada) de algunas funciones bidimensionales. Por ejemplo, se puede usar una transformada de coseno discreta bidimensional (DTC). Los coeficientes de la transformada se cuantifican y la entropía (por ejemplo, de Huffman) se codifican antes de que sean transmitidas al decodificador (corriente de información 1 en las figuras 1 y 2). Debido al error introducido por cuantificación, esta operación generalmente produce cierta degradación (pérdida de información) en la trama de error de predicción $E_n(x, y)$. Para compensar esta degradación, el codificador 10 también comprende un bloque de codificación de error

de predicción 15, donde una trama de error de predicción decodificada $\tilde{E}_n(x, y)$ se construye mediante el uso de coeficientes de transformación. La trama de error de predicción localmente decodificada se añade a la trama de predicción $P_n(x, y)$ en el agregador 19 y la trama actual decodificada resultante $\tilde{I}_n(x, y)$ se almacena en la memoria de trama 17 para su uso posterior como la siguiente trama de referencia $R_{n+1}(x, y)$.

La corriente de información 2 que lleva información acerca de los vectores de movimiento se combina con información acerca del error de predicción en el multiplexor 16 y una corriente de información 3 que contiene normalmente por lo menos esos dos tipos de información es enviada al decodificador 20.

La operación del decodificador de video 20 correspondiente se describirá ahora. La memoria de trama 24 del decodificador 20 almacena una trama de referencia $R_n(x, y)$ previamente reconstruida. La trama de predicción $P_n(x, y)$ se construye en el bloque de predicción compensado en cuanto a movimiento 21 del decodificador 20 de acuerdo con la ecuación 5, mediante el uso de información de coeficiente de movimiento recibida y valores de píxeles de la trama de referencia previamente reconstruida $R_n(x, y)$. Los coeficientes de transformación transmitidos de la trama de error de predicción $E_n(x, y)$ se usan en el bloque de decodificación de error de predicción 22 para construir la trama

de error de predicción decodificada $\tilde{E}_n(x, y)$. Los píxeles de la trama actual decodificada $\tilde{I}_n(x, y)$ son después reconstruidos al agregar la trama de predicción $P_n(x, y)$ y la trama de error de predicción decodificado

$$\tilde{I}_n(x, y) = P_n(x, y) + \tilde{E}_n(x, y) = R_n [x + \tilde{\Delta x}(x, y), y + \tilde{\Delta y}(x, y)] + \tilde{E}_n(x, y). \quad (6)$$

Esta trama actual decodificada puede ser almacenada en la memoria de trama 24 como la siguiente trama de referencia $R_{n+1}(x,y)$.

En la descripción de codificación y decodificación compensada en cuanto a movimiento de video digital presentado anteriormente, el vector de movimiento $[\Delta x(x,y), \Delta y(x,y)]$ que describe el movimiento de un macrobloque en la trama actual con respecto a la trama de referencia $R_n(x,y)$ puede indicar a cualquiera de los píxeles en la trama de referencia. Esto significa que el movimiento entre tramas de una secuencia de video digital sólo puede ser representado a una resolución que se determina por los píxeles de imagen en la trama (denominada resolución de píxeles completa). Sin embargo, el movimiento real tiene precisión arbitraria y, por lo tanto, el sistema anteriormente descrito sólo puede proporcionar una modelación aproximada del movimiento entre tramas sucesivas de una secuencia de video digital. Normalmente, el modelaje de movimiento entre tramas de video con resolución de píxeles completa no es suficientemente exacta para permitir una reducción al mínimo eficiente de la información de error de predicción (PE) asociada con cada macrobloque/trama. Por lo tanto, para permitir un modelaje más exacto del movimiento real y para ayudar a reducir la cantidad de información de PE que debe ser transmitida desde el codificador al decodificador, muchos estándares de codificación de video, tales como H.263 (+)(++) y H.2611, permiten que los vectores de movimiento indiquen "entre" píxeles de imagen. En otras palabras, los vectores de movimiento pueden tener resolución de "subpíxeles". El permitir que los vectores de movimiento tengan resolución de subpíxeles se agrega a la complejidad de las operaciones de codificación y de decodificación que deben realizarse, por lo que es aún ventajoso limitar el grado de resolución espacial que pueda tener un vector de movimiento. Por lo tanto, los estándares de codificación de video, tales como aquellos anteriormente mencionados, normalmente sólo permiten vectores de movimiento que tienen resolución de píxel completo, medio píxel o un cuarto de píxel.

La estimación de movimiento con resolución de subpíxeles generalmente se realiza como un proceso de dos etapas, como se ilustra en la figura 5, para un esquema de codificación de video que permite que los vectores de movimiento tengan resolución de píxel completo o de medio píxel. En la primera etapa, un vector de movimiento que tiene resolución de píxel completo se determina usando cualquier esquema de estimación de movimiento apropiado, tal como el proceso de coincidencia de bloques descrito anteriormente. El vector de movimiento resultante, que tiene resolución de píxel completo se muestra en la figura 5.

En la segunda etapa, el vector de movimiento determinado en la primera etapa se refina para obtener la resolución de medio píxel deseada. En el ejemplo ilustrado en la figura 5, esto se hace al formar ocho bloques de búsqueda nuevos de 16 x 16 píxeles, la posición de la esquina superior izquierda de cada bloque es marcada con una X en la figura 5. Estas posiciones se indican como $[\Delta x + m/2, \Delta y + n/2]$, en donde m y n pueden adoptar los valores de -1, 0 y +1, pero no pueden ser cero al mismo tiempo. Como sólo se conocen los valores de píxeles de imagen original, los valores (por ejemplo valores de luminancia y/o crominancia) de los subpíxeles que residen en posiciones de medio píxel se deben estimar para cada uno de los ocho bloques de búsqueda nuevos, al usar alguna forma de esquema de interpolación.

Una vez que se han interpolado los valores de los subpíxeles a una resolución de medio píxel, cada uno de los ocho bloques de búsqueda se compara con el macrobloque cuyo vector de movimiento se busca. Como en el procedimiento de coincidencia de bloques realizado para determinar el vector de movimiento con resolución de píxel completo, el macrobloque se compara con cada uno de los ocho bloques de búsqueda de acuerdo con cierto criterio, por ejemplo, un SAD. Como resultado de las comparaciones, generalmente se obtendrá un valor de SAD mínimo. Según la naturaleza del movimiento en la secuencia de video, este valor mínimo puede corresponder a la posición especificada por el vector de movimiento original (que tiene resolución de píxel completo) o puede corresponder a una posición que tiene una resolución de medio píxel. Por lo tanto, es posible determinar si un vector de movimiento debe indicar a una posición de píxel completo o subpíxel y si una resolución de subpíxel es apropiada, para determinar el vector de movimiento de resolución de subpíxel correcto. También se debe apreciar que el esquema apenas descrito se puede extender a otras resoluciones de subpíxel (por ejemplo, una resolución de un cuarto de píxel) de una manera completamente análoga.

En la práctica, la estimación de un valor de subpíxel en la trama de referencia se realiza al interpolar el valor del subpíxel a partir de los valores de píxel circundantes. En general, la interpolación de un valor de subpíxel $F(x,y)$ situado en una posición no entera $(x, y) = (n + \Delta x, m + \Delta y)$, se puede formular como una operación bidimensional, representada matemáticamente como:

$$F(x, y) = \sum_{k=-K}^{K+1} \sum_{l=-L}^{L+1} f(k+K, l+L) F(n+k, m+l) \quad (7)$$

donde $f(k,l)$ son coeficientes de filtro y n y m se obtienen al truncar x e y , respectivamente, a valores enteros. Normalmente, los coeficientes de filtro dependen de los valores de x e y , y los filtros de interpolación generalmente se denominan "filtros separables", en cuyo caso el valor de subpíxel $F(x,y)$ se puede calcular como sigue:

$$F(x, y) = \sum_{k=-K}^{K-1} f(k+K) \sum_{l=-K}^{K-1} f(l+K) F(n+k, m+l) \quad (8)$$

Los vectores de movimiento se calculan en el codificador. Una vez que los coeficientes de movimiento correspondientes son transmitidos al decodificador, es un asunto directo interpolar los subpíxeles requeridos mediante el uso de un método de interpolación idéntico al usado en el codificador. De esta manera, una trama que sigue a una trama de referencia en la memoria de trama 24, puede ser reconstruido a partir de la trama de referencia y los vectores de movimiento.

La manera más simple de aplicar interpolación de valor de subpíxel en un codificador de video es interpolar cada valor de subpíxel cada vez que es necesario. Sin embargo, esta no es una solución eficiente en un codificador de video, ya que es probable que el mismo valor de subpíxel se requerirá varias veces y por lo tanto los cálculos para interpolar el mismo valor de subpíxel se realizará múltiples veces. Esto da por resultado un incremento innecesario de complejidad/carga computacional en el codificador.

Un enfoque alternativo, que limita la complejidad del codificador, es calcular previamente y almacenar todos los valores de subpíxel en una memoria asociada con el codificador. Esta solución se denomina interpolación "de antemano" de aquí en adelante en este documento. Aunque se limita la complejidad, la interpolación de antemano tiene el inconveniente de incrementar el uso de memoria en un gran margen. Por ejemplo, si la exactitud del vector de movimiento es un cuarto de píxel tanto en dimensión horizontal como vertical, el almacenamiento de valores de subpíxel precalculados para una imagen completa da por resultado un uso de memoria que es 16 veces la requerida para almacenar imagen no interpolada original. Además, implica el cálculo de algunos subpíxeles que podrían no ser realmente requeridos en el cálculo de vectores de movimiento en el codificador. La interpolación de antemano también es particularmente ineficiente en un decodificador de video, ya que la mayoría de los valores de subpíxel calculados previamente nunca serán requeridos por el decodificador. Por lo tanto, es ventajoso no usar cálculo previo en el decodificador.

La denominada interpolación "bajo demanda" se puede usar para reducir requerimientos de memoria en el codificador. Por ejemplo, si la precisión de píxeles deseada es una resolución de un cuarto de píxel, sólo subpíxeles a una resolución de unidad de un medio se interpolarán de antemano para la trama completa y se almacenarán en la memoria. Los valores de subpíxeles de resolución de un cuarto de píxel sólo se calculan durante el proceso de estimación/compensación de movimiento según y cuando se requiera. En este caso, el uso de memoria es sólo 4 veces el requerido para almacenar la imagen original, no interpolada.

Debe indicarse que cuando se usa la interpolación de antemano, el proceso de interpolación constituye sólo una pequeña fracción de la complejidad/carga computacional total del codificador, ya que cada píxel es interpolado apenas una vez. Por lo tanto, en el codificador, la complejidad del proceso de interpolación mismo no es muy crítica cuando se usa la interpolación de valor de subpíxeles de antemano. Por otra parte, la interpolación bajo demanda posee una carga computacional mucho mayor en el codificador, ya que los subpíxeles pueden ser interpolados muchas veces. Por lo tanto, la complejidad de proceso de interpolación, que se puede considerar en términos del número de operaciones computacionales o ciclos operacionales que se deban realizar para interpolar los valores de subpíxeles, se vuelven una consideración importante.

En el decodificador, los mismos valores de subpíxeles se usan unas cuantas veces cuando mucho y algunos no son necesarios en absoluto. Por lo tanto en el decodificador es ventajoso no usar interpolación de antemano en absoluto, es decir, es ventajoso no calcular previamente cualesquiera valores de subpíxeles.

Dos esquemas de interpolación han sido desarrollados como parte del trabajo en curso en el Sector de Estandarización de Telecomunicaciones de ITU-T, Grupo de Estudio 16, Grupo de Expertos de Codificación de Video (VCEG), Preguntas 6 y 15. Estos enfoques se proponen para incorporar en la recomendación H.26L de ITU-T y se han implementado en valores de prueba (TML) para los propósitos de evaluación y desarrollo posterior. El modelo de prueba correspondiente a la Pregunta 15 se refiere como Modelo de Prueba 5 (TML5), el que resulta de la Pregunta 6 se conoce como Modelo de Prueba 6 (TML6). Los esquemas de interpolación propuestos en TML5 y TML6 no se describirán.

En toda la descripción del esquema de interpolación de valores de subpíxel usado en el modelo de prueba TML5, se hará referencia a la figura 12a, que define una notación para describir posiciones de píxeles y subpíxeles específicas para TML5. Una notación separada, definida en la figura 13a, se usará en la discusión del esquema de interpolación de valor de subpíxeles usado en TML6. Una notación adicional, ilustrada en la figura 14a, se usará posteriormente en el texto en conexión con el método de interpolación de valor de subpíxeles de acuerdo con la invención. Se apreciará que las tres notaciones diferentes usadas en el texto pretenden ayudar a entender cada método de interpolación y ayudar a distinguir diferencias entre los mismos. Sin embargo, en todas las tres figuras, la letra A se usa para indicar píxeles de imagen original (resolución de píxeles completos). De manera más específica, la letra A representa la posición de píxeles en los datos de imagen que representan una imagen de una secuencia de video, los valores de píxeles de los píxeles A son recibidos como trama actual $I_n(x,y)$ de una fuente de video, o

reconstruidos y almacenados como una trama de referencia $R_n(x,y)$ en la Memoria de Trama 17, 24 del codificador 10 o el decodificador 20. Todas las demás letras representan posiciones de subpíxeles, los valores de los subpíxeles situados en las posiciones de subpíxeles se obtienen por interpolación.

5 Algunos otros términos también se usarán de una manera consistente en todo el texto para identificar posiciones particulares de píxeles y subpíxeles. Estas son las siguientes:

10 El término "posición horizontal de unidad" se usa para describir la posición de cualquier subpíxel que se construye en una columna de los datos de imagen original. Los subpíxeles c y e en las figuras 12a y 13a, así como los subpíxeles b y e en la figura 14a tienen posiciones horizontales de unidad.

15 El término "posición vertical de unidad" se usa para describir cualquier subpíxel que es construido en una fila de los datos de imagen original. Los subpíxeles b y d en las figuras 12a y 13a así como los subpíxeles b y d en la figura 14a tienen posiciones verticales de unidad.

Por definición, los píxeles A tienen posiciones horizontales de unidad y verticales de unidad.

20 El término "posición horizontal de medio" se usa para describir la posición de cualquier subpíxel que es construido en una columna que está a una resolución de medio píxel. Los subpíxeles b, c, y e mostrados en las figuras 12a y 13a caen en esta categoría, y también los subpíxeles b, c y f de la figura 14a. De una manera similar el término "posición vertical de medio" se usa para describir la posición de cualquier subpíxel que es construido en una fila que está a una resolución de medio píxel, tal como los subpíxeles c y d en las figuras 12a y 13a, así como los subpíxeles b, c y g en la figura 14a.

25 Además, el término "posición horizontal de un cuarto" se refiere a cualquier subpíxel que es construido en una columna que está a una resolución de un cuarto de píxel, tal como subpíxeles e y d en la figura 12a, subpíxeles d y g en la figura 13a y subpíxeles d, g y h en la figura 14a. De manera análoga, el término "posición vertical de un cuarto" se refiere a subpíxeles que están construidos en una fila que está a una resolución de un cuarto de píxel. En la figura 12a, los subpíxeles e y f caen en esta categoría y también lo son los subpíxeles e, f y g en la figura 13a y los subpíxeles e, f y h en la figura 14a.

30 La definición de cada uno de los términos anteriormente descritos se muestra mediante "sobres" dibujados en las figuras correspondientes.

35 Debe indicarse además que a menudo es conveniente indicar un píxel particular con una referencia bidimensional. En este caso, la referencia bidimensional apropiada se puede obtener al examinar la intersección de los sobres en las figuras 12a, 13a y 14a. Al aplicar este principio, el píxel d en la figura 12a, por ejemplo, tiene una posición horizontal de medio y vertical de medio y el subpíxel e tiene una posición horizontal de unidad y vertical de un cuarto. Además y para facilidad de referencia, los subpíxeles que residen en posiciones horizontales de mitad de unidad y verticales de unidad, posiciones horizontales de unidad y verticales de mitad de unidad así como posiciones horizontales de mitad de unidad y verticales de mitad de unidad, se referirán como subpíxeles de resolución de 1/2. Los subpíxeles que residen en cualquier posición horizontal de un cuarto de unidad y/o posición vertical de un cuarto de unidad se referirán como subpíxeles de resolución de 1/4.

45 Debe indicarse también que las descripciones de los dos modelos de prueba y en la descripción detallada de la invención misma, se supondrá que los píxeles tienen un valor mínimo de cero y un valor máximo de $2^n - 1$ donde n es el número de bits reservados para un valor de píxel. El número de bits es normalmente de 8. Después de que un subpíxel ha sido interpolado, si el valor del subpíxel interpolado excede el valor de $2^n - 1$ se restringe al intervalo de $[0, 2^n - 1]$, es decir, valores menores que el valor permitido mínimo serán el valor mínimo (0) y valores mayores que el máximo serán el valor máximo ($2^n - 1$). Esta operación se denomina de recorte.

50 El esquema de interpolación de valor de subpíxel de acuerdo con TML5 se describirá ahora con detalle con referencia a las figuras 12a, 12b y 12c.

55 1. El valor para el subpíxel en una posición horizontal de media unidad y posición vertical de unidad, es decir subpíxel (b) de resolución 1 en la figura 12a, se calcula mediante el uso de un filtro de seis ramas. El filtro interpola un valor para subpíxel b de resolución de .1 con los valores de los 6 píxeles (A_1 a A_6) situados en una fila en posiciones horizontales de unidad y posiciones verticales de unidad simétricamente alrededor de b, como se muestra en la figura 12b, de acuerdo con la fórmula $b = (A_1 - 5A_2 + 20A_3 + 20A_4 - 5A_5 + A_6 + 16)/32$. El operador / indica división con truncamiento. El resultado es recortado para quedar en el intervalo $[0, 2^n - 1]$.

60 2. Los valores para subpíxeles de resolución de marcados con c se calculan mediante el uso del mismo filtro de seis ramas que se usa en la etapa 1 y los seis píxeles o subpíxeles más cercanos (A o b) en la dirección vertical. Con referencia ahora a la figura 12c, el filtro interpola un valor para el subpíxel c de resolución de $\frac{1}{2}$ ubicada en la posición horizontal de unidad y la posición vertical de medio con base en los

valores de los 6 píxeles (A_1 a A_6) situados en una columna en posiciones horizontales de unidad y posiciones verticales de unidad simétricamente alrededor de c , de acuerdo con la fórmula $c = (A_1 - 5A_2 + 20A_3 + 20A_4 - 5A_5 + A_6 + 16)/32$. De manera similar un valor para el subpíxel c de resolución de en la posición horizontal de medio y vertical de medio se calcula de acuerdo con $c = (b_1 - 5b_2 + 20b_3 + 20b_4 - 5b_5 + b_6 + 16)/32$. Nuevamente, el operador / indica división con truncamiento. Los valores calculados para los subpíxeles c son además recortados para quedar en el intervalo $[0, 2^n - 1]$.

En este punto en el proceso de interpolación los valores de todos los subpíxeles de resolución $1/2$ se han calculado y el proceso procede al cálculo de valores de subpíxeles de resolución $1/4$.

3. Los valores para los subpíxeles de resolución $1/4$ marcados con la letra d se calculan mediante el uso de interpolación lineal y los valores de los píxeles más cercanos / o subpíxeles de resolución $1/2$ en la dirección horizontal. De manera más específica, los subpíxeles de resolución $1/4$ en las posiciones horizontales de un cuarto y verticales de unidad, se calculan al tomar el promedio del píxel inmediatamente vecino en una posición de unidad y vertical de unidad (píxel A) y el subpíxel de resolución $1/2$ inmediatamente vecino en la posición horizontal de medio y horizontal de unidad (subpíxel b), es decir, de acuerdo con $d = (A + b)/2$. Los valores para subpíxeles d de resolución $1/4$ ubicados en posiciones horizontales de un cuarto y verticales de medio, se calculan al tomar un medio de los subpíxeles c de resolución $1/2$ inmediatamente vecinos que están en una posición horizontal de unidad y vertical, media y en posiciones horizontales de medio y verticales de medio respectivamente, es decir, de acuerdo con $d = (c_1 + c_2)/2$. Nuevamente, el operador / indica división con truncamiento.

4. Los valores para los subpíxeles de resolución de $1/4$ marcados con la letra e se calculan usando interpolación lineal y los valores de los subpíxeles más cercanos y / o subpíxeles de resolución de $1/2$ en la dirección vertical. En

particular, los subpíxeles e de resolución $1/4$ en posiciones horizontales de unidad y verticales de un cuarto se calculan al tomar el promedio del píxel inmediatamente vecino en la posición horizontal de unidad y posición vertical de unidad (píxel A) y el subpíxel inmediatamente vecino en la posición horizontal de unidad y vertical de medio (subpíxel c) de acuerdo con $e = (A + c)/2$. Los subpíxeles e_3 de resolución $1/4$ en las posiciones horizontales de medio y verticales de un cuarto se calculan al tomar el promedio del subpíxel inmediatamente vecino y en la posición horizontal de medio y vertical de unidad (subpíxel b) y el subpíxel inmediatamente vecino en la posición horizontal de medio y vertical de medio (subpíxel c), de acuerdo con $e = (b + c)/2$. Además, los subpíxeles e de resolución de $1/4$ en posiciones horizontales de un cuarto y verticales de un cuarto se calculan al tomar el promedio de los subpíxeles inmediatamente vecinos en la posición horizontal de un cuarto y vertical de unidad y el subpíxel correspondiente en la posición horizontal de un cuarto y vertical de medio (subpíxeles d), de acuerdo con $e = (d_1 + d_2)/2$. Una vez más, el operador / indica división con truncamiento.

5. El valor para el subpíxel f de resolución de $1/4$ es interpolado al promediar los valores de los 4 valores de píxeles más cercanos en las posiciones horizontales y verticales de unidad, de acuerdo con $f = (A_1 + A_2 + A_3 + A_4 + 2)/4$, donde los píxeles A_1, A_2, A_3 y A_4 son los cuatro píxeles originales más cercanos.

Un inconveniente de TML5 es que el decodificador es computacionalmente complejo. Esto resulta del hecho de que TML5 utiliza un enfoque en el cual la interpolación de valores de subpíxeles de resolución de $1/4$ depende de la interpolación de valores de subpíxeles de resolución de $1/2$. Esto significa que para interpolar los valores de los subpíxeles de resolución $1/4$, los valores de los subpíxeles de resolución $1/2$ de los cuales se determinaron se deben calcular primero. Además, dado que los valores de algunos de los subpíxeles de resolución $1/4$ dependen de los valores interpolados obtenidos por otros subpíxeles de resolución de $1/4$, el truncamiento de los valores de subpíxeles de resolución $1/4$ tiene un efecto negativo sobre la precisión de algunos de los valores de subpíxeles de resolución de $1/4$. Específicamente, los valores de subpíxeles de resolución de $1/4$ son menos precisos de lo que sería si se calculara a partir de los valores que no han sido truncados y recortados. Otro inconveniente de TML5 es que es necesario almacenar los valores de los subpíxeles de resolución de $1/2$ para interpolar los valores de subpíxeles de resolución de $1/4$. Por lo tanto, se requiere un exceso de memoria para almacenar un resultado que no es finalmente requerido.

El esquema de interpolación de valor de subpíxel de acuerdo con TML6, referido aquí como interpolación directa, se describirá ahora. En el codificador del método de interpolación de acuerdo con TML6 funciona como el método de interpolación de TML5 descrito anteriormente, excepto que la precisión máxima es retenida en todas partes. Esto se logra al usar valores intermedios que no son redondeados ni recortados. Una descripción paso a paso del método de interpolación de acuerdo con TML6 como se aplica en el codificador se da más adelante con referencia a las figuras 13a, 13b y 13c.

1. El valor para el subpíxel en la posición horizontal y media unidad y vertical de unidad, es decir, el subpíxel b de resolución M en la figura 13a, se obtiene al calcular primero un valor calculado b mediante el uso de un filtro de seis ramas. El filtro calcula b , con base en los valores de los 6 píxeles (A_1 a A_6) situados en una fila en posiciones horizontales de unidad y posiciones verticales de unidad simétricamente alrededor de b , como se muestra en la figura 13b, de acuerdo con la fórmula $b = (A_1 - 5A_2 + 20A_3 + 20A_4 - 5A_5 + A_6)$. El valor final de b

se calcula entonces como $b(b + 16)/32$ y se recorta para estar en el intervalo $[0, 2^n - 1]$. Igual que antes, el operador / indica división con truncamiento.

2. Los valores para los subpíxeles de resolución marcados con la letra c se obtienen de calcular primero los valores intermedios c. Con referencia a la figura 13c, un valor intermedio c para el subpíxel c de resolución 1/2 ubicado en la posición horizontal de unidad y vertical de medio se calcula con base en los valores de 6 píxeles (A_1 a A_6) situados en una columna en posiciones horizontales de unidad y posiciones verticales de unidad simétricamente alrededor de c, de acuerdo con la fórmula $c = (A_1 - 5A_2 + 20A_3 + 20A_4 - 5A_5 + A_6)$. El valor final para el subpíxel c de resolución de 1/2 ubicado en una posición horizontal de unidad y vertical de medio se calcula de acuerdo con $c = (c + 16)/32$. De manera similar, un valor intermedio c para el subpíxel c de resolución de en la posición horizontal de medio y vertical de medio de acuerdo con $c = (b_1 - 5b_2 + 20b_3 + 20b_4 - 5b_5 + b_6)$. Un valor final para este subpíxel c de resolución de 1/2 se calcula después de acuerdo con $(c + 512)/1024$. Nuevamente, el operador / indica división con truncamiento y los valores calculados para subpíxeles c de resolución 1/2 son después recortados para estar en el intervalo de $[0, 2^n - 1]$.

3. Los valores para subpíxeles de resolución 1/4 marcados con la letra d se calculan como sigue. Los valores para subpíxeles d de resolución de 1/4 ubicados en posiciones horizontales de un cuarto y verticales de unidad, se calculan a partir del valor del píxel inmediatamente vecino en la posición horizontal de unidad y posición vertical de unidad (píxel A) y el valor b intermedio calculado en la etapa (1) para el subpíxel de resolución de 1/2 inmediatamente vecino en la posición horizontal de medio y vertical de unidad (subpíxel b de resolución 1/2), de acuerdo con $d = (32A + b + 32)/64$. Los valores para subpíxeles d de resolución de 1/4 ubicado en posiciones horizontales de un cuarto y verticales de medio, son interpolados mediante el uso de valores intermedios c calculados para los subpíxeles c de resolución de 1/2 inmediatamente vecinos que están en la posición horizontal de unidad y vertical de medio y las posiciones horizontales de medio y verticales de medio respectivamente, de acuerdo con $d = (32c_1 + c_2 + 1024)/2048$. Nuevamente el operador / indica división con truncamiento y los valores de subpíxel d de resolución de 1/4 y finalmente obtenidos son recortados para estar en el intervalo $[0, 2^n - 1]$.

4. Los valores para los subpíxeles de resolución de 1/4 marcado con la letra e se calculan como sigue. Los valores para subpíxeles e de resolución 1/4 ubicados en posiciones horizontales de unidad y verticales de un cuarto se calculan a partir del valor del píxel inmediatamente vecino en la posición horizontal de unidad y vertical de unidad (píxel A) y el valor c intermedio calculado en la etapa (2) para el subpíxel de resolución de 1/4 inmediatamente vecino en la posición horizontal de unidad y vertical de unidad, de acuerdo con $e = (32A + c + 32)/64$. Los valores para subpíxeles e de resolución de 1/4 ubicados en posiciones horizontales de medio y verticales de un cuarto se calculan a partir del valor b intermedio calculado en la etapa (1) para el subpíxel de resolución de 1/2 inmediatamente vecino en la posición horizontal de medio y vertical de unidad y el valor c intermedio calculado en la etapa (2) para el subpíxel de resolución de 1/2 inmediatamente vecino en posición horizontal de medio y vertical de medio, de acuerdo con $e = (32b + c + 1024)/2048$. Una vez más, el operador / indica división con truncamiento y los valores e de subpíxeles de resolución de 1/4 finalmente obtenidos son recortados para estar en el intervalo $[0, 2^n - 1]$.

5. Los valores para subpíxeles de resolución de 1/4 marcados con la letra g son calculados mediante el uso del valor del siguiente píxel A original más cercano y los valores intermedios de los tres subpíxeles de resolución de 1/2 vecinos más cercanos, de acuerdo con $g = (1024A + 32b + 32c_1 + c_2 + 2048)/4096$. Igual que antes, el operador / indica división con truncamiento y los valores g de subpíxel de resolución de 1/4 son recortados para estar en el intervalo $[0, 2^n - 1]$.

6. El valor para el subpíxel f de resolución de 1/4 es interpolado al promediar los valores de los 4 píxeles más cercanos en posiciones horizontales y verticales de unidad de acuerdo con $f = (A_1 + A_2 + A_3 + A_4 + 2)/4$, donde las posiciones de los píxeles A_1, A_2, A_3 y A_4 son los cuatro píxeles originales más cercanos.

En el decodificador, de valores de subpíxel se pueden obtener directamente al aplicar filtros de seis ramas en direcciones horizontal y vertical. En el caso de resolución de subpíxel de 1/4, con referencia a la figura 13a, los coeficientes de filtro aplicados a los píxeles y subpíxeles en la posición vertical de unidad son $[0, 0, 64, 0, 0, 0]$ para un conjunto de seis píxeles A, $[1, -5, 52, 20, -5, 1]$ para un conjunto de seis subpíxeles d, $[2, -10, 40, 40, -10, 2]$ para un conjunto de seis subpíxeles b, y $[1, -5, 20, 52, -5, 1]$ para un conjunto de seis subpíxeles d. Estos coeficientes de filtro se aplican a conjuntos de píxeles o subpíxeles respectivos en la misma fila que los valores de subpíxeles que son interpolados.

Después de aplicar los filtros en las direcciones 25 horizontal y vertical, el valor interpolado c es normalizado de acuerdo con $c = (c + 2048)/4096$ y recortado para estar en el intervalo $[0, 2^n - 1]$. Cuando un vector de movimiento indica a una posición de píxel integral ya sea en la dirección horizontal o vertical, muchos coeficientes cero se usan. En una implementación práctica de TML6, se usan diferentes ramas en el software que son optimizadas para los diferentes casos de subpíxeles por lo que no hay multiplicaciones por coeficientes cero.

Debe indicarse que en TML6, los valores de subpíxel de resolución de 1/4 se obtienen directamente al usar los valores intermedios referidos anteriormente y no se derivan de valores redondeados y recortados para subpíxeles de

resolución de 1/2. Por lo tanto, en la obtención de valores de subpíxeles de resolución de 1/4, no es necesario calcular valores finales para ninguno de los subpíxeles de resolución de 1/2. Específicamente, no es necesario llevar a cabo operaciones de truncamiento y recorte asociadas con el cálculo de valores finales para los subpíxeles de resolución de 1/2. Tampoco no es necesario tener almacenados valores finales para subpíxeles de resolución de 1/2 para usarse en el cálculo de los valores de subpíxeles de resolución de 1/4. Por lo tanto, TML6 es computacionalmente menos complejo que TML5, ya que se requieren menos operaciones de truncamiento y recorte. Sin embargo, un inconveniente de TML6 es que se requiere alta aritmética de precisión en el codificador y en el decodificador. La interpolación de alta precisión requiere más área de silicio en ASICs y requiere más cálculos en algunas CPUs. Además, la implementación de interpolación directa como se especifica en TML6 de una manera bajo demanda tiene un requerimiento de memoria alto. Este es un factor importante, particularmente en dispositivos integrados.

En vista de la descripción anteriormente presentada, se debe apreciar que debido a los diferentes requerimientos del codificador y decodificador de video con respecto a interpolación de subpíxeles, existe un problema significativo en desarrollar un método de interpolación de valores de subpíxeles capaz de proveer rendimiento satisfactorio tanto en el codificador como en el decodificador. Además, ninguno de los modelos de prueba actuales (TML5, TML6) descritos anteriormente pueden proveer una solución que sea óptima para la aplicación tanto en el codificador como en el decodificador.

La patente US 5.521.642 divulgar un sistema de descodificación simplificada para proporcionar una trama de imagen reducida a un receptor de televisión de alta definición con una pantalla de tamaño pequeño a través del uso de coeficientes de transformación de CC. El sistema de decodificación decodifica selectivamente y cuantifica de manera inversa coeficientes de transformación de CC para producir un conjunto de datos de diferencia, cada uno de los cuales representa un promedio de las diferencias de píxeles entre un bloque de píxeles bidimensionales de una trama actual y un bloque correspondiente de su trama anterior. Cada uno de los vectores de movimiento en dos dimensiones también se decodifica y modifica para derivar los datos de píxeles de la trama de imagen reducida anterior. Los datos de píxeles derivados y el valor medio de diferencia de píxel se combinan sucesivamente para generar la trama reducida.

La solicitud de patente europea publicada EP 0 576 290 se refiere a métodos de codificación y decodificación de señal de imagen que eliminan una situación tal que, cuando una imagen codificada de un sistema de televisión de alta definición se hace más fina a la mitad en cada una de las direcciones vertical y horizontal y se muestran en un receptor de televisión de un sistema de definición inferior, la imagen mostrada no exhibe un movimiento suave debido a la pérdida de una estructura entrelazada. En un codificador, los datos de elementos de imagen son procesados mediante procesamiento DCT para obtener 8 x 8 datos de coeficiente, y los datos de coeficiente se transmiten. En un decodificador, de los 8 x 8 datos de coeficiente, sólo 4 x 4 datos de coeficiente en la esquina superior izquierda se muestrean y se procesan mediante procesamiento IDCT para obtener datos de elementos de imagen originales. En el procesamiento IDCT, aquellos de los 4 x 4 datos de coeficiente que pertenecen a la cuarta fila se sustituyen por los de los datos de coeficientes de la octava fila de los 8 x 8 datos de coeficiente.

Sumario de la Invención

De acuerdo con un primer aspecto de la invención, se proporciona un método de interpolación en codificación de video en el cual una imagen que comprende píxeles dispuestos en filas y columnas y representados por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, es interpolada para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, las posiciones horizontales y verticales fraccionadas son definidas de acuerdo con $1/2^x$, en donde x es un entero positivo que tiene un valor máximo N, comprendiendo el método:

a) cuando se requieren los valores para los subpíxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y en posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad, la interpolación de estos valores es directamente con el uso de sumas de píxeles ponderadas que residen en posiciones horizontales de unidad y verticales de unidad;

b) cuando se requieren los valores para los subpíxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de $1/2^{N-1}$ de unidad, la interpolación de esos valores es directamente con el uso de una elección de una primera suma ponderada de valores para subpíxeles que residen en las posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y una segunda suma ponderada de valores para subpíxeles que residen en posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad, la primera y segunda sumas ponderadas de valores se calculan de acuerdo con la etapa (a); y

c) cuando se requiere un valor para un subpíxel situado en una posición horizontal $1/2^N$ de unidad y vertical de $1/2^N$ de unidad, la interpolación de ese valor al tomar un promedio de grado de valor de un primer subpíxel situado en la posición horizontal de $1/2^{N-m}$ de unidad y vertical de $1/2^{N-n}$ de unidad y el valor de un segundo subpíxel ubicado en una posición horizontal de $1/2^{N-p}$ de unidad y vertical de $1/2^{N-q}$ de unidad, las variables m, n, p y q toman valores enteros en el intervalo de 1 a N, de tal manera que el primer y segundo subpíxeles o píxeles están ubicados diagonalmente con respecto al subpíxel en la posición horizontal de $1/2^N$ de unidad y

vertical de $1/2^N$.

5 Preferiblemente, una primera y segunda ponderación se usan en el promedio ponderado referido en (c), las magnitudes relativas de las ponderaciones son inversamente proporcionales a la proximidad (diagonal en línea recta) del primer y segundo subpíxel o píxel para el subpíxel en la posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$.

10 En una situación donde el primer y segundo subpíxel o píxel están ubicados simétricamente con respecto al (equidistante del) subpíxel en la posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$, la primera y segunda ponderaciones pueden tener valores iguales.

15 La primera suma ponderada de valores para subpíxeles que residen en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad en la etapa b) se puede usar cuando se requiere un subpíxel en la posición horizontal de $1/2^{N-1}$ de unidad y vertical de $1/2^N$ de unidad.

La segunda suma ponderada de valores de subpíxeles residen en las posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad en la etapa b) se pueden usar cuando se requiere un subpíxel en la posición horizontal de $1/2^{N-1}$ de unidad y vertical de $1/2^{N-1}$ de unidad.

20 En una realización, cuando se requieren los valores para subpíxeles en las posiciones horizontales de $1/2^N$ de unidad y verticales de unidad, y las posiciones horizontales de $1/2^N$ y verticales de $1/2^{N-1}$, esos valores son interpolados al sacar el promedio de los valores de un primer píxel o subpíxel ubicado en una posición vertical correspondiente a la del subpíxel que se calcula y la posición horizontal de unidad y un segundo píxel o subpíxel ubicado en una posición vertical correspondiente a la del píxel que se calcula y la posición horizontal de $1/2^{N-1}$ de unidad.

30 Cuando se requieren los valores para subpíxeles en las posiciones horizontales de unidad y verticales de $1/2^N$ de unidad, y las posiciones horizontal de $1/2^{N-1}$ de unidad y vertical de $1/2^N$ de unidad, pueden ser interpolados al sacar el promedio de los valores de un primer píxel o subpíxel ubicado en una posición horizontal correspondiente a la del subpíxel que se calcula y una posición vertical de unidad y un segundo píxel o subpíxel ubicado en una posición horizontal correspondiente a la del subpíxel que se calcula y la posición vertical de $1/2^{N-1}$ de unidad.

35 Los valores para subpíxeles en las posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^N$ de unidad pueden ser interpoladas al sacar el promedio de los valores de un píxel ubicado en una posición horizontal de unidad y vertical de unidad, y un píxel ubicado en una posición horizontal de $1/2^{N-1}$ de unidad y vertical de $1/2^{N-1}$ de unidad.

40 Los valores para subpíxeles en posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^N$ de unidad se pueden interpolar sacando el promedio de valores de un subpíxel ubicado en una posición horizontal de $1/2^{N-1}$ de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de unidad y vertical de $1/2^{N-1}$ de unidad.

45 Los valores para la mitad de subpíxeles en las posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^N$ de unidad pueden ser interpolados sacando el promedio de un primer par de valores de un subpíxel ubicado en una posición horizontal de $1/2^{N-1}$ de unidad y vertical de unidad y un subpíxel ubicado en una posición horizontal de unidad y vertical de $1/2^{N-1}$ de unidad y valores para la otra mitad de los subpíxeles en posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^N$ de unidad se interpolan al sacar el promedio de un segundo par de valores de un píxel ubicado en una posición horizontal de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de $1/2^{N-1}$ de unidad y vertical de $1/2^{N-1}$ de unidad.

50 Los valores para subpíxeles en posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^N$ de unidad son alternativamente interpolados para un subpíxel al sacar el promedio de un primer par de valores de un subpíxel ubicado en una posición horizontal de $1/2^N$ de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de unidad y vertical de $1/2^{N-1}$ de unidad y los valores para un subpíxel vecino al sacar el promedio de un segundo par de valores de un píxel ubicado en la posición horizontal de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de $1/2^{N-1}$ de unidad y vertical de $1/2^{N-1}$ de unidad.

55 Los subpíxeles de posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^N$ de unidad pueden ser alternativamente interpolados en una dirección horizontal.

60 Cuando se requieren los valores para algunos subpíxeles en posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^N$ de unidad, esos valores pueden ser interpolados al sacar el promedio de una pluralidad de píxeles vecinos más cercanos.

65 Por lo menos una de la etapa a) y de la etapa b) que interpolan valores de subpíxeles directamente al usar sumas ponderadas puede implicar el cálculo de un valor intermedio para los valores de subpíxeles que tienen un intervalo dinámico mayor que el intervalo dinámico especificado.

El valor intermedio para un subpíxel que tiene resolución de subpíxel $1/2^{N-1}$ se puede usar en el cálculo de un valor de subpíxel que tiene una resolución de subpíxel $1/2^N$.

De acuerdo con un segundo aspecto de la invención, se proporciona un método de interpolación en codificación de video en el cual una imagen que comprende píxeles dispuestos en filas y columnas y representada por valores que tienen un intervalo dinámico especificado, los píxeles en las filas que residen en posiciones horizontales de unidad y los píxeles en las columnas que residen en posiciones verticales de unidad, se interpola para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, comprendiendo el método:

- a) cuando los valores para subpíxeles en posiciones horizontales de media unidad y verticales de unidad, y en posiciones horizontales de unidad y verticales de media unidad se requieren, la interpolación de esos valores usa directamente sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;
- b) cuando los valores para subpíxeles en posiciones horizontales de media unidad y verticales de media unidad se requieren, la interpolación de esos valores directamente usa una suma ponderada de valores para subpíxeles que residen en posiciones horizontales de media unidad y verticales de unidad calculadas de acuerdo con la etapa a); y
- c) cuando los valores para subpíxeles en posiciones horizontales de un cuarto de unidad y verticales de un cuarto de unidad se requieren, se interpolan esos valores al sacar el promedio de por lo menos un par de un primer par de valores de un subpíxel ubicado en una posición horizontal de media unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de unidad y vertical de media unidad y un segundo par de valores de un píxel ubicado en una posición horizontal de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de media unidad y vertical de media unidad.

De acuerdo con un tercer aspecto de la invención, se proporciona un método de interpolación en codificación de video en el cual una imagen que comprende píxeles dispuestos en filas y columnas y representado por valores que tienen un intervalo dinámico especificado, los píxeles en las filas que residen en posiciones horizontales de unidad y los píxeles en las columnas que residen en posiciones verticales de unidad, se interpola para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, las posiciones horizontales y verticales fraccionadas se definen de acuerdo con $1/2^x$ donde x es un entero positivo que tiene un valor máximo N, comprendiendo el método:

- a) cuando los valores para los subpíxeles en las posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad, y las posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad se requieren, se interpolan esos valores directamente al usar sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;
- b) cuando un valor para un subpíxel en una posición horizontal de subpíxel y vertical de subpíxel se requiere, se interpola ese valor directamente al usar una elección de una primera suma de valores ponderada para subpíxeles ubicados en una posición vertical correspondiente a la del subpíxel que se calcula y una segunda suma de valores ponderada de subpíxeles ubicados en un lugar horizontal correspondiente al del subpíxel que se calcula.

Los subpíxeles usados en la primera suma ponderada pueden ser subpíxeles que residen en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y la primera suma ponderada se puede usar para interpolar un valor para un subpíxel en la posición horizontal de $1/2^{N-1}$ de unidad y vertical de $1/2^N$ de unidad.

Los subpíxeles usados en la segunda suma ponderada pueden ser subpíxeles que residen en posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad y la segunda suma ponderada se puede usar para interpolar un valor para un subpíxel en posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^{N-1}$ de unidad.

Cuando los valores para subpíxeles en las posiciones horizontales de $1/2^N$ de unidad y verticales de $1/2^N$ de unidad se requieren, se pueden interpolar al sacar el promedio de por lo menos un par de un primer par de valores de un subpíxel ubicado- en una posición horizontal de $1/2^{N-1}$ de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de unidad y vertical de $1/2^{N-1}$ de unidad y un segundo par de valores de un píxel ubicado en una posición horizontal de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de $1/2^{N-1}$ de unidad y vertical de $1/2^{N-1}$ de unidad.

Anteriormente, N puede ser igual a un entero seleccionado de una lista que consiste de valores 2, 3, y 4.

Los subpíxeles en la posición horizontal de un cuarto de unidad se han de interpretar como subpíxeles que tienen como su vecino más cercano a la izquierda un píxel en una posición horizontal de unidad y su vecino más cercano a la derecha un subpíxel en la posición horizontal de media unidad así como subpíxeles que tienen como su vecino más cercano a la izquierda un subpíxel en una posición horizontal de unidad y como su vecino más cercano a la derecha un píxel en una posición horizontal de unidad. De manera correspondiente, los subpíxeles en la posición de un cuarto de unidad se han de interpretar como subpíxeles que tienen como vecino más cercano superior un píxel en una posición vertical de unidad y como su vecino más cercano inferior un subpíxel en una posición vertical de

media unidad así como subpíxeles que tienen como su vecino más cercano superior un subpíxel en una posición vertical de media unidad y como su vecino más cercano inferior un píxel en una posición vertical de unidad.

El término intervalo dinámico se refiere al intervalo de valores que puede tomar los valores de subpíxel y las sumas ponderadas.

Al cambiar preferiblemente el intervalo dinámico, ya sea al extenderlo o reducirlo, se cambia el número de bits que se usa para representar el intervalo dinámico.

En una realización de la invención, el método se aplica a una imagen que es subdividida en un número de bloques de imagen. Preferiblemente, cada bloque de imagen comprende cuatro esquinas, cada esquina es definida por un píxel ubicado en una posición horizontal de unidad y vertical de unidad. Preferiblemente, el método se aplica a cada bloque de imagen a medida que el bloque queda disponible para interpolación de valor de subpíxel. Alternativamente, la interpolación de valor de subpíxel de acuerdo con el método de la invención se realiza una vez que todos los bloques de imagen de una imagen se han vuelto disponibles para interpolación de valor de subpíxel.

Preferiblemente, el método se usa en codificación de video. Preferiblemente, el método se usa en decodificación de video.

En una realización de la invención, cuando se usa en codificación, el método se lleva a cabo como interpolación de antemano, en la cual los valores para todos los píxeles en posiciones de media unidad y valores para todos los píxeles en posiciones de cuarto de unidad se calculan y se almacenan antes de usarse posteriormente en la determinación en un trama de predicción durante la codificación predictiva de movimiento. En realizaciones alternativas, el método se lleva a cabo a una combinación de interpolación de antemano y bajo demanda. En este caso, una cierta proporción o categoría de valores de subpíxeles se calcula y se almacena antes de ser usada en la determinación de una trama de predicción y algunos otros valores de subpíxeles se calculan solo cuando se requieren durante la codificación de predicción de movimiento.

Preferiblemente, cuando el método se usa en decodificación, los subpíxeles solo son interpolados cuando su necesidad es indicada por un vector de movimiento.

De acuerdo con un cuarto aspecto de la invención, se proporciona un codificador de video para codificar una imagen que comprende píxeles dispuestos en filas y columnas y representados por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, el codificador de video comprende un interpolador adaptado para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, las posiciones horizontales y verticales fraccionadas se definen de acuerdo con $1/2^x$, en donde x es un entero positivo que tiene un valor máximo N , el interpolador está adaptado para:

- a) interpolar valores para sus píxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;
- b) interpolar valores para subpíxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar una elección de una primera suma ponderada de valores para subpíxeles que residen en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y una segunda suma ponderada de valores para subpíxeles que residen en posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad, la primera y segunda sumas ponderadas de valores se calculan de acuerdo con la etapa (a); y
- c) interpolar un valor para un subpíxel situado en una posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$ de unidad al sacar un promedio ponderado del valor de un primer subpíxel o píxel situado en una posición horizontal de $1/2^{N-m}$ de unidad y vertical de $1/2^{N-m}$ de unidad y el valor de un segundo subpíxel o píxel ubicado en una posición horizontal de $1/2^{N-p}$ de unidad y vertical de $1/2^{N-q}$ de unidad, las variables m , n , p y q toman valores enteros en el intervalo de 1 a N de tal manera que el primer y segundo subpíxeles o píxeles están ubicados diagonalmente con respecto al subpíxel en la posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$.

El codificador de video puede comprender un codificador de video. Puede comprender un decodificador de video. Puede haber un códec que comprenda tanto un codificador como un decodificador de video.

De acuerdo con un quinto aspecto de la invención, se proporciona una terminal de comunicaciones que comprende un codificador de video para codificar una imagen que comprende píxeles dispuestos en filas y columnas y representado por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, el codificador de video comprende un interpolador adaptado para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, las posiciones horizontales y verticales fraccionadas están definidas de acuerdo con $1/2^x$, en donde x es un entero positivo que tiene un valor N máximo, el interpolador está adaptado para:

- a) interpolar valores para sus píxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar sumas ponderadas de

píxeles que residen en posiciones-horizontales de unidad y vertical de unidad;

b) interpolar valores para subpíxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar una elección de una primera suma ponderada de valores para subpíxeles que residen en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y una segunda suma ponderada de valores para subpíxeles que residen en posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad, la primera y segunda sumas ponderadas de valores se calculan de acuerdo con la etapa (a); y

c) interpolar un valor para un subpíxel situado en una posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$ de unidad al sacar un promedio ponderado del valor de un primer subpíxel o píxel situado en una posición horizontal de $1/2^{N-m}$ de unidad y vertical de $1/2^{N-n}$ de unidad y el valor de un segundo subpíxel o píxel ubicado en una posición horizontal de $1/2^{N-p}$ de unidad y vertical de $1/2^{N-q}$ de unidad, las variables m, n, p y q toman valores enteros en el intervalo de 1 a N de tal manera que el primer y segundo subpíxeles o píxeles están ubicados diagonalmente con respecto al subpíxel en la posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$.

La terminal de comunicaciones puede comprender un codificador de video. Puede comprender un decodificador de video. Preferiblemente, comprende un codee de video que comprende un codificador de video y un decodificador de video.

Preferiblemente, la terminal de comunicaciones que comprende una interfaz de usuario, un procesador y por lo menos uno de un bloque de transmisión y un bloque receptor, y un códec de video de acuerdo con por lo menos uno del tercer y cuarto aspectos de la invención. Preferiblemente, el procesador controla la operación del bloque de transmisión y/o el bloque receptor y el codificador de video.

De acuerdo con un sexto aspecto de la invención, se proporciona un sistema de telecomunicaciones que comprende una terminal de comunicaciones y una red, la red de telecomunicaciones y la terminal de comunicaciones están conectadas por un enlace de comunicaciones sobre el cual se puede transmitir video codificado, la terminal de comunicaciones comprende un codificador de video para codificar una imagen que comprende píxeles dispuestos en filas y columnas y representado por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, el codificador de video comprende un interpolador adaptado para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, las posiciones horizontales y verticales fraccionadas están definidas de acuerdo con $1/2^x$, en donde x es un entero positivo que tiene un valor N máximo, el interpolador está adaptado para:

a) interpolar valores para sus píxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;

b) interpolar valores para subpíxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar una elección de una primera suma ponderada de valores para subpíxeles que residen en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y una segunda suma ponderada de valores para subpíxeles que residen en posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad, la primera y segunda sumas ponderadas de valores se calculan de acuerdo con la etapa (a); y

c) interpolar un valor para un subpíxel situado en una posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$ de unidad al sacar un promedio ponderado del valor de un primer subpíxel o píxel situado en una posición horizontal de $1/2^{N-m}$ de unidad y vertical de $1/2^{N-n}$ de unidad y el valor de un segundo subpíxel o píxel ubicado en una posición horizontal de $1/2^{N-p}$ de unidad y vertical de $1/2^{N-q}$ de unidad, las variables m, n, p y q toman valores enteros en el intervalo de 1 a N de tal manera que el primer y segundo subpíxeles o píxeles están ubicados diagonalmente con respecto al subpíxel en la posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$.

Preferiblemente, el sistema de telecomunicaciones es un sistema de telecomunicaciones móvil que comprende una terminal de comunicaciones móvil y una red inalámbrica, la conexión entre la terminal de comunicaciones móvil y la red inalámbrica está formada por un enlace de radio. Preferiblemente, la red permite que la terminal de comunicaciones comunique con otras terminales de comunicaciones conectadas a la red sobre enlaces de comunicaciones entre las otras terminales de comunicaciones y la red.

De acuerdo con un séptimo aspecto de la invención, se proporciona un sistema de telecomunicaciones que comprende una terminal de comunicaciones y una red, la red de telecomunicaciones y la terminal de comunicaciones están conectadas por un enlace de comunicaciones sobre el cual se puede transmitir video codificado, la red comprende un codificador de video para codificar una imagen que comprende píxeles dispuestos en filas y columnas y representado por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, el codificador de video comprende un interpolador adaptado para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, las posiciones horizontales y verticales fraccionadas están definidas de acuerdo con $1/2^x$, donde x es un entero positivo que tiene un valor N máximo, el interpolador está adaptado para:

a) interpolar valores para sus píxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y

posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;

b) interpolar valores para subpíxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar una elección de una primera suma ponderada de valores para subpíxeles que residen en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y una segunda suma ponderada de valores para subpíxeles que residen en posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad, la primera y segunda sumas ponderadas de valores se calculan de acuerdo con la etapa (a); y

c) interpolar un valor para un subpíxel situado en una posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$ de unidad al sacar un promedio ponderado del valor de un primer subpíxel o píxel situado en una posición horizontal de $1/2^{N-m}$ de unidad y vertical de $1/2^N$ de unidad y el valor de un segundo subpíxel o píxel ubicado en una posición horizontal de $1/2^{N-p}$ de unidad y vertical de $1/2^{N-q}$ de unidad, las variables m, n, p y q toman valores enteros en el intervalo de 1 a N de tal manera que el primer y segundo subpíxeles o píxeles están ubicados diagonalmente con respecto al subpíxel en la posición horizontal de $1/2^N$ de unidad y vertical de $1/2^N$.

De acuerdo con un octavo aspecto de la invención, se proporciona un codificador de vídeo para codificar una imagen que comprende píxeles dispuestos en filas y columnas y representado por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, el codificador comprende un interpolador adaptado para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, la resolución de los subpíxeles está determinada por un entero positivo N, el interpolador está adaptado para:

a) interpolar valores para sus píxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;

b) interpolar un valor para subpíxeles en una posición horizontal de subpíxel y posición vertical de subpíxel se requiere directamente al usar una elección de una primera suma ponderada de valores para subpíxeles ubicados en una posición vertical correspondiente a la del subpíxel que se calcula y una segunda suma ponderada de valores para subpíxeles ubicados en una posición horizontal correspondiente a la del subpíxel que se calcula.

El interpolador puede estar adaptado además para formar la primera suma ponderada de valores de subpíxeles que residen en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y usar la primera suma ponderada para interpolar un valor para un subpíxel en una posición horizontal de $1/2^{N-1}$ de unidad y una posición vertical de $1/2^{N-1}$ de unidad.

El interpolador puede estar adaptado además para formar la segunda suma ponderada de valores de subpíxeles que residen en posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad y para usar un subpíxel en una posición horizontal de $1/2^{N-1}$ de unidad y una posición vertical de $1/2^{N-1}$ de unidad.

El interpolador puede estar adaptado además para interpolar valores para subpíxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de $1/2^{N-1}$ de unidad al sacar el promedio de por lo menos un par de un primer par de valores de subpíxel ubicado en una posición horizontal de $1/2^N$ de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de unidad y vertical de $1/2^{N-1}$ de unidad y un segundo par de valores de un píxel ubicado en una posición horizontal de unidad y vertical de unidad, y un subpíxel ubicado en una posición horizontal de $1/2^N$ de unidad y vertical de $1/2^{N-1}$ de unidad.

De acuerdo con un noveno aspecto de la invención, se proporciona una terminal de comunicaciones que comprende un codificador de video para codificar una imagen que comprende píxeles dispuestos en filas y columnas y representado por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, el codificador comprende un interpolador adaptado para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, la resolución de los subpíxeles está determinada por un entero positivo N, el interpolador está adaptado para:

a) interpolar valores para sus píxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;

b) interpolar un valor para subpíxeles en una posición horizontal de subpíxel y vertical de subpíxel directamente al usar una elección de una primera suma ponderada de valores para subpíxeles ubicados en una posición vertical correspondiente a la del subpíxel que se calcula y una segunda suma ponderada de valores para subpíxeles ubicados en una posición horizontal correspondiente a la del subpíxel que se calcula.

De acuerdo con un décimo aspecto de la invención, se proporciona un sistema de telecomunicaciones que comprende una terminal-de comunicaciones y una red, la red de telecomunicaciones y la terminal de comunicaciones están conectadas por un enlace de comunicaciones sobre el cual se puede transmitir video codificado, la terminal de comunicaciones comprende un codificador de video para codificar una imagen que

comprende píxeles dispuestos en filas y columnas y representado por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, el codificador de video comprende un interpolador adaptado para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, la resolución de los subpíxeles está determinada por un entero positivo N, el interpolador está adaptado para:

- a) interpolar valores para sus píxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;
- b) interpolar un valor para subpíxeles en una posición horizontal de subpíxel y posición vertical de subpíxel directamente al usar una elección de una primera suma ponderada de valores para subpíxeles ubicados en una posición vertical correspondiente a la del subpíxel que se calcula y una segunda suma ponderada de valores para subpíxeles ubicados en una posición horizontal correspondiente a la del subpíxel que se calcula.

De acuerdo con un décimo primer aspecto de la invención, se proporciona un sistema de telecomunicaciones que comprende una terminal de comunicaciones y una red, estando la red de telecomunicaciones y la terminal de comunicaciones conectadas por un enlace de comunicaciones sobre el cual se puede transmitir video codificado, la red comprende un codificador de video para codificar una imagen que comprende píxeles dispuestos en filas y columnas y representado por valores que tienen un intervalo dinámico especificado, los píxeles en las filas residen en posiciones horizontales de unidad y los píxeles en las columnas residen en posiciones verticales de unidad, el codificador comprende un interpolador adaptado para generar valores para subpíxeles en posiciones horizontales y verticales fraccionadas, la resolución de los subpíxeles está determinada por un entero positivo N, el interpolador está adaptado para:

- a) interpolar valores para sus píxeles en posiciones horizontales de $1/2^{N-1}$ de unidad y verticales de unidad y posiciones horizontales de unidad y verticales de $1/2^{N-1}$ de unidad directamente al usar sumas ponderadas de píxeles que residen en posiciones horizontales de unidad y verticales de unidad;
- b) interpolar un valor para subpíxeles en una posición horizontal de subpíxel y posición vertical de subpíxel se requiere directamente al usar una elección de una primera suma ponderada de valores para subpíxeles ubicados en una posición vertical correspondiente a la del subpíxel que se calcula y una segunda suma ponderada de valores para subpíxeles ubicados en una posición horizontal correspondiente a la del subpíxel que se calcula.

Breve descripción de las figuras

Una realización de la invención se describirá ahora a modo de ejemplo sólo con referencia a los dibujos adjuntos, en los cuales:

- La figura 1 muestra un codificador de video de acuerdo con la técnica anterior;
- La figura 2 muestra un decodificador de video de acuerdo con la técnica anterior;
- La figura 3 muestra los tipos de tramas usadas en codificación de video;
- Las figuras 4a, 4b y 4c muestran las etapas en la coincidencia de bloques;
- La figura 5 ilustra el proceso de estimación de movimiento para resolución de subpíxeles;
- La figura 6 muestra un dispositivo de terminal que comprende codificación de video y equipo de codificación en el cual se puede implementar el método de la invención;
- La figura 7 muestra un codificador de video de acuerdo con una realización de la presente invención;
- La figura 8 muestra un decodificador de video de acuerdo con una realización de la presente invención;
- Las figuras 9 y 10 no se han usado y cualquiera de esas figuras se debe desechar;
- La figura 11 muestra un diagrama esquemático de una red de telecomunicaciones móvil de acuerdo con una realización de la presente invención;
- La figura 12a muestra una notación para describir posiciones de píxeles y subpíxeles específicas para TML5;
- La figura 12b muestra interpolación de subpíxeles de media resolución;
- La figura 12c muestra interpolación de subpíxeles de media resolución;
- La figura 13a muestra una notación para describir posiciones de píxel y subpíxel específicas para TML6;
- La figura 13b muestra interpolación de subpíxeles de media resolución;
- La figura 13c muestra interpolación de subpíxeles de media resolución;
- La figura 14 muestra una notación para describir posiciones de píxel y subpíxel específicas para la invención;
- La figura 14b muestra interpolación de subpíxeles de media resolución de acuerdo con la invención;
- La figura 14c muestra interpolación de subpíxeles de media resolución de acuerdo con la invención;
- La figura 15 muestra posibles elecciones de interpolación diagonal para subpíxeles;
- La figura 16 muestra los valores de subpíxeles de media resolución requeridos para calcular otros valores de subpíxeles de media resolución;
- La figura 17a muestra los valores de subpíxeles de media resolución que se deben calcular para interpolar valores para subpíxeles de un cuarto de resolución en un bloque de imagen que usa el método de interpolación de TML5;
- La figura 17b muestra los valores de subpíxeles de media resolución que se deben calcular para interpolar

valores para subpíxeles de un cuarto de resolución en un cuarto de imagen mediante el uso del método de interpolación de acuerdo con la invención;

La figura 18a muestra los números de subpíxeles de media resolución que se deben calcular para obtener valores para subpíxeles de un cuarto de resolución dentro de un bloque de imagen mediante el uso del método de interpolación de valor de subpíxel de acuerdo con TML5;

La figura 18b muestra los números de subpíxeles de media resolución que se deben calcular para obtener valores para subpíxeles de un cuarto de resolución dentro de un bloque de imagen mediante el uso del método de interpolación de valor de subpíxel de acuerdo con la invención;

La figura 19 muestra un esquema de numeración para cada una de las posiciones de subpíxeles;

La figura 20 muestra la nomenclatura usada para describir píxeles, subpíxeles de media resolución, subpíxeles de un cuarto de resolución y subpíxeles de un octavo de resolución;

La figura 21a muestra la dirección diagonal para usarse en la interpolación de cada subpíxel de un octavo de resolución en una realización de la invención;

La figura 21b muestra la dirección diagonal para ser usada en la interpolación de cada subpíxel de un octavo de resolución en otra realización de la invención; y

La figura 22 muestra la nomenclatura usada para describir 15 subpíxeles de un octavo de resolución dentro de un bloque de imagen.

Descripción detallada de la invención

Las figuras 1 a 5, 12a, 12b, 12c, 13a, 13b, y 13c se han descrito anteriormente.

La figura 6 presenta un dispositivo terminal que comprende equipo de codificación y decodificación de video que se puede adaptar para operar de acuerdo con la presente invención. De manera más precisa, la figura ilustra una terminal multimedia 60 implementada de acuerdo con la recomendación H.324 de ITU-T. La terminal puede considerarse como un dispositivo transceptor multimedia. Incluye elementos que capturan, codifican y multiplexan corriente de datos multimedia para transmisión a través de una red de comunicaciones, así como elementos que reciben, desmultiplexan, decodifican y exhiben contenido multimedia recibido. La recomendación H.324 de ITU-T define la operación global de la terminal y se refiere a otras recomendaciones que gobiernan la operación de sus diversas partes constituyentes. Este tipo de terminal multimedia se puede usar en aplicaciones de tiempo real tal como videotelefonía de conversación o aplicaciones que no son de tiempo real tales como la recuperación/transmisión en paquetes de video clips, por ejemplo de un seguidor de contenido multimedia en Internet.

En el contexto de la presente invención, se debe apreciar que la terminal H.324 mostrada en la figura 6 es solamente una de un número de implementaciones terminales multimedia alternativas adecuadas para la aplicación del método de la invención. También debe indicarse que existe un número de alternativas relacionadas con la posición e implementación del equipo terminal. Como se ilustra en la figura 6, la terminal multimedia se puede localizar en equipo de comunicaciones conectado a una red de telefonía de línea fija tal como una red de telefonía de computación pública (PSTAT) análoga. En este caso, la terminal multimedia está equipada con un módem 71, de acuerdo con las recomendaciones V.8, V.34 de ITU-T y opcionalmente V.8bis. En forma alternativa, la terminal multimedia se puede conectar un módem externo. El módem permite la conversión de los datos digitales multiplexados y señales de control producidas por la terminal multimedia en una forma análoga adecuada para transmisión en la PSTN. Además permite que la terminal multimedia reciba datos y señales de control en forma análoga de la PSTN y las convierta en una corriente de datos digitales que pueden ser desmultiplexados y procesados de una manera apropiada por la terminal.

Una terminal multimedia H.324 también se puede implementar de tal manera que pueda ser conectada directamente a una red de línea fija digital, tal como una red digital de servicios integrados (ISDN). En este caso, el módem 71 es reemplazado por una interfaz de red de usuario ISDN. En la figura 6, esta interfaz de usuario-red de ISDN está representada por el bloque 72 alternativo.

Las terminales multimedia H.324 también se pueden adaptar para usarse en comunicaciones móviles. Se usa con un enlace de comunicación inalámbrico, el módem 71 puede ser reemplazado por cualquier interfaz inalámbrica apropiada, como se representa mediante el bloque 73 en la figura 6. Por ejemplo, una terminal multimedia H.324/M puede incluir un transceptor de radio que permite conexión a la red de telefonía móvil GSM de 2a generación actual, o al sistema de telefonía móvil universal (UMTS) de tercera generación.

Debe indicarse que en terminales multimedia diseñadas para comunicación en dos sentidos, es decir, para transmisión y recepción de datos de video, es ventajoso proveer tanto un codificador de video como un decodificador de video implementado de acuerdo con la presente invención. Ese par de codificador y decodificador con frecuencia es implementado como una unidad funcional combinada individual, referida como un "códec".

Como un codificador de video de acuerdo con la invención realiza codificación de video compensada en cuanto a movimiento a una resolución de subpíxel mediante el uso de un esquema de interpolación específico y una combinación particular de interpolación de valor de subpíxel de antemano y sobredemanda, es generalmente

necesario un decodificador de video de una terminal receptora sea implementado de una manera compatible con el codificador de la terminal de transmisión que formó la corriente de datos de video comprimidos. El no asegurar esta compatibilidad puede tener un efecto adverso sobre la calidad de la compensación en movimiento y la exactitud de tramas de video reconstruidos.

5 Una terminal multimedia H.324 típica se describirá ahora con detalle adicional con referencia a la figura 6.

La terminal multimedia 60 incluye varios elementos referidos como "equipo terminal". Este incluye dispositivos de video, audio y telemáticos, indicados generalmente por los números de referencia 61, 52 y 63, respectivamente. El equipo de video 61 puede incluir, por ejemplo, una cámara de video para capturar imágenes de video, un monitor para exhibir contenido de video recibido y un equipo de procesamiento de video opcional. El equipo de audio 62 normalmente incluye un micrófono, por ejemplo para capturar mensajes hablados, y una bocina para reproducir contenido de audio recibido. El equipo de audio también puede incluir unidades de procesamiento de audio adicionales. El equipo telemático 63, puede incluir una terminal de datos, teclado, pizarrón electrónico o un transceptor de imagen quieta, tal como una unidad de fax.

El equipo de video 61 está acoplado a un códec de video 65. El códec de video 65 comprende un codificador de video y un decodificador de video correspondiente ambos implementados de acuerdo con la invención. Dicho codificador y decodificador se describirán a continuación. El códec de video 65 es responsable de codificar datos de video capturados en una forma apropiada para transmisión adicional en un enlace de comunicaciones y decodificar contenido de video comprimido recibido de la red de comunicaciones. En el ejemplo ilustrado en la figura 6, el códec de video es incrementado de acuerdo con la recomendación H.324 de ITU-T, con modificaciones apropiadas a implementar el método de interpolación de valor de subpíxel de acuerdo con la invención tanto en el codificador como en el decodificador del códec de video.

De manera similar, el equipo de audio de la terminal está acoplado a un códec de audio, indicado en la figura 6 por el número de referencia 66. Igual que el códec de video, el códec de audio comprende un par de codificador/decodificador. Convierte los datos de audio capturados por el equipo de audio terminal en una forma adecuada para transmisión en la red de comunicaciones y transforma los datos de audio codificados recibidos de la red de nuevo a una forma adecuada para reproducción, por ejemplo un altavoz de la terminal. La salida del códec de audio pasa a un bloque de demora 67. Esto compensa las demoras introducidas por el proceso de codificación de video y por lo tanto asegura la sincronización de contenido de audio y video.

El bloque de control de sistema 64 de la terminal multimedia controla la señalización de extremo a red que usa un protocolo de control apropiado (bloque de señalización 68) para establecer un modo común de, operación entre una terminal de transmisión y una de recepción. El bloque de señalización 68 intercambia información acerca de las capacidades de codificación y decodificación de las terminales de transmisión y recepción y se puede usar para permitir los diversos modos de codificación del codificador de video. El bloque de control del sistema 64 también controla el uso de cifrado de datos. Información referente al tipo de cifrado para usarse en la transmisión de datos se hace pasar del bloque de cifrado 69 al multiplexor/desmultiplexor (unidad MUX/DMUX) 70.

Durante la transmisión de datos desde la terminal multimedia, la unidad de MUX/DMUX 70 combina corrientes de video y audio codificadas y sincronizadas con entrada de datos desde el equipo telemático 63 y posibles datos de control, para formar una sola corriente de bits. La información referente al tipo de cifrado de datos (si acaso la hay) se puede aplicar a la corriente de bits, provista por el bloque de cifrado 69, se usa para seleccionar un modo de cifrado. De manera correspondiente, cuando una corriente de bits multimedia multiplexada posiblemente cifrada es recibida, la unidad MUX/DMUX 70 es responsable del cifrado de la corriente de bits, la divide en sus componentes multimedia constituyentes y hace pasar esos componentes al códec(s) apropiado(s) y/o el tipo de terminal la decodificación y reproducción.

Debe indicarse que los elementos funcionales de la terminal multimedia, codificador de video, decodificador y códec de video de acuerdo con la invención se pueden implementar como software o hardware dedicado, o una combinación de los dos. Los métodos de codificación y decodificación de video de acuerdo con la invención son particularmente adecuados para la implementación en forma de un programa de ordenador que comprende instrucciones legibles por máquina para realizar las etapas funcionales de la invención. Como tales, el codificador y decodificador de acuerdo con la invención se pueden implementar como un código de software almacenado en un medio de almacenamiento y ejecutado en un ordenador, tal como un ordenador personal, para proveer a la computadora de funcionalidad de codificación/decodificación de video.

Si la terminal multimedia 60 es una terminal móvil, es decir si está equipada con un transceptor de radio 73, los expertos en la técnica entenderán que también puede comprender elementos adicionales. En una realización, comprende una interfaz de usuario que tiene una pantalla y un teclado, que permiten la operación de la terminal multimedia 60 por parte de un usuario, junto con bloques funcionales necesarios que incluyen una unidad de procesamiento central, tal como un microprocesador, que controla los bloques responsables para diferentes funciones en la terminal multimedia, una memoria de acceso aleatorio RAM, una memoria de solo lectura ROM, y una cámara digital. Las instrucciones de operación del microprocesador, es decir un código de programa

correspondiente a las funciones básicas de la terminal multimedia 60, es almacenado en la memoria de sólo lectura ROM y puede ser ejecutado según se requiera por el microprocesador, por ejemplo bajo el control del usuario. De acuerdo con el código de programa, el microprocesador usa el transceptor de radio 73 para formar una conexión con la red de comunicación móvil, lo que permite a la terminal multimedia 60 transmitir información y recibir información de la red de comunicación móvil en una trayectoria de radio.

El microprocesador monitoriza el estado de la interfaz del usuario que controla la cámara digital. En respuesta a un comando de usuario, el microprocesador instruye a la cámara para grabar imágenes digitales en la RAM. Una vez que la imagen es capturada, o alternativamente durante el procedimiento de captura, el microprocesador segmenta la imagen en segmentos de imagen (por ejemplo macrobloques). Y usa el codificador para realizar codificación compensada en movimiento para los segmentos para generar una secuencia de imágenes comprimidas como se explica en la descripción anterior. Un usuario puede ordenar a la terminal multimedia 60 que exhiba las imágenes capturadas en su pantalla o envíe la secuencia de imágenes comprimidas mediante el uso del transceptor de radio 73 a otra terminal multimedia, un teléfono de video conectado a una red de línea fija (PSTN) o algún otro dispositivo de telecomunicaciones. En una realización preferida, la transmisión de datos de imagen se inicia tan pronto como el primer segmento es codificado, por lo que el receptor puede iniciar un proceso de decodificación correspondiente con una demora mínima.

La figura 11 es un diagrama esquemático de una red de telecomunicaciones móvil de acuerdo con una realización de la invención. Las terminales multimedia MS están en comunicación con las estaciones de base BTS por medio de un enlace de radio. Las estaciones de radio BTS son posiblemente conectadas, a través de una denominada interfaz de Abis, a un controlador de estación de base BSC, que controla y administra varias estaciones de base.

La entidad formada por un número de estaciones de base BTS (normalmente por algunas decenas de estaciones de base) y un solo controlador de estación de base BSC, que controla las estaciones de base, se denomina un subsistema de estación de base BSS. Particularmente, el controlador de estación de base BSC administra canales de comunicación de radio y trasposos. El controlador de estación de base BSC también está conectado, a través de una denominada interfaz A, a un centro de conmutación de servicios móviles MSC, que coordina la formación de conexiones hacia y desde estaciones móviles. Se hace una conexión adicional a través del centro de conmutación de servicios móviles MSC, al exterior de la red de comunicaciones móviles. Fuera de la red de comunicaciones móviles puede residir otra red(es) conectada a la red de comunicaciones móviles mediante una pasarela(s) GTW, por ejemplo Internet o una red de telefonía computada pública (PSTN). En esa red externa, o dentro de la red de telecomunicaciones, puede haber estaciones de decodificación o codificación de video localizadas, tales como computadoras PC. En una realización de la invención, la red de telecomunicaciones móviles comprende un servidor de video VSRVR para proveer datos de video a un MS que suscribe a ese servicio. Los datos de video se comprimen mediante el uso del método de compresión de video compensado en movimiento como se describe anteriormente. El servidor de video puede funcionar como una pasarela a una fuente de video en línea o puede comprender video clips previamente grabados. Las aplicaciones de video telefonía pueden implicar, por ejemplo, dos estaciones móviles o una estación móvil MS y un videoteléfono conectado a la PSTN, una PC conectada a Internet o una terminal compatible con H.261 conectada ya sea a Internet o a la PSTN.

La figura 7 muestra un codificador de video 700 de conformidad con una realización de la invención. La figura 8 muestra un decodificador de video 800 de acuerdo con una realización de la invención.

El codificador 700 comprende una entrada 701 para recibir una señal de video de una cámara u otra fuente de video (no mostrada). Además comprende un transformador de DCT 705, un cuantificador 706, un cuantificador inverso 709, un transformador de DCT inverso 710, combinadores 712 y 716, un bloque de incorporación de subpíxeles de antemano 730, un almacén de tramas 740 y un bloque de interpolación de subpíxeles sobredemanda 750, implementado en combinación con un bloque de estimación de movimiento 760. El codificador también comprende un bloque de codificación de campo de movimiento 770 y un bloque de predicción compensado en movimiento 780. Los conmutadores 702 y 714 son operados cooperativamente por un administrador de control 720 para conmutar el codificador entre un modo INTRA de codificación de video y un modo INTER de codificación de Video. El codificador 700 también comprende una unidad multiplexora (MUX/RMUX) 790 para formar una sola corriente de bits es de numerosos tipos de información producidos por el codificador 700 para transmisión posterior a una terminal receptora remota, o por ejemplo para almacenamiento en un medio de almacenamiento masivo tal como una unidad de disco duro de ordenador (no mostrada).

Debe indicarse que la presencia y las implementaciones de bloque de interpolación de subpíxeles de antemano 730 y bloque de interpolación de valor de subpíxeles sobredemanda 750 en la arquitectura del codificador dependen de la forma en la cual el método de interpolación de subpíxeles de conformidad con la invención se aplica. En realizaciones de la invención en las cuales no se realiza interpolación de valor de subpíxeles de antemano, el codificador 700 no comprende bloque de interpolación de valor de subpíxeles de antemano 730. En otras realizaciones de la invención, solo la interpolación de subpíxeles de antemano se realiza y por lo tanto el codificador no incluye bloque de interpolación de valor de subpíxeles sobredemanda 750. En realizaciones en las cuales se realiza interpolación de valor de subpíxeles de antemano y sobredemanda, ambos bloques 730 y 750 están presentes en el codificador 700.

La operación del codificador 700 de acuerdo con la invención se describirá ahora con detalle, En la descripción, se supondrá que cada trama de video no comprimido, recibido desde la fuente de video en la entrada 701, es recibida y procesada sobre una base de macrobloque por bloque, preferiblemente en un orden de escaneo de fondo. Se supondrá además que cuando la codificación de una nueva secuencia de video empieza, la primera trama de la secuencia se codifica en modo INTRA. Posteriormente, el codificador está programado para codificar cada trama en formato INTER, a menos que se cumpla una de las siguientes condiciones: 1) se juzgue que la trama actual que es codificado es tan diferente de la trama de referencia usada en su predicción que se produce información de error de producción excesiva; 2) un intervalo de repetición INTRA trama predefinido a estirado; o 3) la retroalimentación es recibida desde una terminal receptora que indica una solicitud para una trama que ha de ser codificado en trama INTRA..

La aparición de la condición 1) se detecta al monitorizar la salida del combinador 716. El combinador 716 forma una diferencia entre el macrobloque actual de la trama que es codificado y su predicción, producido en el bloque de predicción compensado en movimiento 780. Si una medición de esta diferencia (por ejemplo una suma de diferencias absolutas de valores de píxel) excede un umbral predeterminado, el combinador 716 informa al administrador de control 720 mediante una línea de control 717 y el administrador de control 720 opera los conmutadores 702 y 714 para conmutar el codificador 700 en modo de codificación INTRA. La aparición de la condición 2) es monitoreada por medio de un controlador de tiempo o contador de tramas implementado en el administrador de control 720, de tal manera que si el controlador de tiempo expira, o el contador de trama alcanza un número predeterminado de tramas, el administrador de control 720 opera los conmutadores 702 y 714 para conmutar el codificador en modo de codificación INTRA. La condición 3) es activada si el administrador de control 720 recibe una señal de retroalimentación desde, por ejemplo, una terminal receptora, a través de una línea de control 718 que indica que una renovación de trama INTRA es requerida por la terminal receptora. Esa condición podría surgir, por ejemplo, si una trama previamente transmitida fuera corrompida por interferencia durante su transmisión, haciendo imposible decodificar en el receptor. En esta situación, el receptor emitiría una solicitud para que -el siguiente-trama fuera codificado el formato INTRA, con lo que se reinicializaría la secuencia de codificación.

Se supondrá además que el codificador y decodificador son implementados de tal manera que permiten la determinación de vectores de movimiento con una resolución espacial de una resolución de hasta un cuarto de píxel. Como se verá a continuación, también son posibles niveles más finos de resolución.

La operación del codificador 700 en modo de codificación INTRA se describirá ahora. En el modo INTRA, el administrador de control 720 opera el conmutador 702 para aceptar entrada de vídeo desde la línea de entrada 719. La entrada de serial de video es recibida macrobloque por macrobloque desde la entrada 701 a través de la línea de entrada 719 y cada macrobloque de píxeles de imagen original es transformado a coeficientes de DCT por el transformador DCT 705. Los coeficientes de DCT después se hacen pasar al cuantificador 706, donde son cuantificados mediante el uso de un parámetro de cuantificación QP. La selección del parámetro de cuantificación QP es controlado por el administrador de control 720 a través de la línea de control 722. Cada macrobloque transformado y cuantificado por DCT que constituye la información de imagen codificada por INTRA 723 de la trama se hace pasar desde el cuantificador 706 hasta el MUX/DMUX' 790. El MUX/DMUX 790 combina la información de imagen codificada por INTRA con información de control posible (por ejemplo datos de encabezamiento, información de parámetros de cuantificación, datos de corrección de errores, etc.) para formar una sola corriente de bits de información de imagen codificadas 725. La codificación de longitud variable (VLC) se usa para reducir la redundancia de la corriente de bits de video comprimido, como lo conocen los expertos en la técnica.

Una imagen localmente decodificada se forma en el codificador 700 al pasar la salida de datos por el cuantificador 706 a través del codificador inverso 709 y aplicar una transformación de DCT inversa 710 a los datos cuantificados inversos. Los datos resultantes son después introducidos al combinador 712. En el modo INTRA, el conmutador 714 se fija de manera que la entrada al combinador 712 a través del conmutador 714 se fije a cero. De esta manera la operación realizada por el combinador 712 es equivalente a pasar los datos de imagen decodificados formados por el cuantificador inverso 709 y la transformación de DCT inversa 710 no alterada.

En realizaciones de la invención en las cuales se realiza interpolación de valor de subpíxel de antemano, la salida del combinador 712 se aplica al bloque de interpolación de subpíxeles de antemano 730. La entrada al bloque de interpolación de valor de subpíxeles de antemano 730 toma la forma de bloque de imagen decodificada. En el bloque de interpolación de valor de subpíxel de antemano 730, cada macrobloque decodificado es sometido a interpolación de subpíxel de tal manera que un subconjunto predeterminado de valores de subpíxeles de resolución de subpíxel se calcula de acuerdo con el método de interpolación de la invención y se almacena junto con los valores de píxel decodificados en el almacén de tramas 740.

En realizaciones en las cuales no se realiza la interpolación de subpíxeles de antemano, el bloque de interpolación de subpíxeles de antemano no está presente en la arquitectura del codificador y la salida del combinador 710, que comprende bloques de imagen decodificadas, se aplica directamente al almacén de tramas 740.

A medida que los macrobloques posteriores de la trama actual son recibidos y pasan por las etapas de codificación y

5 decodificación anteriormente descritas en los bloques 705, 706, 709, 710, 712, una versión decodificada de las tramas INTRA se acumula en el almacén de trama 740. Cuando el último macrobloque de la trama actual ha sido INTRA codificado y posteriormente decodificado, el almacén de tramas 740 contiene una trama completamente decodificado, disponible para usarse como una trama de referencia de predicción en la codificación de una trama de video posteriormente recibido en el formato INTER. En realizaciones de la invención en las cuales se realiza interpolación de valor de subpíxel de antemano, la trama de referencia mantenido en el almacén de trama 740 es por lo menos parcialmente interpolado a resolución de subpíxel.

10 La operación del codificador 70 en el modo de codificación INTER se describirá ahora. En el modo de codificación INTER, el administrador de control 720 opera el conmutador 702 para recibir su entrada desde la línea 721, que comprende la salida del combinador 716. El combinador 716 forma información de error de predicción que representa la diferencia entre el macrobloque actual de la trama que es codificado y su predicción, producida en el bloque de predicción compensado en movimiento 780. La información de error de predicción es transformada por DCT en el bloque 705 y cuantificada en el bloque 706 para formar un macrobloque de la información de error de predicción transformada por DCT y cuantificada. Cada macrobloque de información de error de predicción transformada por DCT y cuantificada se hace pasar desde el cuantificador 706 hasta la unidad MUX/DMUX 790. La unidad MUX/DMUX 790 combina la información de error de predicción 723 con coeficientes de movimiento 724 (descritos a continuación) e información de control (por ejemplo datos de encabezamiento, información de parámetros de cuantificación, datos de corrección de errores, etc.) para formar una sola corriente de bits de información de imagen codificada 725.

20 La información de error de predicción localmente decodificada para cada macrobloque de la trama INTER-codificado es después formado en el codificador 700 al hacer pasar la información de error de predicción codificada 723 enviada por el cuantificador 706 a través del cuantificador inverso 709 y aplicar una transformación por DCT en el bloque 710. El macrobloque localmente decodificado resultante de la información de error de predicción es después introducido al combinador 712. En el modo INTER, el conmutador 714 se fija de modo que el combinador 712 también reciba macrobloques predichos en movimiento para el INTER-trama actual, producido en el bloque de predicción carpeteado en movimiento 780. El combinador 712 combina estas dos piezas de información para producir bloques de imagen reconstruidas del INTER-trama actual.

30 Como se describió antes, cuando se consideran las INTRA-tramas codificadas, en realizaciones de la invención en los cuales se realiza interpolación de valor de subpíxel de antemano, la salida del combinador 712, se aplica al bloque de interpolación de subpíxel de antemano 730. Por lo tanto, la entrada al bloque de interpolación de valor de subpíxel de antemano 730 en el modo de codificación INTER también adopta la forma de bloques de imagen decodificados. En el bloque de interpolación de valor de subpíxel de antemano 730, cada macrobloque decodificado es sometido a interpolación de subpíxel de tal manera que un subconjunto predeterminado de valores de subpíxel se calcula de acuerdo con el método de interpolación de la invención y se almacena junto con los valores de píxel decodificados en el almacén de trama 740. En realizaciones en las cuales no se realiza interpolación de subpíxel de antemano, el bloque de interpolación de subpíxel de antemano no está presente en la arquitectura del codificador y la salida del combinador 12, que comprende bloques de imagen decodificadas, se aplica directamente al almacén de trama 740.

45 A medida que los macrobloques posteriores de la señal de video son recibidos desde la fuente de video y pasan por las etapas de codificación y decodificación previamente descritas en los bloques 705, 706, 709, 710, 712, una versión codificada del INTER-trama se construye en el almacén de tramas 740. Cuando el último macrobloque de la trama ha sido INTER-codificado y posteriormente decodificado, el almacén de tramas 740 contiene una trama completamente decodificado, disponible para usarse como una trama de referencia de predicción en la codificación de una trama de video posteriormente recibida en formato INTER. En las realizaciones de la invención en las cuales se realiza interpolación de valor de subpíxel de antemano, la trama de referencia mantenido en el almacén de trama 740 es por lo menos parcialmente interpolado a resolución de subpíxel.

50 La formación de una predicción de un macrobloque de la trama actual se describirá ahora.

55 Cualquier trama codificada en formato INTER requiere una trama de referencia para predicción compensada en movimiento. Esto significa, entre otras cosas, que cuando se codifica una secuencia de video, la primera trama que ha de ser codificada, ya sea la primera trama en la secuencia o alguna otra trama, debe ser codificada en formato INTRA. Esto, a su vez, significa que cuando el codificador de video 700 es conmutado en modo de codificación INTER por el administrador de control 720, una trama de referencia completa, formado al decodificar localmente una trama previamente codificada, ya está disponible en el almacén de trama 740 del codificador. En general, la trama de referencia se forma por decodificación local ya sea de una trama INTRA-codificada o de una trama INTER-codificada.

60 La primera etapa en la formación de una predicción para un macrobloque de la trama actual se realiza mediante el bloque de estimación de movimiento 760. El bloque de estimación de movimiento 760 recibe el macrobloque actual de la trama que es codificado a través de la línea 727 y realiza una operación de coincidencia de bloques para identificar una región en la trama de referencia que corresponda sustancialmente al macrobloque actual. De acuerdo con la invención, el proceso de coincidencia de bloques se realiza a resolución de subpíxel de una manera que

depende de la implementación del codificador 700 y el grado de interpolación de subpíxel de antemano realizado. Sin embargo, el principio básico en que se basa el proceso de coincidencia de bloque es similar en todos los casos. Específicamente, el bloque de estimación de movimiento 760 realiza coincidencia de bloques al calcular los valores de diferencia (por ejemplo, la suma de las diferencias absolutas) que representa la diferencia en valores de píxel entre el macrobloque de la trama actual bajo examen y las regiones de mejor coincidencia candidatas de píxeles/subpíxeles en la trama de referencia. Un valor de diferencia se produce para todos los posibles desplazamientos (por ejemplo, desplazamientos x , y en precisión de un cuarto o un octavo de subpíxel) entre el macrobloque de la trama actual y la región de prueba candidata dentro de una región de búsqueda predefinida de la trama de referencia y el bloque de estimación de movimiento 760 determina el valor de diferencia calculado más pequeño. El desplazamiento entre el macrobloque en la trama actual y la región de prueba candidata de valores de píxel/valores de subpíxel en la trama de referencia que produce el valor de diferencia más pequeño define el vector de movimiento para el macrobloque en cuestión. En ciertas realizaciones de la invención, una estimación inicial para el vector de movimiento que tiene precisión de píxel de unidad se determina primero y después se refina a un nivel más fino de precisión de subpíxel, como se describe anteriormente.

En realizaciones del codificador en el cual no se realiza interpolación de valor de subpíxel de antemano, todos los valores de subpíxeles requeridos en el proceso de coincidencia de bloques se calcula en el bloque de interpolación de valor de subpíxel bajo demanda 750. El bloque de estimación de movimiento 760 controla al bloque de interpolación de valor de subpíxel bajo demanda 750 para calcular cada valor de subpíxel necesario en el proceso de coincidencia de bloques de una manera bajo demanda, siempre y cuando se requiera. En este caso, el bloque de estimación de movimiento 760 se puede implementar para realizar coincidencia de bloques como un proceso de una etapa, en cuyo caso un vector de movimiento con la resolución de subpíxeles deseada se busca directamente, o se puede implementar para realizar coincidencia de bloques como un proceso de dos etapas. Si se adopta el proceso de dos etapas, la primera etapa puede comprender una búsqueda, por ejemplo, de un vector de movimiento de resolución de píxel completo o medio píxel y la segunda etapa se realiza para refinar el vector de movimiento de la resolución de subpíxel deseada. Puesto que la coincidencia de bloques es un proceso exhaustivo, en el cual los bloques de $n \times m$ píxeles en la trama actual son comparados uno por uno con los bloques de $n \times m$ píxeles o subpíxeles en la trama de referencia interpolados, se debe apreciar que un subpíxel calculado de una manera bajo demanda por el bloque de interpolación de píxel bajo demanda 750 puede necesitar ser calculado muchas veces a medida que se determinan los valores de diferencia sucesivos. En un codificador de video, este enfoque no es el más eficiente posible en términos de complejidad/carga computacional.

En realizaciones del codificador que usa sólo interpolación de valor de subpíxel de antemano, la coincidencia de bloque se puede realizar como un proceso de una etapa, ya que todos los valores de subpíxeles de la trama de referencia requeridos para determinar un vector de movimiento con la resolución de subpíxel deseada se calculan de antemano en el bloque 730 y se almacenan en el almacén de tramas 740. Por lo tanto, están directamente disponibles para usarse en el proceso de coincidencia de bloques y pueden recuperarse según se requiera del almacén de trama 740 mediante el bloque de estimación de movimiento 760. Sin embargo, aún en el caso en donde todos los valores de subpíxel están disponibles del almacén de trama 740, es más eficiente computacionalmente realizar coincidencia de bloques como un proceso de dos etapas, ya que se requieren menos cálculos de diferencia. Cabe apreciar que aunque la interpolación de valor de subpíxel de antemano completa reduce la complejidad computacional en el codificador, no es el enfoque más eficiente en términos de consumo de memoria.

En realizaciones del codificador en las cuales se usa interpolación de valor de subpíxel de antemano y bajo demanda, el bloque de estimación de movimiento 760 se implementa de tal manera que pueda recuperar valores de subpíxel previamente calculados en bloque de interpolación de valor de subpíxel de antemano 730 y almacenar en el almacén de tramas 740 y además controlar el bloque de interpolación de valores de subpíxel bajo demanda 750 para calcular cualesquiera valores de subpíxel adicionales que pudieran ser requeridos. El proceso de coincidencia de bloques se puede realizar como un proceso de una sola etapa o de dos etapas. Si se usa una implementación de dos etapas, los valores de subpíxel calculados de antemano recuperados del almacén de tramas 740 se puede usar en la primera etapa del proceso y la segunda etapa se puede implementar para usar valores de subpíxel calculados por el bloque de interpolación de valor de subpíxel bajo demanda 750. En este caso, ciertos valores de subpíxel usados en la segunda etapa del proceso de coincidencia de bloques pueden necesitar calcular muchas veces a medida que se hacen comparaciones sucesivas, pero el número de esos cálculos duplicados es significativamente menor que si no se usara cálculo de valor de subpíxeles de antemano. Además, el consumo de memoria se reduce con respecto a realizaciones en las cuales se usa sólo interpolación de valor de subpíxel de antemano.

Una vez que el bloque de estimación de movimiento 760 ha producido un vector de movimiento del macrobloque de la trama actual bajo examen, envía el vector de movimiento al bloque de codificación de campo 770. El bloque de codificación de campo de movimiento 770 entonces se aproxima al vector de movimiento recibido del bloque de estimación de movimiento 760 mediante el uso de un modelo de movimiento. El modelo de movimiento generalmente comprende un conjunto de funciones básicas. De manera más específica, el bloque de codificación de campo de movimiento 770 representa al vector de movimiento como un conjunto de valores de coeficiente (conocidos como coeficientes de movimiento) que, cuando se multiplican por las funciones básicas, forman una aproximación del vector de movimiento. Los coeficientes de movimiento 724 se hacen pasar desde el bloque de codificación de campo de movimiento 770 al bloque de predicción compensado en movimiento 780. El bloque de

predicción compensado en movimiento 780 también recibe el valor de píxel/subpíxel de la región de prueba candidata de mejor coincidencia de la trama de referencia identificado por el bloque de estimación de movimiento 760. En la figura 7, se muestra que estos valores se hacen pasar a través de la línea 729 desde el bloque de interpolación de subpíxeles bajo demanda 750. En realizaciones alternativas de la invención, los valores de píxeles en cuestión se proporcionan desde el propio bloque de estimación de movimiento 760.

Mediante el uso de la representación aproximada del vector de movimiento generado por el bloque de codificación de campo de movimiento 770 y los valores de píxel/subpíxel de la región de prueba candidata de mejor coincidencia, el bloque de predicción compensado en movimiento 780 produce un macrobloque de valores de píxel predichos. El macrobloque de valores de píxel predichos representa una predicción de los valores de píxel del macrobloque actual generado desde la trama de referencia interpolado. El macrobloque de valores de píxel predichos se hace pasar al combinador 716 donde es sustraído de la trama actual nuevo para producir información de error de predicción 723 desde el macrobloque, como se describió anteriormente.

Los coeficientes del movimiento 724 formados por el bloque de codificación de campo de movimiento también se hacen pasar a la unidad MUX/DMUX 790, en donde se combinan con información de error de predicción 723 del macrobloque en cuestión y posible información de control desde el administrador de control 720 para formar una corriente de video codificada 725 para transmisión a una terminal receptora.

La operación-de un decodificador de video 800 de acuerdo con la invención se describirá ahora. Con referencia a la figura 8, el decodificador 800 comprende una unidad desmultiplexora (MUX/DMUX) 810, que recibe la corriente de video codificada 725 del codificador 700 y la desmultiplexa, un cuantificador inverso 820, un transformador de DCT inverso 830, un bloque de predicción compensado en movimiento 840, un almacén de tramas 850, un combinador 860, un administrador de control 870, una salida 880, un bloque de interpolación de valor de subpíxel de antemano 845 y un bloque de interpolación de subpíxel bajo demanda 890 asociado con el bloque de predicción compensado en movimiento 840. En la práctica, el administrador de control 870 del codificador 800 y el administrador de control 720 del codificador 700 pueden ser el mismo procesador. Este puede ser el caso si el codificador 700 y el decodificador 800 son parte del mismo código de video.

La figura 8 muestra una realización en la cual una combinación de interpolación de valor de subpíxel de antemano y bajo demanda se usa en el decodificador. En otras realizaciones, sólo se usa la interpolación de valor de subpíxel de antemano, en cuyo caso el decodificador 800 no incluye bloque de interpolación de valor de subpíxel bajo demanda 809. En una realización preferida de la invención, no se usa interpolación de valor de subpíxel de antemano en el decodificador y, por lo tanto, el bloque de interpolación de valor de subpíxel de antemano 845 se omite desde la arquitectura del decodificador. Si se realiza tanto la interpolación de valor de subpíxel de antemano y bajo demanda, el decodificador comprende los bloques 845 y 990.

El administrador de control 870 controla la operación del decodificador 800 en respuesta a si se decodifica un INTRA-trama o un INTER-trama. Una señal de control de activación INTRA/INTER, que hace que el decodificador conmute entre modos de decodificación se deriva, por ejemplo, de a información de tipo de imagen provista en la porción de encabezamiento de cada trama de video comprimida recibida del codificador. La señal de control de activación de INTRA/INTER se hace al administrador de control 870 mediante la línea de control 815, junto con otras señales de control de código de video desmultiplexadas de la corriente de video codificada 725 por la unidad MUX/DMUX 810.

Cuando una INTRA-trama se decodifica, la corriente de video codificada 725 es desmultiplexada en macrobloques INTRA-codificados e información de control. No se incluyen vectores de movimiento en la corriente de video codificada 725 para una trama INTRA-codificada. El proceso de decodificación se realiza macrobloque por macrobloque. Cuando la información codificada 723 para un macrobloque es extraída de la corriente de video 725 por la unidad de MUX/DMUX 810, se hace pasar al cuantificador inverso 820. El administrador de control controla el cuantificador inverso 820 para aplicar un nivel adecuado de cuantificación inversa al macrobloque de información codificada, de acuerdo con información de control provista en la corriente de video 725. El macrobloque cuantificado inverso es después inversamente transformado en el transformador de DCT inverso 830 para formar un bloque decodificado de información de imagen. El administrador de control 870 controla el combinados 860 para evitar que cualquier información de referencia sea usada en la decodificación de macrobloque INTRA-codificado. El bloque decodificado de información de imagen se hace pasar a la salida de video 880 del decodificador.

En realizaciones del decodificador que emplean interpolación de valor de subpíxel de antemano, el bloque decodificado de información de imagen (es decir, valores de píxel) producidos como operaciones de cuantificación inversa y transformada inversa realizadas en los bloques 820 y 830 se hace pasar al bloque de interpolación de de antemano 845. Aquí, la interpolación de se realiza de acuerdo con el método de la invención, el grado de interpolación de valor de subpíxel de antemano aplicado que es determinado por los detalles de implementación del decodificador. En realizaciones de la invención en las cuales no se realiza interpolación de valor de subpíxel bajo demanda, el bloque de interpolación de valor de subpíxel de antemano 845 interpola todos los valores de subpíxel. En realizaciones que usan una combinación de interpolación de valor de subpíxel de antemano y bajo demanda, el bloque de interpolación de valor de subpíxel de antemano 845 interpola un cierto subconjunto de valores de

subpíxel. Esto puede comprender, por ejemplo, todos los subpíxeles en posiciones de medio píxel, o una combinación de subpíxeles en posiciones de medio píxel o un cuarto de píxel. En cualquier caso, después de la interpolación de valores de subpíxel de antemano, los valores de subpíxel interpolados se almacenan en el almacén de tramas 850, junto con los valores de píxel decodificados originales. A medida que los macrobloques posteriores son decodificados, interpolados de antemano y almacenados, - una trama decodificado, por lo menos parcialmente interpolado a resolución de subpíxel es progresivamente ensamblado en el almacén de tramas 850 y se vuelve disponible para usarse como una trama de referencia para predicción compensada en movimiento.

En realizaciones del decodificador que no emplean interpolación de valor de subpíxel de antemano, el bloque decodificado de información de imagen (es decir, valores de píxel) producidos como resultado de la cuantificación inversa y operaciones de transformación inversa realizadas en el macrobloque en los bloques 820 y 830 se hace pasar directamente al almacén de tramas 850. A medida que los macrobloques posteriores son decodificados y almacenados, una trama decodificado, que tiene una resolución de píxel de unidad es progresivamente ensamblado en el almacén de tramas 850 y se vuelve disponible para usarse como una trama de referencia para predicción compensada en movimiento.

Cuando una INTER-trama es decodificada, la corriente de video codificada 725 es desmultiplexada en información de error de predicción codificada 723 para cada macrobloque en la trama, coeficientes de movimiento asociados 724 e información de control. Nuevamente, el proceso de decodificación se realiza macrobloque por macrobloque. Cuando la información de error de predicción codificada 723 para un macrobloque es extraída de la corriente de video 725 por la unidad MUX/DMUX 810, se hace pasar al cuantificador inverso 820. El administrador de control 870 controla al cuantificador inverso 820 para aplicar un nivel adecuado de cuantificación inversa al macrobloque de información de error de predicción codificada, de acuerdo con la información de control recibida en la corriente de video 725. El macrobloque cuantificado inverso de información de error de predicción es después inversamente transformado en el transformador de DCT inverso 830 para producir información de error de predicción decodificada para el macrobloque.

Los coeficientes de movimiento 724 asociados con el macrobloque en cuestión son extraídos de la corriente de video 725 por la unidad de MUX/DMUX 810 y se hacen pasar al bloque de predicción compensado en movimiento 840, que reconstruye un vector de movimiento para el macrobloque mediante el uso del mismo modelo de movimiento que el usado para codificar el macrobloque INTER-codificado en el codificador 700. El vector de movimiento reconstruido se aproxima al vector de movimiento originalmente determinado por el bloque de estimación de movimiento 760 del codificador. El bloque de predicción compensado en movimiento 840 del decodificador usa el vector de movimiento reconstruido para identificar la posición de un bloque de valores de píxel/subpíxel en una trama de referencia de predicción almacenado en el almacén de tramas 850. La trama de referencia puede ser, por ejemplo, una DTPA-trama previamente decodificada, o una INTER-trama previamente decodificada. En cualquier caso, el bloque de valores de píxel/subpíxel indicados por el vector de movimiento reconstruido, representa la predicción del macrobloque en cuestión.

El vector de movimiento reconstruido puede indicar a cualquier píxel o subpíxel. Si el vector de movimiento indica que la predicción para el macrobloque actual se forma a partir de valores de píxel (es decir, los valores de píxel en las posiciones de píxel de unidad), estos simplemente pueden ser recuperados del almacén de tramas 850, ya que los valores en cuestión son obtenidos directamente durante la decodificación de cada trama. Si el vector de movimiento indica que la predicción para el macrobloque actual se forma a partir de valores de subpíxel, estos deben ser recuperados del almacén de tramas 850, o calculados en el bloque de interpolación de subpíxel bajo demanda 890. El que los valores de subpíxel deban ser calculados, o simplemente puedan ser recuperados del almacén de tramas, depende del grado de interpolación de valor de subpíxel antemano usado en el decodificador.

En realizaciones del decodificador que no emplean interpolación de valor de subpíxel de antemano, los valores de subpíxel requeridos son calculados todos en el bloque de interpolación de valore de subpíxel bajo demanda 890. Por otra parte, en realizaciones en las cuales todos los valores de subpíxel son interpolados de antemano, el bloque de predicción compensado en movimiento 840 puede recuperar los valores de subpíxel requeridos directamente del almacén de tramas 850. En realizaciones que usan una combinación de antemano y una interpolación de valor de subpíxel bajo demanda, la acción requerida para obtener los valores de subpíxel requeridos depende de qué valores de subpíxel son interpolados de antemano. Al tomar como un ejemplo una realización en la cual todos los valores de subpíxel en posiciones de medio píxel se calculan de antemano, es evidente que si un vector de movimiento reconstruido para un macrobloque indica a un píxel en una posición de unidad o un subpíxel en una posición de medio píxel, todos los valores de píxel o subpíxel requeridos para formar la predicción para el macrobloque están presentes en el almacén de tramas 850 y pueden ser recuperados por el bloque de predicción compensado en movimiento 840. Sin embargo, si el vector de movimiento indica un subpíxel en una posición de un cuarto de píxel, los píxeles requeridos para formar la predicción para los macrobloques no están presentes en el almacén de tramas 850 y por lo tanto se calculan en el bloque de interpolación de valor de subpíxel bajo demanda 890. En este caso, el bloque de interpolación de valor de subpíxel bajo demanda 890 recupera cualquier píxel o subpíxel requerido para formar la interpolación del almacén de tramas 850 y aplica el método de interpolación descrito más adelante. Los valores de subpíxel calculados en el bloque de interpolación de valor de subpíxel bajo demanda 890 se hacen pasar al bloque de predicción compensado en el bloque de predicción 840.

Una vez que se ha obtenido una predicción de un macrobloque, la predicción (es decir, un macrobloque de valores de píxel predicho) se hace pasar desde el bloque de predicción compensado en movimiento 840 hasta el combinador 860 donde se combina con la información de error de predicción decodificada para que el macrobloque forme un bloque de imagen reconstruida que, a su vez, se hace pasar a la salida de video 880 del decodificador.

5 Cabe apreciar que en implementaciones prácticas de codificador 700 y decodificador 800, el grado al cual las tramas son valores de subpíxel interpolados de antemano, y por lo tanto la cantidad de interpolación de valor de subpíxel bajo demanda que se realiza se puede escoger de acuerdo con o determinarse por la implementación de hardware del codificador de video 700, o el ambiente en el cual se ha de usar. Por ejemplo, si la memoria disponible para el
 10 codificador de video es limitada, o la memoria debe ser reservada para otras funciones, es apropiado limitar la cantidad de interpolación de valor de subpíxel de antemano que se realiza. En otros casos, donde el microprocesador que realiza la operación de codificación de video tiene capacidad de procesamiento limitada, por ejemplo, el número de operaciones por segundo que se pueden ejecutar es comparativamente baja, es más apropiada para restringir la cantidad de interpolación de valor de subpíxel bajo demanda que se realiza. En un ambiente de comunicaciones móviles, por ejemplo, cuando la funcionalidad de codificación y decodificación de video se incorpora en un teléfono móvil o una terminal inalámbrica similar para comunicación con una red de telefonía móvil, tanto la memoria como la potencia de procesamiento pueden ser limitadas. En este caso, una combinación de interpolación de valor de subpíxel de antemano y bajo demanda puede ser la mejor elección para obtener una implementación eficiente en el codificador de vídeo. En el decodificador de video 800, el uso de un valor de subpíxel de antemano generalmente no es preferido, ya que normalmente da por resultado el cálculo de muchos valores de subpíxel que realmente no se usan en el proceso de decodificación. Sin embargo, cabe apreciar que aunque se pueden usar diferentes cantidades de interpolación de antemano y bajo demanda en el codificador y decodificador para optimizar la operación de cada uno, tanto el codificador como el decodificador se pueden implementar para usar la misma división entre interpolación de valor de subpíxel de antemano y bajo demanda.

25 Aunque la descripción anterior no describe la construcción de tramas bidireccionalmente predichas (tramas B) en el codificador 700 y el decodificador 800, se debe entender que en realizaciones de la invención esa capacidad se puede proveer. El proveer esa capacidad se considera dentro de la capacidad de un experto en la técnica.

30 Un codificador 700 o un decodificador 800 de conformidad con la invención pueden utilizarse mediante el uso de hardware o software, o mediante el uso de una combinación adecuada de ambos. Un codificador o decodificador implementado en software puede ser, por ejemplo, un programa separado o un bloque de construcción de software que puede ser usado por varios programas. En la descripción anterior y en los dibujos, los bloques funcionales están representados como unidades separadas, pero la funcionalidad de estos bloques puede ser implementada, por
 35 ejemplo, en una unidad de programa de software.

El codificador 700 y decodificador 800 se pueden combinar para formar un códec de video que tenga funcionalidad de codificación como de decodificación. Además de implementarse en una terminal multimedia, el códec también se
 40 puede implementar en una red. Un códec de acuerdo con la invención puede ser un programa de ordenador o un elemento de programa de ordenador, o puede implementarse por lo menos parcialmente mediante el uso de hardware.

El método de interpolación de subpíxeles usado en el codificador 700 y el decodificador 800 de acuerdo con la invención se describirán ahora con detalle. El método primero será introducido a un nivel conceptual general y después se describirán dos realizaciones preferidas. En la primera realización preferida, la incorporación de valor de subpíxeles se realiza a una resolución de 1/4 de píxel y en el segundo método es una resolución de 1/8 de píxel.

45 Debe indicarse que la interpolación debe producir valores idénticos en el codificador y el decodificador, pero su implementación debe ser optimizada para ambas entidades por separado. Por ejemplo, en un codificador de acuerdo con la primera realización de la invención en el cual la interpolación de valores de subpíxeles se realiza a una resolución de 1/4 de píxel, lo más eficiente es calcular píxeles de una resolución de 1/2 de antemano y calcular valores para subpíxeles de una resolución de 1/4 de una manera sobre demanda, sólo cuando sea necesario durante la estimación de movimiento. Esto tiene el efecto de limitar el uso de memoria mientras se mantiene la complejidad/carga computacional a un nivel aceptable. El decodificador, por otra parte, es ventajoso no calcular previamente ninguno de los subpíxeles. Por lo tanto, cabe apreciar que una realización preferida del decodificador no incluye bloque de interpolación de valor de subpíxel de antemano 845 y toda la interpolación de valor de subpíxel se realiza en un bloque de interpolación de valor de subpíxel bajo demanda 890.

50 En la descripción del método de interpolación que se proporciona más adelante, se hacen referencias a las posiciones de píxel ilustradas en la figura 14a. En esta figura, los píxeles marcados con la letra A representan píxeles originales (es decir, píxeles que residen en posiciones horizontales y verticales). Los píxeles marcados con otras letras representan subpíxeles que han de ser interpolados. La descripción siguiente se añadirá a las convenciones previamente introducidas referentes a la descripción de posición de píxeles y subpíxeles.

65 A continuación, se describirán las etapas requeridas para interpolar todas las posiciones de subpíxeles:

Los valores para los subpíxeles de resolución 1/2 marcados con la letra b se obtienen al calcular primero un valor b intermedio con el uso de un filtro de orden K-ésimo, de acuerdo con:

$$b = \sum_{i=1}^K x_i A_i \quad (9)$$

5 donde X_i es un vector de coeficientes de filtro, A_i es un vector correspondiente de valores de píxel originales A situados en posiciones horizontales de unidad y verticales de unidad y K es un entero que define el orden del filtro. Por lo tanto, la ecuación 9 se puede expresar nuevamente como:

$$10 \quad b = x_1 A_1 + x_2 A_2 + x_3 A_3 + \dots + x_{K-1} A_{K-1} + x_K A_K \quad (10)$$

Los valores de los coeficientes de filtro X_i y el orden del filtro K pueden variar de una realización a otra. Igualmente, valores de coeficientes diferentes se pueden usar en el cálculo de diferentes subpíxeles dentro de una realización. En otras realizaciones, los valores de coeficientes de filtro X_i y el orden del filtro pueden depender de cuál de los subpíxeles de resolución b de 1/2 es interpolada. Los píxeles A_i están dispuestos simétricamente con respecto al subpíxel b de resolución de 1/2 que es interpolada y son los vecinos más cercanos de ese subpíxel. En el caso del subpíxel b de resolución de 1/2 situado en una posición horizontal de medio y una posición vertical de unidad, los píxeles A_i están dispuestos horizontalmente con respecto a b (como se muestra en la figura 14b). Si el subpíxel b de resolución 1/2 situado en una posición horizontal de unidad y una posición vertical de medio es interpolado, los subpíxeles A_i están dispuestos verticalmente con respecto a b (como se muestra en la figura 14c).

Un valor final para el subpíxel b de resolución de 1/2 se calcula al dividir el valor b intermedio entre una escala constante, que lo trunca para obtener un número entero y ajustando el resultado para que quede en el intervalo $[0, 2^n - 1]$. En realizaciones alternativas de la invención, el redondeo se puede realizar en lugar del truncamiento. Preferiblemente, la escala constante se escoge para que sea igual a la suma de los coeficientes de filtro X_i .

Un valor para el subpíxel de resolución de 1/2 marcado con la letra c también se obtiene al calcular un valor intermedio c mediante el uso de un filtro de orden M -ésimo, de acuerdo con:

$$30 \quad c = \sum_{i=1}^M y_i b_i \quad (11)$$

donde y_i es un vector de coeficientes de filtro, b_i es un vector correspondiente de valores intermedios b_i en la dirección horizontal o vertical, es decir:

$$35 \quad c = y_1 b_1 + y_2 b_2 + y_3 b_3 + \dots + y_{M-1} b_{M-1} + y_M b_M \quad (12)$$

Los valores de los coeficientes de filtro y_i y el orden del filtro M pueden variar de una realización a otra. Igualmente, los valores de coeficientes diferentes se pueden usar en el cálculo de diferentes subpíxeles dentro de una realización. Preferiblemente, los valores de b son valores intermedios para subpíxeles b de resolución de 1/2 que están dispuestos simétricamente con respecto al subpíxel c de resolución de 1/2 y son los vecinos más cercanos del subpíxel c . En una realización de la invención, los subpíxeles b de resolución de 1/2 están dispuestos horizontalmente con respecto al subpíxel c , en una realización alternativa, están dispuestos verticalmente con respecto al subpíxel c .

Un valor final del subpíxel c de resolución de 1/2 se calcula al dividir el valor c intermedio entre una escala constante, al truncarlo para obtener un número entero y ajustando el resultado para que quede en el intervalo $[0, 2^n - 1]$. En realizaciones alternativas de la invención, se puede realizar redondeo en lugar de truncamiento. Preferiblemente, la escala constante es igual a la escala₁ * escala₂.

Debe indicarse que el uso de valores b intermedios en la dirección horizontal conduce al mismo resultado que si se usaran valores b intermedios en la dirección vertical.

Existen dos alternativas para interpolar valores para los subpíxeles de resolución de 1/4 marcados con la letra h . Ambas implican interpolación lineal a lo largo de una línea diagonal que enlaza subpíxeles de resolución de 1/2 vecinos al subpíxel h de resolución de 1/4 que es interpolado. En una primera realización, un valor para subpíxel h se calcula al promediar los valores de los dos subpíxeles b de resolución de 1/2 más cercanos al subpíxel h . En una segunda realización, un valor para el subpíxel h se calcula al promediar los valores del píxel A más cercano y el subpíxel c de resolución de 1/2 más cercano. Cabe apreciar que esto proporciona la posibilidad de usar diferentes combinaciones de interpolaciones diagonales para determinar los valores para subpíxeles h dentro de los confines de diferentes grupos de 4 píxeles A de imagen. Sin embargo, también cabe apreciar que la misma combinación se debe usar tanto en el codificador como en el decodificador para producir resultados de interpolación idénticos. La

figura 15 ilustra 4 posibles elecciones de interpolación diagonal para subpíxeles h en grupos adyacentes de 4 píxeles dentro de una imagen. Las simulaciones en el ambiente TML han verificado que ambas realizaciones dan por resultado una eficiencia de compresión similar. La segunda realización tiene mayor complejidad, ya que el cálculo del subpíxel c requiere el cálculo de varios valores intermedios. Por lo tanto, se prefiere la primera realización.

Los valores para subpíxeles de resolución de 1/4 marcados con la letra d y g se calculan a partir de los valores de sus vecinos horizontales más cercanos mediante el uso de interpolación lineal. En otras palabras, un valor para el subpíxel de resolución de 1/4 se obtiene al promediar valores de sus vecinos horizontales más cercanos, el píxel A de imagen original y el subpíxel b de resolución de 1/2. De manera similar, un valor para un subpíxel g de resolución de 1/4 se obtiene al sacar el promedio de sus dos vecinos horizontales más cercanos, los subpíxeles b y c de resolución de 1/2.

Los valores para subpíxeles de resolución de 1/4 marcados con la letra e, f e i se calculan a partir de los valores de sus vecinos más cercanos en su dirección vertical mediante el uso de interpolación lineal. De manera más específica, un valor para subpíxel e de resolución de 1/4 se obtiene al promediar los valores de sus dos vecinos verticales más cercanos, el píxel A de imagen original y el subpíxel b de resolución de 1/2. De manera similar, un valor para el subpíxel f de resolución de 1/4 se obtiene al sacar el promedio de sus dos vecinos verticales más cercanos, los subpíxeles b y c de resolución de 1/2. En una realización de la invención, un valor para el subpíxel i de resolución de 1/4 se obtiene de una manera idéntica a la que se acaba de describir en conexión con el subpíxel f de resolución de 1/4. Sin embargo, en una realización alternativa de la invención, y en común con los modelos de prueba TML5 y TML6 de H.26 previamente descritos, el subpíxel i de resolución de 1/4 se determina mediante el uso de los valores de los cuatro píxeles de imagen originales más cercanos de acuerdo con $(A_1 + A_2 + A_3 + A_4 + 2)/4$.

Debe indicarse también que en todos los casos en donde un promedio que implique valores de píxel y/o subpíxel se determina, el promedio se puede formar de cualquier manera apropiada. Por ejemplo, el valor para el subpíxel d de resolución de 1/4 se puede definir como $d = (A + b)/2$ o como $d = (A + b + 1)/2$. La adición de 1 a la suma de valores para el píxel A y el subpíxel b de resolución de 1/2 tiene el efecto de causar cualquier operación de redondeo o truncamiento posteriormente aplicada para redondear o truncar el valor para d al siguiente valor entero más alto. Esto es cierto para cualquier suma de valores enteros y se puede aplicar a cualquiera de las operaciones de promedio realizadas de acuerdo con el método de la invención para controlar los efectos de redondeo o truncamiento.

Debe indicarse que el método de interpolación de valor de subpíxel de acuerdo con la invención proporciona ventajas sobre cada uno de TML5 y TML6.

A diferencia de TML5, en el cual los valores de algunos de los subpíxeles de resolución de 1/4 dependen de valores previamente interpolados obtenidos para otros subpíxeles de resolución de 1/4, y en el método de acuerdo con la invención, todos los subpíxeles de resolución de 1/4 se calculan a partir de píxeles de imagen original o posiciones de subpíxel de resolución de 1/2 mediante el uso de interpolación lineal. Por lo tanto, la reducción en precisión de esos valores de subpíxel de resolución de 1/4 se produce en TML5 debido al truncamiento y al ajuste intermedio de los otros subpíxeles de resolución de 1/4 a partir de los cuales se calculan, no tiene lugar en el método de acuerdo con la invención. En particular, con referencia a la figura 14a, los subpíxeles h de resolución de 1/4 (y el subpíxel i en una realización de la invención) se interpolan diagonalmente para reducir la dependencia de los otros subpíxeles de 1/4. Además, en el método de acuerdo con la invención, el número de cálculos (y por lo tanto el número de ciclos de procesador) requeridos para obtener un valor para aquellos subpíxeles de resolución de 1/4 en el decodificador se reducen en comparación con TML5. Además, el cálculo de cualquier valor de subpíxel de resolución de 1/4 requiere un número de cálculos que es sustancialmente similar al número de cálculos requeridos para determinar cualquier otro valor de subpíxel de resolución 1/4. De manera más específica, en una situación en donde los valores de subpíxel de resolución de 1/2 requeridos ya están disponibles, por ejemplo, han sido calculados de antemano, el número de cálculos requeridos para interpolar un valor de subpíxel de resolución de 1/4 a partir de los valores de subpíxel de resolución de 1/2 precalculados es el mismo que el número de cálculos requeridos para calcular cualquier otro valor de subpíxel de resolución de 1/4 a partir de los valores de subpíxel de resolución de 1/2 disponibles.

En comparación con TML6, el método de acuerdo con la invención no requiere aritmética de precisión alta para usarse en el cálculo de todos los subpíxeles. Específicamente, puesto que todos los valores de subpíxel de resolución de 1/4 se calculan a partir de píxeles de imagen original o valores de subpíxel de resolución de 1/2 usando interpolación lineal, se puede usar aritmética de precisión inferior en su interpolación. Consecuentemente, implementaciones de hardware del método de la invención, por ejemplo en un Circuito Integrado Específico de Aplicación, (ASIC), el uso de aritmética de precisión más baja reduce el número de componentes (por ejemplo, compuertas) que deben ser dedicadas al cálculo de valores de subpíxel de resolución de 1/4. Esto, a su vez, reduce el área global de silicio que se debe dedicar a la función de interpolación. Puesto que la mayoría de los subpíxeles son de hecho subpíxeles de resolución de 1/4 (12 de los 15 subpíxeles ilustrados en la figura 14a), la ventaja proporcionada por la invención a este respecto es particularmente significativa. En implementaciones de software, donde la interpolación de subpíxel se realiza mediante el uso del conjunto de instrucciones estándares de una

unidad de procesador central (CPU) para propósitos generales o mediante el uso de un Procesador de Señal Digital (DSP), una reducción en la precisión de la aritmética requerida generalmente conduce a un incremento en la velocidad a la cual se pueden realizar los cálculos. Esto es particularmente ventajoso en implementaciones de "bajo coste", en las cuales es conveniente usar una CPU para propósitos generales más que cualquier forma de ASIC.

El método de acuerdo con la invención proporciona ventajas adicionales comparadas con TML5. Como se mencionó anteriormente, en el decodificador sólo una de las 15 posiciones de subpíxel se requiere en cualquier tiempo dado, a saber, aquella que se indica por información de vector de movimiento recibida. Por lo tanto, es ventajoso si el valor de un subpíxel en cualquier posición de subpíxel puede calcularse con el número mínimo de etapas que dan como resultado un valor correctamente interpolado. El método de acuerdo con la invención provee esta capacidad. Como se menciona en la descripción detallada anteriormente dada, el subpíxel c de resolución de $1/2$ se puede interpolar al filtrar ya sea en la dirección vertical u horizontal del mismo valor que se obtiene para c independientemente de que filtro horizontal o vertical se use. El decodificador por lo tanto puede sacar ventaja de esta propiedad al calcular valores para subpíxeles f y g de resolución de $1/4$, de tal manera que reduzca al mínimo el número de operaciones requeridas para obtener los valores requeridos. Por ejemplo, si el decodificador requiere un valor para subpíxel f de resolución de $1/4$, el subpíxel c de resolución de $1/2$ debe ser interpolado en la dirección vertical. Si se requiere un valor para el subpíxel g de resolución de $1/4$, es ventajoso interpolar un valor para c en la dirección horizontal. Por lo tanto, en general, se puede decir que el método de acuerdo con la invención provee flexibilidad en la forma en la cual los valores son derivados de ciertos subpíxeles de resolución de $1/4$. No se proporciona tal flexibilidad en TML5.

Dos realizaciones específicas se describirán ahora con detalle. La primera representa una realización preferida para calcular subpíxeles hasta con una resolución de de píxel, mientras que en la segunda, el método de acuerdo con la invención se extiende al cálculo de valores para subpíxeles que tienen una resolución de píxel de hasta $1/8$. Para ambas realizaciones se proporciona una comparación entre la flexibilidad/carga computacional que resulta de usar el método de acuerdo con la invención y que resultaría del uso de los métodos de interpolación de acuerdo con TML5 y TML6 en circunstancias equivalentes.

La realización preferida para interpolar subpíxeles a resolución .de píxel de $1/4$ se describirá con referencia a las figuras 14a, 14b y 14c. A continuación, se supondrá que todos los píxeles de imagen y valores interpolados finales para subpíxeles están representados con 8 bits.

Cálculo de subpíxeles de resolución de $1/2$ en i) posición horizontal de media unidad y posición vertical de unidad ii) posición horizontal de unidad y posición vertical de media unidad.

1. Un valor para el subpíxel en una posición horizontal de media unidad y vertical de unidad, es decir, subpíxel b de resolución de $1/2$ en la figura 14a, se obtiene al calcular primero el valor intermedio $b = (A_1 - 5A_2 + 20A_3 + 20A_4 - 5A_5 + A_6)$ mediante el uso de valores de seis píxeles (A_1 a A_6) que están situados en posiciones horizontales de unidad y verticales de unidad ya sea en la fila o en la columna de píxeles que contienen b y que están dispuestos simétricamente alrededor de b , como se muestra en las figuras 14b y 14c. Un valor final para su píxel b de resolución de $1/2$ se calcula como $(b + 16)/32$ donde el operador $/$ indica división con truncamiento. El resultado es ajustado para quedar en el intervalo $[0, 255]$.

Cálculo de subpíxeles de resolución de $1/2$ en la posición horizontal de media unidad y posición vertical de media unidad.

2. Un valor para el subpíxel en la posición horizontal de media unidad y posición vertical de media unidad, es decir subpíxel c de resolución de $1/2$ en la figura 14a, se calcula como $c = (b_1 - 5b_2 + 20b_3 + 20b_4 - 5b_5 + b_6 + 512)/1024$ mediante el uso de valores intermedios b para los seis subpíxeles de resolución de $1/2$ más cercanos que están situados ya sea en la fila o la columna de subpíxeles que contienen c y que están dispuestos simétricamente alrededor de c . Nuevamente, el operador $/$ indica división con truncamiento y el resultado se ajusta para quedar en el intervalo de $[0, 255]$. Como se explicó anteriormente, el uso de valores intermedios para subpíxeles b de resolución .de $1/2$ en la dirección horizontal conduce al mismo resultado que el uso de valores intermedios b para subpíxeles b de resolución de $1/2$ en la dirección vertical. Por lo tanto, en un codificador de acuerdo con la invención, la dirección para interpolar subpíxeles b de resolución de $1/2$ se puede escoger de acuerdo con un modo de implementación preferido. En un decodificador de acuerdo con la invención, la dirección para interpolar subpíxeles b se escoge de acuerdo con cuáles subpíxeles, si acaso hay alguno, de resolución de $1/4$ serán interpolados mediante el uso del resultado obtenido para el subpíxel c de resolución de $1/2$.

Cálculo de subpíxeles de resolución de $1/4$ en i) posición horizontal de un cuarto de unidad y posición vertical de unidad; ii) posición horizontal de un cuarto de unidad y posición vertical de media unidad; iii) posición horizontal de unidad y posición vertical de un cuarto de unidad; y iv) posición horizontal de media unidad y posición vertical de un cuarto de unidad.

3. Los valores para subpíxeles de resolución de $1/4$, situados en posición horizontal de un cuarto de unidad y posición vertical de unidad se calculan de acuerdo con $d = (A + b)/2$ mediante el uso del píxel A de imagen original más cercano y el subpíxel b de resolución de $1/2$, más cercano en la dirección horizontal. De manera similar, los valores para subpíxeles g de resolución de $1/4$, situados en una posición horizontal de un cuarto de unidad y una posición vertical de media unidad se calculan de acuerdo con $g = (b + c)/2$ mediante el uso de los dos subpíxeles de resolución de $1/2$ más cercanos en la dirección horizontal. De manera similar, los valores

para subpíxeles e de resolución de 1/4, situados en una posición horizontal de unidad y una posición vertical de un cuarto de unidad, se calculan de acuerdo con $e = (A + b)/2$ mediante el uso del píxel A de imagen original más cercano y el subpíxel b de resolución de 1/2 más cercano en la dirección vertical. Los valores para subpíxeles f de resolución de 1/4, situados en una posición horizontal de media unidad y una posición vertical de un cuarto de unidad, se determinarla partir de $f = (b + c)/2$ mediante el uso de los dos subpíxeles de resolución de 1/2 más cercanos en la dirección vertical. En todos los casos, el operador / indica división con truncamiento.

Cálculo de subpíxeles de resolución de 1/4 en una posición horizontal de un cuarto de unidad y una posición vertical de un cuarto de unidad.

4. Los valores para subpíxeles h de resolución de 1/4 situados en una posición horizontal de un cuarto de unidad y una posición vertical de un cuarto de unidad se calculan de acuerdo con $h = (b_1 + b_2)/2$, mediante el uso de los dos subpíxeles b de resolución de 1/2 más cercanos en la dirección diagonal. Nuevamente, el operador / indica división con truncamiento.

5. Un valor para el subpíxel de resolución de 1/4 marcado con la letra i se calcula a partir de $i = (A_1 + A_2 + A_3 + A_4 + 2)/4$ mediante el uso de los cuatro subpíxeles A originales más cercanos. Una vez más, el operador / indica división con truncamiento.

Un análisis de la complejidad computacional de la primera realización preferida de la invención se presentará ahora.

En el codificador, es probable que los mismos valores de subpíxel se calculen múltiples veces. Por lo tanto, y como se explicó anteriormente, la complejidad del codificador se puede reducir al calcular previamente todos los valores de subpíxel y almacenarlos en memoria. Sin embargo, esta solución incrementa el uso de memoria por un margen. En una realización preferida de la invención, en la cual la precisión de vector de movimiento es resolución de píxel de 1/4 tanto en la dimensión horizontal como en la vertical, el almacenamiento de valores de subpíxel calculados previamente para la imagen completa requiere 16 veces la memoria requerida para almacenar la imagen no interpolada original. Para reducir el uso de memoria, todos los subpíxeles de resolución de 1/2 se pueden interpolar de antemano y los subpíxeles de resolución de 1/4 se pueden calcular bajo demanda, es decir, sólo cuando sean necesarios. De acuerdo con el método de la invención, la interpolación bajo demanda de valores para subpíxeles de resolución de 1/4 sólo requiere la interpolación lineal de subpíxeles de resolución de 1/2. Cuatro veces la memoria de imagen original se requiere para almacenar subpíxeles de resolución de 3 precalculados, ya que sólo 8 bits son necesarios para representarlos.

Sin embargo, si la misma estrategia de calcular previamente todos los subpíxeles de resolución de 1/2 al usar interpolación de antemano se usa junto con el esquema de interpolación director de TML6, los requerimientos de memoria incrementan hasta 9 veces la memoria requerida para almacenar la imagen no interpolada original. Esto resulta del hecho de que un número mayor de bits se requiere para almacenar los valores intermedios de alta precisión asociados con cada subpíxel de resolución de 1/2 en TML6. Además, la complejidad de interpolación de subpíxel durante la estimación de movimiento es mayor en TML6, ya que el escalado y el ajuste se tienen que realizar para cada posición de subpíxel de 1/2 y 1/4.

A continuación, la complejidad del método de interpolación de valor de subpíxel de acuerdo con la invención, cuando se aplica en un decodificador de video, se compara con el de los esquemas de interpolación usados en TML5 y TML6. En todo el análisis que sigue, se supone que en cada método la interpolación de cualquier valor de subpíxel se realiza mediante el uso únicamente del número mínimo de etapas requeridas para obtener un valor correctamente interpolado. Además, se supone que cada método es implementado de una manera basada en bloques, es decir, valores intermedios comunes para todos los subpíxeles para ser interpolados en el bloque N x M particular se calculan sólo una vez. Un ejemplo ilustrativo se proporciona en la figura 16. Con referencia a la figura 16, se puede ver que para calcular un bloque 4 x 4 de subpíxeles c de resolución de 1/2 primero se calcula un bloque de 9 x 4 de subpíxeles b de resolución de 1/2.

En comparación con el método de interpolación de valor de subpíxel en TML5, el método de acuerdo con la invención tiene una complejidad computacional más baja por las siguientes razones:

1. A diferencia del esquema de interpolación de valor de subpíxel usado en TML5, de acuerdo con el método de la invención, un valor para subpíxel c de resolución de 1/2 se puede obtener al filtrar ya sea en la dirección vertical u horizontal. Por lo tanto, para reducir el número de operaciones, el subpíxel c de resolución de 1/2 se puede interpolar en la dirección vertical si se requiere un valor para subpíxel f de resolución de 1/4 y en la dirección horizontal si se requiere un valor para un píxel g de subpíxel de resolución de 1/4. Como un ejemplo, la figura 17 muestra todos los valores de subpíxel de resolución de 1/2 que se deben calcular para interpolar valores para subpíxeles g de resolución de 1/4 en un bloque de imagen definido por píxeles de imagen originales de 4 x 4 mediante el uso del método de interpolación TML5 (figura 17a) y mediante el uso del método de acuerdo con la invención (figura 17b). En este ejemplo, el método de interpolación de valor de subpíxel de acuerdo con TML5 requiere un total de 88 subpíxeles de resolución de 1/2 para ser interpolada, mientras que el método de acuerdo con la invención requiere el cálculo de 72 subpíxeles de resolución de 1/2. Como se puede ver a partir de la figura 17b, de acuerdo con la invención, los subpíxeles c de resolución de 1/2 son interpolados en la dirección horizontal para reducir el número de cálculos requeridos.

2. De acuerdo con el método de la invención, el subpíxel h de resolución de 1/4 se calcula por interpolación lineal a partir de sus dos subpíxeles de resolución de 1/2 vecinos más cercanos en la dirección diagonal. Los números respectivos de subpíxeles de resolución de que se deben calcular para obtener valores para subpíxeles h de resolución de 1/4 dentro de un bloque de 4 x 4 de píxeles de imagen original mediante el uso del método de interpolación de valor de subpíxel de acuerdo con TML5 y el método de acuerdo con la invención se muestra en las figuras 18(a) y 18(b), respectivamente. Al usar el método de acuerdo con TML5, es necesario interpolar un total de 56 subpíxeles de resolución de 1/2, mientras que de acuerdo con el método de la invención, es necesario interpolar 40 subpíxeles de resolución de 1/2.

La tabla 1 resume las complejidades de decodificador de los tres métodos de interpolación de valor de subpíxel considerados aquí, que de acuerdo con TML5, el método de interpolación directo usado en TML6 y el método de acuerdo con la invención. La complejidad se mide en términos del número de operaciones de interpolación de filtro de seis ramas y lineal realizadas. Se supone que la interpolación de subpíxel i de resolución de 1/4 se calcula de acuerdo con $i = (A_1 + A_2 + A_3 + A_4 + 2)/4$ que es interpolación bilineal y efectivamente comprende dos operaciones de interpolación lineal. Las operaciones necesarias para interpolar los valores del subpíxel con un bloque de 4 x 4 de los píxeles de imagen original se listan para cada una de las 15 posiciones de subpíxel, por conveniencia de referencia, están numeradas de acuerdo con el esquema mostrado en la figura 19. Con referencia a la figura 19, la posición 1 es la posición en un píxel A de imagen original y las posiciones 2 a 16 son posiciones de subpíxel. La posición 16 es la posición de subpíxel i de resolución de 1/4. Para calcular el número promedio de operaciones se ha supuesto que la probabilidad de un vector de movimiento que indica a cada posición de subpíxel es la misma. La complejidad promedio por lo tanto es el promedio de las 15 sumas calculadas para cada posición de subpíxel y la posición de píxel completo individual.

Tabla 1: Complejidad de interpolación de subpíxel de resolución de 1/4 en TML5, TML6 y el método de acuerdo con la invención

Posición	TML5		TML6		Método de la invención	
	lineal	6 ramas	Lineal	6 ramas	lineal	6 ramas
1	0	0	0	0	0	0
3,9	0	16	0	16	0	16
2,4,5,13	16	16	0	16	16	16
11	0	52	0	52	0	52
7,15	16	52	0	52	16	52
10,12	16	68	0	52	16	52
6,8,14	48	68	0	52	16	32
16	32	0	32	0	32	0
Promedio	19	37	2	32	13	28,25

Se puede ver a partir de la Tabla 1 que el método de acuerdo con la invención requiere menos operaciones de filtro de seis ramas que el método de interpolación de valor de subpíxel de acuerdo con TML6 y sólo unas cuantas operaciones de interpolación lineal adicionales. Puesto que las operaciones de filtro de seis ramas son mucho más complejas que las operaciones de interpolación lineal, la complejidad de los dos métodos es similar. El método de interpolación de valor de subpíxel de acuerdo con TML5 tiene una complejidad considerablemente más alta.

La realización preferida para interpolar subpíxeles hasta con una resolución de píxel de 1/8 se describirá ahora con referencia a las figuras 20, 21 y 22.

La figura 20 presenta la nomenclatura usada para describir píxeles, subpíxeles de resolución de 1/2, subpíxeles de resolución de 1/4 y subpíxeles de resolución de 1/8 en esta aplicación extendida al método de acuerdo con la invención.

1. Los valores para subpíxeles de resolución de 1/2 y resolución de 1/4 marcados con las letras b^1 , b^2 y b^3 en la figura 20 se obtienen al calcular primero los valores intermedios $b^1 = (-3A_1 + 12A_2 - 37A_3 + 229A_4 + 71A_5 + 21A_6 + 6A_7 - A_8)$; $b^2 = (-3A_1 + 12A_2 - 39A_3 + 158A_4 + 158A_5 - 39A_6 + 12A_7 - 3A_8)$; y $b^3 = (-A_1 + 6A_2 - 21A_3 + 71A_4 + 229A_5 - 37A_6 + 13A_7 - 3A_8)$; mediante el uso de los valores de los ocho píxeles de imagen más cercanos (A_1 a A_8) situados en una posición horizontal de unidad y una posición vertical de unidad ya sea en la fila o la columna que contiene b^1 , b^2 y b^3 y dispuestos simétricamente alrededor del subpíxel b^2 de resolución de 1/2. Las asimetrías en sus coeficientes usados para obtener valores intermedios b^1 y b^3 explican el hecho, de que los píxeles A_1 a A_8 no están siméricamente ubicados con respecto a los subpíxeles b^1 y b^3 de resolución de 1/4. Los valores finales para los subpíxeles b^i , $i = 1, 2, 3$ se calculan de acuerdo con $b^i = (b^i + 128)/256$ donde el operador / indica división con truncamiento. El resultado se ajusta para quedar en el intervalo [0, 255].

2. Los valores para los subpíxeles de resolución de 1/2 y resolución de 1/4 marcados con c^{ij} , $i, j = 1, 2, 3$, se calculan de acuerdo con $c^{1j} = (-3b_1^j + 12b_2^j - 37b_3^j + 229b_4^j + 71b_5^j - 21b_6^j + 6b_7^j - b_8^j + 32768)/65536$, $c_2^j = (-3b_1^j + 12b_2^j - 39b_3^j + 158b_4^j + 158b_5^j + 39b_6^j + 12b_7^j - 3b_8^j + 32768)/65536$ y $c_3^j = (-b_1^j + 6b_2^j - 21b_3^j + 71b_4^j - 229b_5^j$

5 $- 37b_6^j + 13b_7^j - 3b_8^j + 32768)/65536$ mediante el uso de los valores intermedios b^1, b^2 y b^3 calculados para los ocho subpíxeles más cercanos (b_1^j a b_8^j) en la dirección vertical, los subpíxeles b_3 están situados en la columna que comprende los subpíxeles de resolución de y y resolución de $1/2$ que son interpolados y están dispuestos simétricamente alrededor del subpíxel c^{2j} de resolución de $1/4$. Las asimetrías en los coeficientes de filtro usados para obtener valores para subpíxeles c^{1j} y c^{3j} explican el hecho de que los subpíxeles b_1^j a b_8^j no están simétricamente ubicados con respecto a los subpíxeles c^{1j} y c^{3j} de resolución de $1/4$. Una vez más, el operador / indica división con truncamiento. Antes de que los valores interpolados para los subpíxeles sean almacenados en la memoria de trama son ajustados para quedar en $[0, 255]$. En una realización alternativa de la invención, los subpíxeles de resolución de $1/2$ y de resolución de $1/4$ se calculan mediante el uso de una manera análoga de valores intermedios b^1, b^2 y b^3 en la dirección horizontal.

10
15 3. Los valores para subpíxeles de resolución de $1/8$ marcados con la letra d se calculan usando interpolación lineal a partir de los valores de su píxel de imagen vecino más cercano, subpíxeles de resolución de $1/2$ o resolución de $1/4$ en la dirección horizontal o vertical. Por ejemplo, el subpíxel d de resolución de $1/8$ que está más a la izquierda superior se calcula de acuerdo con $d = (A + b^1 + 1)/2$. Igual que antes, el operador / indica división con truncamiento.

20 4. Los valores para subpíxeles de resolución de $1/8$ marcados con las letras e y f se calculan al usar la interpolación lineal a partir de los valores de píxeles de imagen, subpíxeles de resolución de $1/2$ o de resolución de $1/4$ en la dirección diagonal. Por ejemplo, con referencia a la figura 20, el subpíxel e de resolución de $1/8$ más a la izquierda superior se calcula de acuerdo con $e = (b^1 + b^1 + 1)/2$. La dirección diagonal que se ha de usar en la interpolación de cada subpíxel de resolución de $1/8$ en una primera realización de la invención, de aquí en adelante referido como "método preferido 1", se indica en la figura 21(a). Los valores para subpíxeles de resolución de $1/8$ marcados con la letra g se calculan de acuerdo con $g = (A + 3c^{22} + 3)/4$. Como siempre, el operador / indica división con truncamiento. En una realización alternativa de la invención, de aquí en adelante referida como "método preferido 2", la complejidad computacional se reduce además por interpolación de subpíxeles f de resolución de $1/8$ mediante el uso de interpolación a partir de subpíxeles b_2 de resolución de $1/8$, es decir, de acuerdo con la relación $f = (3b^2 + b^2 + 2)/4$. El subpíxel b^2 que es más cercano a f se multiplica por 3. El esquema de interpolación diagonal usado en esta realización alternativa de la invención se ilustra en la figura 21(b). En realizaciones alternativas adicionales, se pueden contemplar esquemas de interpolación diagonal diferentes.

35 Debe indicarse que en todos los casos en donde un promedio que implique valores de píxel y/o subpíxel se usa en la determinación de subpíxeles de resolución de $1/8$, el promedio se puede formar de cualquier manera apropiada. La adición de 1 a la suma de valores usados para calcular un promedio tiene el efecto de calcular cualquier operación de redondeo o truncamiento posteriormente aplicado para redondear o truncar el promedio en cuestión al siguiente valor entero más alto. En realizaciones alternativas de la invención, la adición de 1 no se usa.

40 Como en el caso de interpolación de valor de subpíxel a resolución de píxel de $1/4$ previamente descrito, los requerimientos de memoria en el codificador se pueden reducir al calcular previamente sólo una parte de los valores de subpíxel que han de ser interpolados. En el caso de interpolación de valor de subpíxel a resolución de píxel de $1/8$, es ventajoso calcular todos los subpíxeles de resolución de $1/2$ y de resolución de $1/4$ de antemano para calcular los valores de subpíxeles de resolución de $1/8$ de una manera sobredemanda, sólo cuando sean requeridos. Cuando se tome este enfoque, tanto el método de interpolación de acuerdo con TML5 y de acuerdo con la invención requieren 16 veces la memoria de imagen original para almacenar los valores de subpíxel de resolución de $1/2$ y de resolución de $1/4$. Sin embargo, si el método de interpolación directo de acuerdo con TML6 se usa de la misma manera, los valores intermedios para los subpíxeles de resolución de $1/2$ y de resolución de $1/4$ se deben almacenar. Estos valores intermedios están representados con una precisión de 32 bits y esto da por resultado un requerimiento de memoria 64 veces el de la imagen no interpolada original.

50 A continuación, la complejidad del método de interpolación de valor de subpíxel de acuerdo con la invención, cuando se aplica en un decodificador de video para calcular valores para subpíxeles a una resolución de píxel de hasta $1/8$, se compara con la de los esquemas de interpolación usados en TML5 y TML6. Como en el análisis equivalente para interpolación de valor de subpíxel de resolución de $1/4$ de píxel anteriormente descrita, se supone que en cada método la interpolación de cualquier valor de subpíxel se analiza mediante el uso de sólo un número mínimo de etapas requeridas para obtener un valor correctamente interpolado. También se supone que cada método es implementado de una manera basada en bloques, de tal manera que los valores intermedios comunes para todos los subpíxeles que han de ser interpolados en un bloque de $N \times M$ se calculan sólo una vez.

60 La tabla 2 resume las complejidades de los tres métodos de interpolación. La complejidad se mide en términos del número de filtro de 8 ramas y operaciones de interpolación lineal realizadas en cada método. La tabla presenta el número de operaciones requeridas para interpolar cada uno de los 63 subpíxeles de resolución de $1/8$ con un bloque de 4×4 de píxeles de imagen originales, cada posición de subpíxel está identificada con un número correspondiente, como se ilustra en la figura 22. En la figura 22, la posición 1 es la posición de un píxel de imagen original y las posiciones de 2 a 64 son posiciones de subpíxel. Cuando se calcula el número promedio de operaciones, se ha supuesto que la probabilidad de un vector de movimiento que indica a cada posición de subpíxel

es la misma. La complejidad promedio es por lo tanto en promedio de las 63 sumas calculadas para cada posición de subpíxel y la posición de píxel completo individual.

5 Tabla 2: Complejidad de interpolación de subpíxel de resolución de 1/8 en TML5, TML6 y el método de acuerdo con la invención. (Los resultados se muestran por separado para el método preferido 1 y el método preferido 2)

Posición	TML5		TML6		Método preferido 1		Método preferido 2	
	Lineal	8 ramas	Lineal	8 ramas	Lineal	8-tap	Lineal	8 ramas
1	0	0	0	0	0	0	0	0
3,5,7,17,33,49	0	16	0	16	0	16	0	16
19,21,23,35,37,39,51,53,55	0	60	0	60	0	60	0	60
2,8,9,57	16	16	0	16	16	16	16	16
4,6,25,41	16	32	0	16	16	32	16	32
10,16,58,64	32	76	0	60	16	32	16	32
11,13,15,59,61,63	16	60	0	60	16	60	16	60
18,24,34,40,50,56	16	76	0	60	16	60	16	60
12,14,60,62	32	120	0	60	16	32	16	32
26,32,42,48	32	108	0	60	16	32	16	32
20,22,36,38,52,54	16	120	0	60	16	76	16	76
27,29,31,43,45,47	16	76	0	60	16	76	16	76
28,30,44,46	32	152	0	60	16	60	16	60
Promedio	64	290,25	0	197,75	48	214,75	48	192,75

10 Como se puede ver a partir de la tabla 2, el número de operaciones de filtro de 8 ramas realizadas de acuerdo con los métodos preferidos 1 y 2 son, respectivamente, 26 % y 34 % menores que el número de operaciones de filtración de 8 ramas realizadas en el método de interpolación de valor de subpíxel de TML5. El número de operaciones lineales es 25 % menor, tanto en el método preferido 1 como en el método preferido 2, en comparación con TML5, pero esta mejora es de menor importancia en comparación con la reducción de las operaciones de filtración de 8 ramas. Se puede ver además que el método de interpolación directa usado en TML6 tiene una complejidad comparable con la de los métodos preferidos 1 y 2 cuando se usan para interpolar valores para subpíxeles de resolución de 1/8.

15

REIVINDICACIONES

1. Un método de interpolación de una imagen, comprendiendo la imagen píxeles (A) dispuestos en filas y columnas y representados por valores que tienen un intervalo dinámico especificado, residiendo los píxeles (A) de las filas en posiciones horizontales de unidad y residiendo los píxeles (A) de las columnas en posiciones verticales de unidad, para generar valores para subpíxeles (b, c, d, e, f, g, h, i), estando situado un subpíxel (b, c, d, e, f, g, h, i) en al menos uno de una posición horizontal fraccionada y vertical fraccionada, pudiéndose representar las posiciones verticales fraccionarias y horizontales fraccionarias de acuerdo con la notación matemática $1/2^x$, donde x es un número entero positivo en el intervalo de 1 a N, representando $1/2^x$ un nivel particular de interpolación de subpíxel y N representando un nivel máximo de interpolación de subpíxel, comprendiendo el método:
- a) interpolar valores para subpíxeles (b) situados en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad, y para subpíxeles (b) situados en las posiciones horizontales de unidad y verticales $1/2^{N-1}$ de unidad utilizando sumas ponderadas de píxeles (A) que residen en respectivas posiciones horizontales de unidad y verticales de unidad;
 - b) interpolar un valor para un subpíxel (c) situado en una posición horizontal $1/2^{N-1}$ de unidad y vertical $1/2^{N-1}$ de unidad utilizando una primera suma ponderada de los valores interpolados para subpíxeles (b) que residen en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad o una segunda suma ponderada de los valores interpolados para subpíxeles (b) que residen en las posiciones horizontales de unidad y verticales $1/2^{N-1}$ de unidad obtenidas en la etapa a); y
 - c) interpolar un valor para un subpíxel (h, i) situado en una posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad utilizando:
 - una media ponderada del valor de un primer subpíxel (b) situado en una posición horizontal $1/2^{N-m}$ de unidad y vertical $1/2^{N-n}$ de unidad y el valor de un segundo subpíxel (b) situado en una posición horizontal $1/2^{N-p}$ de unidad y vertical $1/2^{N-q}$ de unidad, o
 - una media ponderada del valor de un píxel (A) situado en una de posición horizontal de unidad y vertical de unidad y el valor de un subpíxel (c) situado en una posición horizontal $1/2^{N-m}$ de unidad y vertical $1/2^{N-n}$ de unidad,
 tomando las variables m, n, p y q valores enteros en el intervalo de 1 a N, tal que los respectivos primer y segundo subpíxeles (b) o el respectivo píxel (A) y subpíxel (c) se encuentran en diagonal con respecto al subpíxel (h, i) en la posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad que se interpola.
2. Un método de acuerdo con la reivindicación 1, en el que una primera y una segunda ponderaciones se utilizan en la media ponderada que se hace referencia en la etapa c), siendo las magnitudes relativas de las ponderaciones proporcionales a la proximidad en la línea recta diagonal del subpíxel (h, i) en posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad en los respectivos primer y segundo subpíxeles (b) o el respectivo píxel (A) y subpíxel (c) utilizado en la etapa c).
3. Un método de acuerdo con la reivindicación 2, en el que en una situación donde los respectivos primer y segundo subpíxeles (b) o el respectivo píxel (A) y subpíxel (c) utilizados en la etapa c) están situados simétricamente con respecto al subpíxel (h, i) en la posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad que se interpola, teniendo la primera y la segunda ponderaciones valores iguales.
4. Un método de acuerdo con la reivindicación 1, en el que la primera suma ponderada de los valores de subpíxeles (b) que residen en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad en la etapa b) se utiliza cuando se interpola un valor para un subpíxel (f) en la posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad.
5. Un método de acuerdo con la reivindicación 1, en el que la segunda suma ponderada de los valores de subpíxeles (b) que residen en las posiciones horizontales de unidad y verticales $1/2^N$ de unidad en la etapa b) se utiliza cuando se interpola un valor para un subpíxel (g) en la posición horizontal $1/2^N$ de unidad y vertical $1/2^{N-1}$ de unidad.
6. Un método de acuerdo con la reivindicación 1, que comprende interpolar valores para subpíxeles (h, i) en las posiciones horizontales $1/2^N$ de unidad y verticales $1/2^N$ de unidad tomando un promedio del valor de un píxel (A) que está situado en una posición horizontal de unidad y vertical de unidad y el valor de un subpíxel (c) situado en una posición horizontal $1/2^{N-1}$ de unidad y vertical $1/2^{N-1}$ de unidad.
7. Un método de acuerdo con la reivindicación 1, que comprende interpolar valores para subpíxeles (h, i) en posiciones horizontales $1/2^N$ de unidad y verticales $1/2^N$ de unidad tomando un promedio del valor de un subpíxel (b) situado en una posición horizontal $1/2^{N-1}$ de unidad y vertical de unidad y el valor de un subpíxel (b) situado en una posición horizontal de unidad y vertical $1/2^{N-1}$ de unidad.
8. Un método de acuerdo con la reivindicación 1, en el que N es uno de los valores enteros 2, 3, y 4.
9. Un método de acuerdo con la reivindicación 1, en el que en al menos una de la etapa a) y de la etapa b),

interpolando un valor de subpíxel usando una suma ponderada implica el cálculo de un valor intermedio para el valor de subpíxel, teniendo dicho valor intermedio un intervalo dinámico mayor que el intervalo dinámico especificado.

5 **10.** Un método de acuerdo con la reivindicación 9, en el que el valor intermedio para un subpíxel que tiene una resolución de subpíxel $1/2^{N-1}$ se utiliza en la interpolación de un valor de subpíxel para un subpíxel que tiene una resolución de subpíxel $1/2^N$.

10 **11.** Un método de acuerdo con la reivindicación 1, en el que el intervalo dinámico especificado corresponde al intervalo de valores que dichos píxeles (A) puede tomar.

15 **12.** Un método de acuerdo con la reivindicación 11, que comprende representar un valor de subpíxel interpolado en la etapa a) utilizando un primer valor intermedio, teniendo dicho primer valor intermedio un intervalo dinámico igual al intervalo dinámico especificado de los valores de los píxeles respectivos utilizados en la suma ponderada de la etapa a) multiplicado por un valor igual a la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa a).

20 **13.** Un método de acuerdo con la reivindicación 12, en el que el intervalo dinámico de dicho primer valor intermedio se define por el número de bits utilizados para representar los valores de los píxeles respectivos utilizados en la suma ponderada de la etapa a) más el número de bits necesarios para representar la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa a).

25 **14.** Un método de acuerdo con la reivindicación 12, que comprende truncar el primer valor intermedio dividiendo matemáticamente el primer valor intermedio por un primer factor de escala, teniendo dicho primer factor de escala un valor igual a la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa a), formando de este modo un valor de subpíxel con un intervalo dinámico igual al intervalo dinámico especificado de dichos valores de píxel.

30 **15.** Un método según la reivindicación 12, que comprende el uso de primeros valores intermedios de subpíxel calculados en la etapa a) como los valores interpolados en las respectivas sumas ponderadas de la etapa b), y que representan un valor de subpíxel interpolado en la etapa b) utilizando un segundo valor intermedio, teniendo dicho segundo valor intermedio un intervalo dinámico igual al intervalo dinámico de los valores de subpíxeles respectivos utilizados en la suma ponderada de la etapa b) multiplicado por un valor igual a la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa b).

35 **16.** Un método según la reivindicación 15, que comprende truncar el segundo valor intermedio dividiendo matemáticamente el segundo valor intermedio por un segundo factor de escala, teniendo dicho segundo factor de escala un valor igual a la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa a) multiplicado por la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa b), formando de este modo un valor de subpíxel con un intervalo dinámico igual al intervalo dinámico especificado de dichos valores de píxel.

40 **17.** Un método de acuerdo con la reivindicación 14, que comprende el uso de unos primeros valores de subpíxel truncados intermedios cuando se interpola un valor para un subpíxel (h, i) que tiene una resolución de subpíxel $1/2^N$ en la etapa c).

45 **18.** Un método de acuerdo con la reivindicación 16, que comprende el uso de unos segundos valores de subpíxel truncados intermedios cuando se interpola un valor para un subpíxel (h, i) que tiene una resolución de subpíxel $1/2^N$ en la etapa c).

50 **19.** Un interpolador (730, 750, 845, 890) para interpolar una imagen, comprendiendo la imagen píxeles (A) dispuestos en filas y columnas y representados por valores que tienen un intervalo dinámico especificado, los píxeles (A) en las filas residiendo en posiciones horizontales de unidad y los píxeles (A) en las columnas residiendo en posiciones verticales de unidad, estando el interpolador (730, 750, 845, 890) adaptados para generar valores para subpíxeles (b, c, d, e, f, g, h, i), estando un subpíxel (b, c, d, e, f, g, h, i) situado en al menos uno de una posición horizontal fraccionada y una vertical fraccionada, pudiéndose representar las posiciones verticales fraccionarias y horizontales fraccionarias de acuerdo con la notación matemática $1/2^X$, donde x es un número entero positivo en el intervalo de 1 a N, representando $1/2^X$ un nivel particular de interpolación de subpíxel y representando N un nivel máximo de interpolación de subpíxel, estando el interpolador (730, 750, 845, 890) adaptado para:

60 a) interpolar valores para subpíxeles (b) situados en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad, y por subpíxeles (b) situados en posiciones horizontales $1/2^{N-1}$ de unidad y verticales utilizando sumas ponderadas de píxeles (A) que residen en respectivas posiciones horizontales de unidad y verticales de unidad;

65 b) interpolar un valor para un subpíxel (c) situado en una posición horizontal $1/2^{N-1}$ de unidad y vertical $1/2^{N-1}$ de unidad utilizando una primera suma ponderada de los valores interpolados para subpíxeles (b) que residen en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad o una segunda suma ponderada

de los valores interpolados para subpíxeles (b) que residen en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad obtenidas en la etapa a); y
 c) interpolar un valor para un subpíxel (h, i) situado en una posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad utilizando:

- 5
- una media ponderada del valor de un primer subpíxel (b) situado en una posición horizontal $1/2^{N-m}$ de unidad y vertical $1/2^{N-n}$ de unidad y el valor de un segundo subpíxel (b) situado en una posición horizontal $1/2^{N-p}$ de unidad y vertical $1/2^{N-q}$ de unidad, o
 - una media ponderada del valor de un píxel (A) situado en una de posición horizontal de unidad y vertical de unidad y el valor de un subpíxel (c) situado en una posición horizontal $1/2^{N-m}$ de unidad y vertical $1/2^{N-n}$ de unidad,
- 10
- tomando las variables m, n, p y q valores enteros en el intervalo de 1 a N, tal que los respectivos primer y segundo subpíxeles (b) o el respectivo píxel (A) y subpíxel (c) se encuentran en diagonal con respecto al subpíxel (h, i) en la posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad que se interpola.
- 15

20. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 19, en donde el interpolador (730, 750, 845, 890) está configurado para utilizar una primera y una segunda ponderaciones en la media ponderada que se hace referencia en la etapa c), siendo las magnitudes relativas de las ponderaciones proporcionales a la proximidad en la línea recta diagonal del subpíxel (h, i) en posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad en los respectivos primer y segundo subpíxeles (b) o el respectivo píxel (A) y subpíxel (c) utilizado en la etapa c).

20

21. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 20, en donde el interpolador (730, 750, 845, 890) está configurado para utilizar una primera y una segunda interpolaciones que tienen valores iguales en una situación donde los respectivos primer y segundo sub -píxeles (b) o el correspondiente píxel (A) y subpíxel (c) que se utiliza en la etapa c) están colocados simétricamente con respecto al subpíxel (h, i) en posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad que se interpola.

25

22. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 19, en donde el interpolador (730, 750, 845, 890) está configurado para utilizar la primera suma ponderada de los valores de subpíxeles (b) que residen en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad en la etapa b) cuando se interpola un valor para un subpíxel (f) en posición horizontal $1/2^{N-1}$ de unidad y vertical $1/2^N$ de unidad.

30

23. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 19, en donde el interpolador (730, 750, 845, 890) está configurado para utilizar la segunda suma ponderada de los valores de subpíxeles (b) que residen en las posiciones horizontales de unidad y verticales $1/2^{N-1}$ de unidad en la etapa b) cuando se interpola un valor para un subpíxel (g) en posición horizontal $1/2^N$ de unidad y vertical $1/2^{N-1}$ de unidad.

35

24. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 19, en donde el interpolador (730, 750, 845, 890) está configurado para interpolar valores de subpíxeles (h, i) en posiciones horizontales $1/2^N$ de unidad y verticales $1/2^N$ de unidad tomando un promedio del valor de un píxel (A) situado en una posición horizontal de unidad y vertical de unidad y el valor de un subpíxel (c) situado en una posición horizontal $1/2^{N-1}$ de unidad y vertical $1/2^{N-1}$ de unidad.

40

25. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 19, en donde el interpolador (730, 750, 845, 890) está configurado para interpolar valores de subpíxeles (h, i) en posiciones horizontales $1/2^N$ de unidad y verticales $1/2^N$ de unidad tomando un promedio del valor de un subpíxel (b) situado en una posición horizontal $1/2^{N-1}$ de unidad y vertical de unidad, y el valor de un subpíxel (b) situado en una posición horizontal de unidad y vertical $1/2^{N-1}$ de unidad.

45

26. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 19, en el que N es uno de los valores enteros 2, 3, y 4.

50

27. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 19, en donde en al menos una de la etapa a) y de la etapa b), el interpolador (730, 750, 845, 890) está configurado para interpolar un valor de subpíxel usando una suma ponderada que implica el cálculo de un valor intermedio para el valor de subpíxel, teniendo dicho valor intermedio un intervalo dinámico mayor que el intervalo dinámico especificado.

55

28. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 27, en donde el interpolador (730, 750, 845, 890) está configurado para utilizar el valor intermedio para un subpíxel que tiene una resolución de subpíxel $1/2^{N-1}$ en la interpolación de un valor de subpíxel para un subpíxel que tiene una resolución de subpíxel $1/2^N$.

60

29. Un interpolador (730, 750, 845, 890) de acuerdo la reivindicación 19, en el que el intervalo dinámico especificado corresponde al intervalo de valores que dichos píxeles (A) pueden tomar.

65

30. Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 29, en donde el interpolador (730, 750,

845, 890) está configurado para representar un valor de subpíxel interpolado en la etapa a) utilizando un primer valor intermedio, teniendo dicho primer valor intermedio un intervalo dinámico igual al intervalo dinámico especificado de los valores de los píxeles respectivos utilizados en la suma ponderada de la etapa a) multiplicado por un valor igual a la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa a).

5 **31.** Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 30, en el que el intervalo dinámico de dicho primer valor intermedio se define por el número de bits utilizados para representar los respectivos valores de los píxeles utilizados en la suma ponderada de la etapa a) más el número de bits requeridos para representar la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa a).

10 **32.** Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 30, en donde el interpolador (730, 750, 845, 890) está configurado para truncar el primer valor intermedio dividiendo matemáticamente el primer valor intermedio por un primer factor de escala, teniendo dicho primer factor de escala un valor igual a la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa a), formando de este modo un valor de subpíxel con un intervalo dinámico igual al intervalo dinámico especificado de dichos valores de píxel.

15 **33.** Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 30, en donde el interpolador (730, 750, 845, 890) está configurado para utilizar los primeros valores de subpíxel intermedios calculados en la etapa a) como los valores interpolados en las respectivas sumas ponderadas de la etapa b), y para representar un valor de subpíxel interpolado en la etapa b) utilizando un segundo valor intermedio, teniendo dicho segundo valor intermedio un intervalo dinámico igual al intervalo dinámico de los valores de subpíxeles respectivos utilizados en la suma ponderada de la etapa b) multiplicado por un valor igual a la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa b).

20 **34.** Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 33, en donde el interpolador (730, 750, 845, 890) está configurado para truncar el segundo valor intermedio dividiendo matemáticamente el segundo valor intermedio por un segundo factor de escala, teniendo dicho segundo factor de escala un valor igual a la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa a) multiplicado por la suma de las respectivas ponderaciones utilizadas en la suma ponderada de la etapa b), formando de este modo un valor de subpíxel con un intervalo dinámico igual al intervalo dinámico especificado de dichos valores de píxel.

25 **35.** Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 32, en donde el interpolador (730, 750, 845, 890) está configurado para utilizar los primeros valores truncados intermedios de subpíxel cuando se interpola un valor para un subpíxel (h, i) que tiene una resolución de subpíxel $1/2^N$ en la etapa c).

30 **36.** Un interpolador (730, 750, 845, 890) de acuerdo con la reivindicación 34, en donde el interpolador (730, 750, 845, 890) está configurado para utilizar los segundos valores truncados intermedios de subpíxel cuando se interpola un valor para un subpíxel (h, i) que tiene resolución de subpíxel $1/2^N$ en la etapa c).

35 **37.** Un codificador de vídeo (700), un decodificador de vídeo (800) o un códec de vídeo (700, 800) que comprende un interpolador (730, 750, 845, 890) de acuerdo con cualquiera de las reivindicaciones 19 a 36.

40 **38.** Un terminal de comunicaciones (60, MS) que comprende un codificador de vídeo, un decodificador de vídeo o un códec de vídeo de acuerdo con la reivindicación 37.

45 **39.** Un programa de ordenador para interpolar una imagen, comprendiendo la imagen píxeles (A) dispuestos en filas y columnas y representados por valores que tienen un intervalo dinámico especificado, residiendo los píxeles (A) de las filas en posiciones horizontales de unidad y los píxeles (A) de las columnas residiendo en posiciones verticales de unidad, para generar valores para subpíxeles (b, c, d, e, f, g, h, i), estando situado un subpíxel (b, c, d, e, f, g, h, i) en al menos uno de una posición horizontal fraccionada y una vertical fraccionada, pudiéndose representar las posiciones verticales fraccionarias y horizontales fraccionarias de acuerdo con la notación matemática $1/2^x$, donde x es un número entero positivo en el intervalo de 1 a N, representando $1/2^x$ un nivel particular de interpolación de subpíxel y N representando un nivel máximo de interpolación de subpíxel, comprendiendo el programa de ordenador:

50 a) un código de programa para interpolar valores para subpíxeles (b) situados en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad, y para subpíxeles (b) situados en posiciones horizontales de unidad y verticales $1/2^{N-1}$ de unidad utilizando sumas ponderadas de píxeles (A) que residen en respectivas posiciones horizontales de unidad y verticales de unidad;

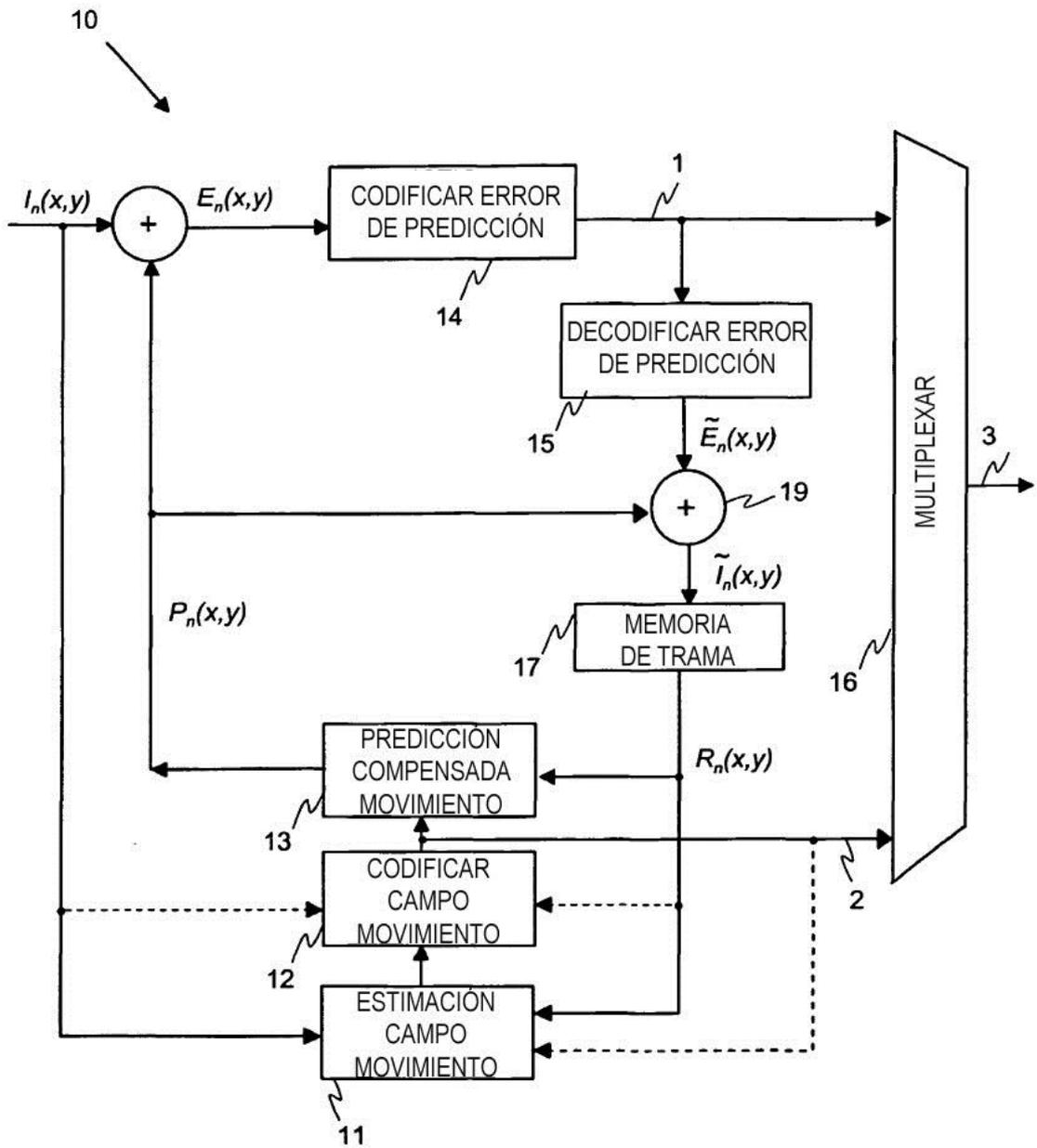
55 b) un código de programa para interpolar un valor para un subpíxel (c) situado en una posición horizontal $1/2^{N-1}$ de unidad y vertical $1/2^{N-1}$ de unidad utilizando una primera suma ponderada de los valores interpolados para subpíxeles (b) que residen en posiciones horizontales $1/2^{N-1}$ de unidad y verticales de unidad o una segunda suma ponderada de los valores interpolados para subpíxeles (b) que residen en las posiciones horizontales de unidad y verticales $1/2^{N-1}$ de unidad obtenidas en la etapa a); y

60 c) un código de programa para interpolar un valor para un subpíxel (h, i) situado en una posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad utilizando:

- una media ponderada del valor de un primer subpíxel (b) situado en una posición horizontal $1/2^{N-m}$ de unidad y vertical $1/2^{N-n}$ de unidad y el valor de un segundo subpíxel (b) situado en una posición horizontal $1/2^{N-p}$ de unidad y vertical $1/2^{N-q}$ de unidad, o

5 - una media ponderada del valor de un píxel (A) situado en una de posición horizontal de unidad y vertical de unidad y el valor de un subpíxel (c) situado en una posición horizontal $1/2^{N-m}$ de unidad y vertical $1/2^{N-n}$ de unidad,

10 tomando las variables m, n, p y q valores enteros en el intervalo de 1 a N, tal que los respectivos primer y segundo subpíxeles (b) o el respectivo píxel (A) y subpíxel (c) se encuentran en diagonal con respecto al subpíxel (h, i) en la posición horizontal $1/2^N$ de unidad y vertical $1/2^N$ de unidad que se interpola.



TÉCNICA ANTERIOR

Fig. 1

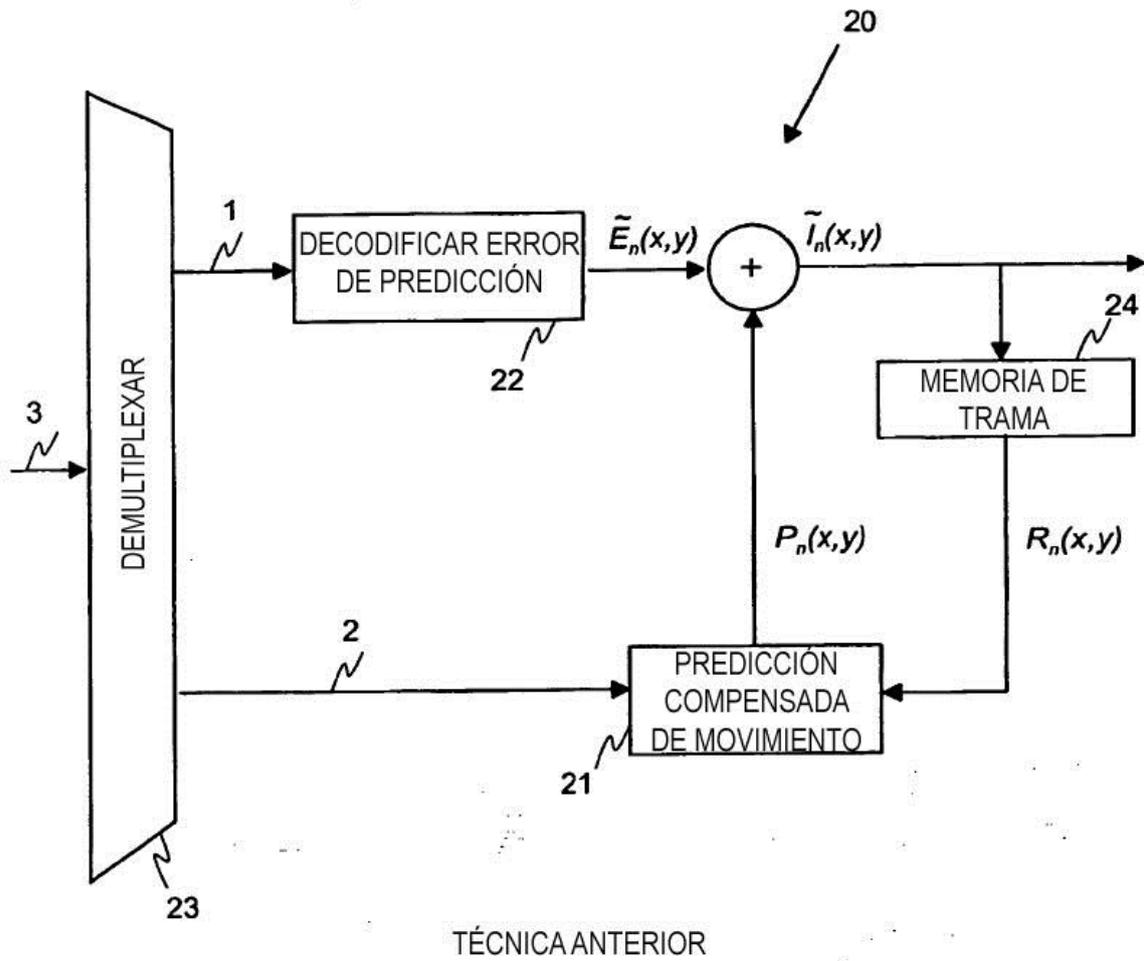


Fig. 2

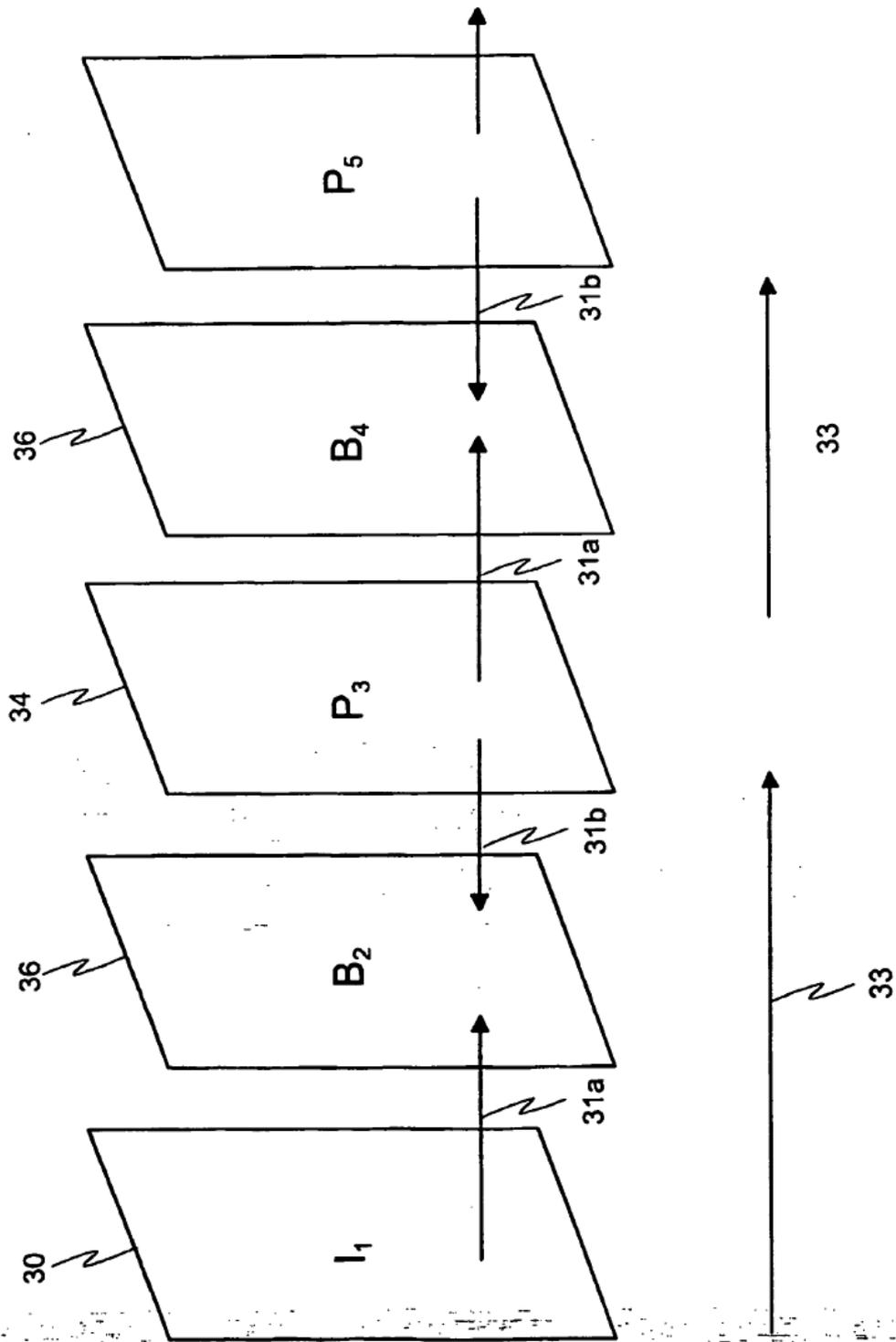


Fig. 3

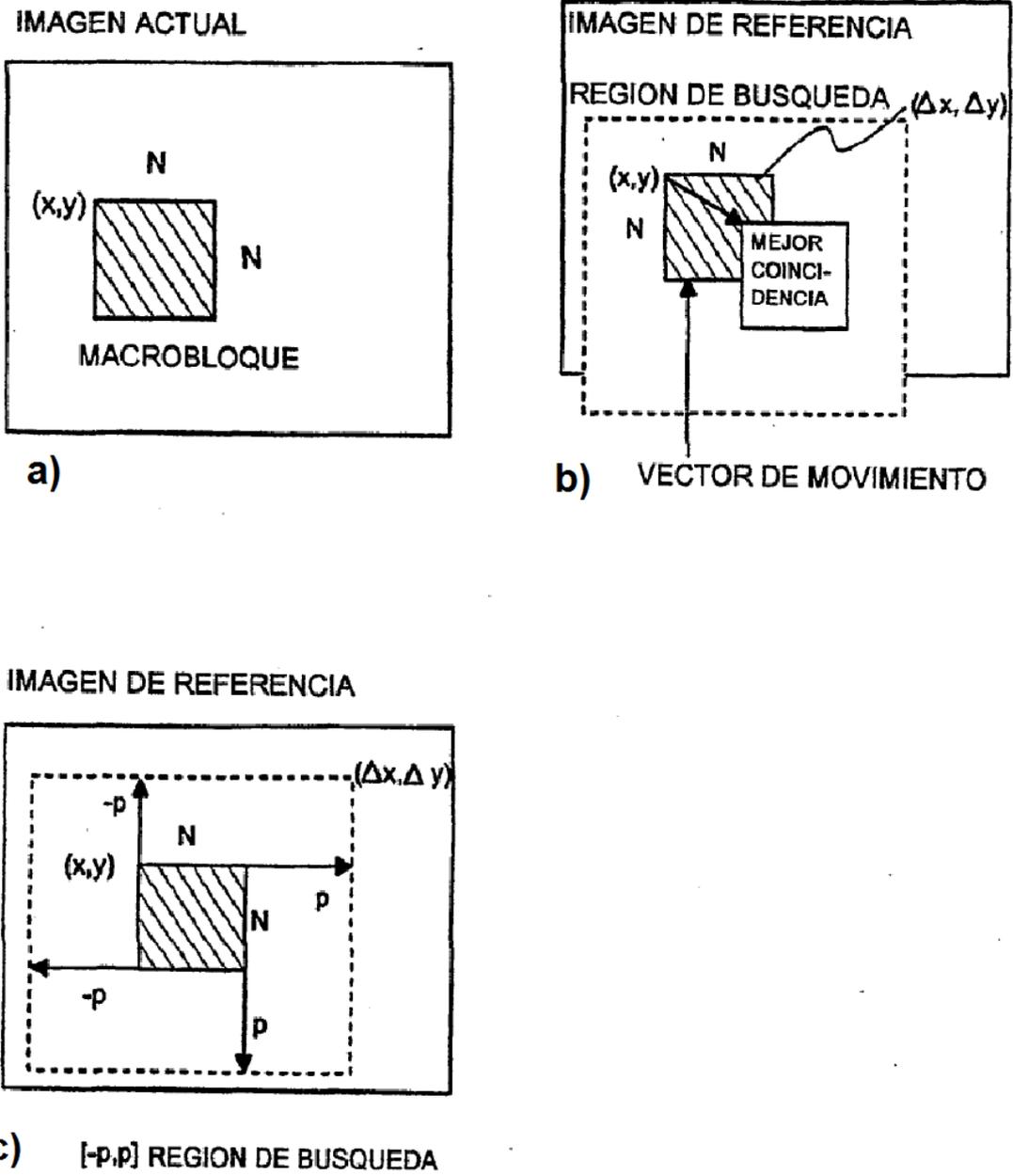


Fig. 4

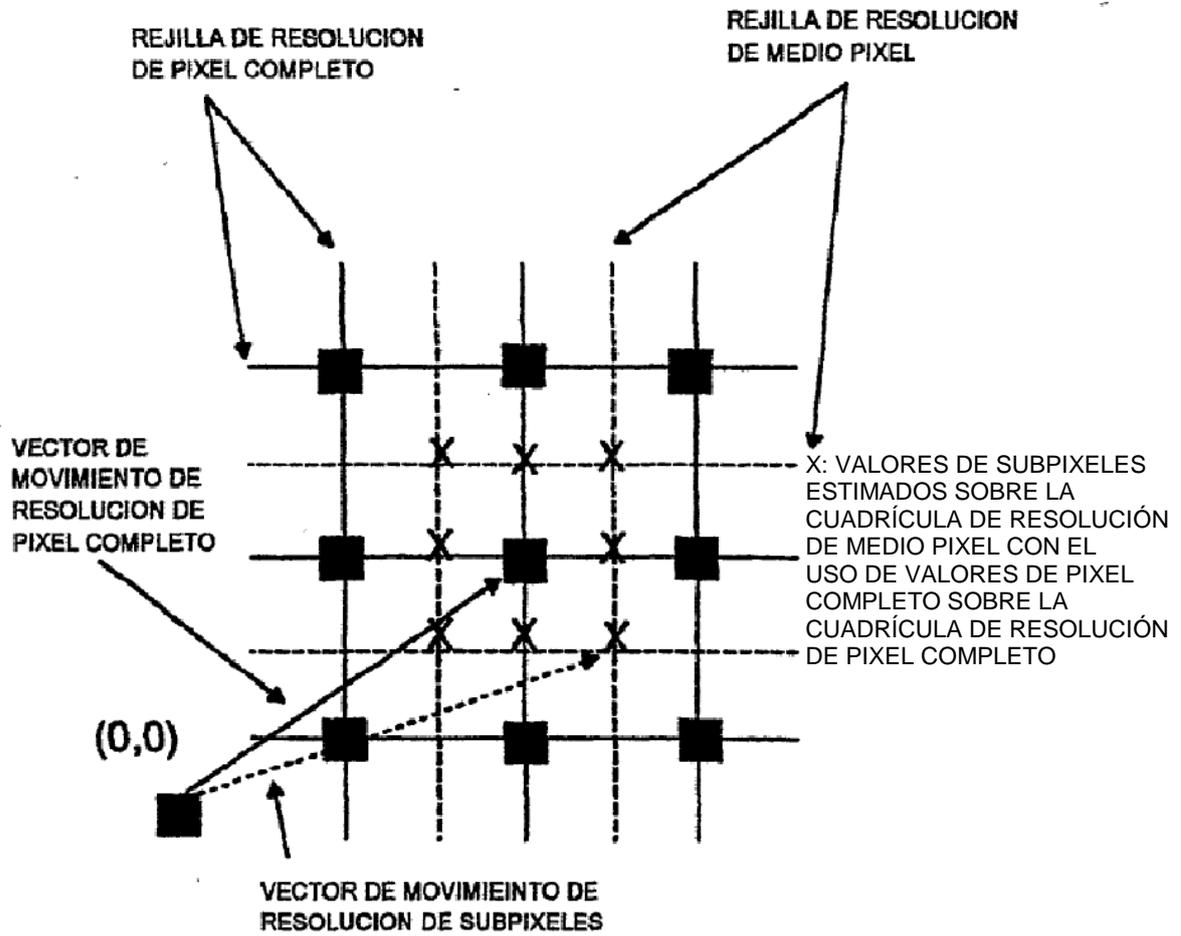


Fig. 5

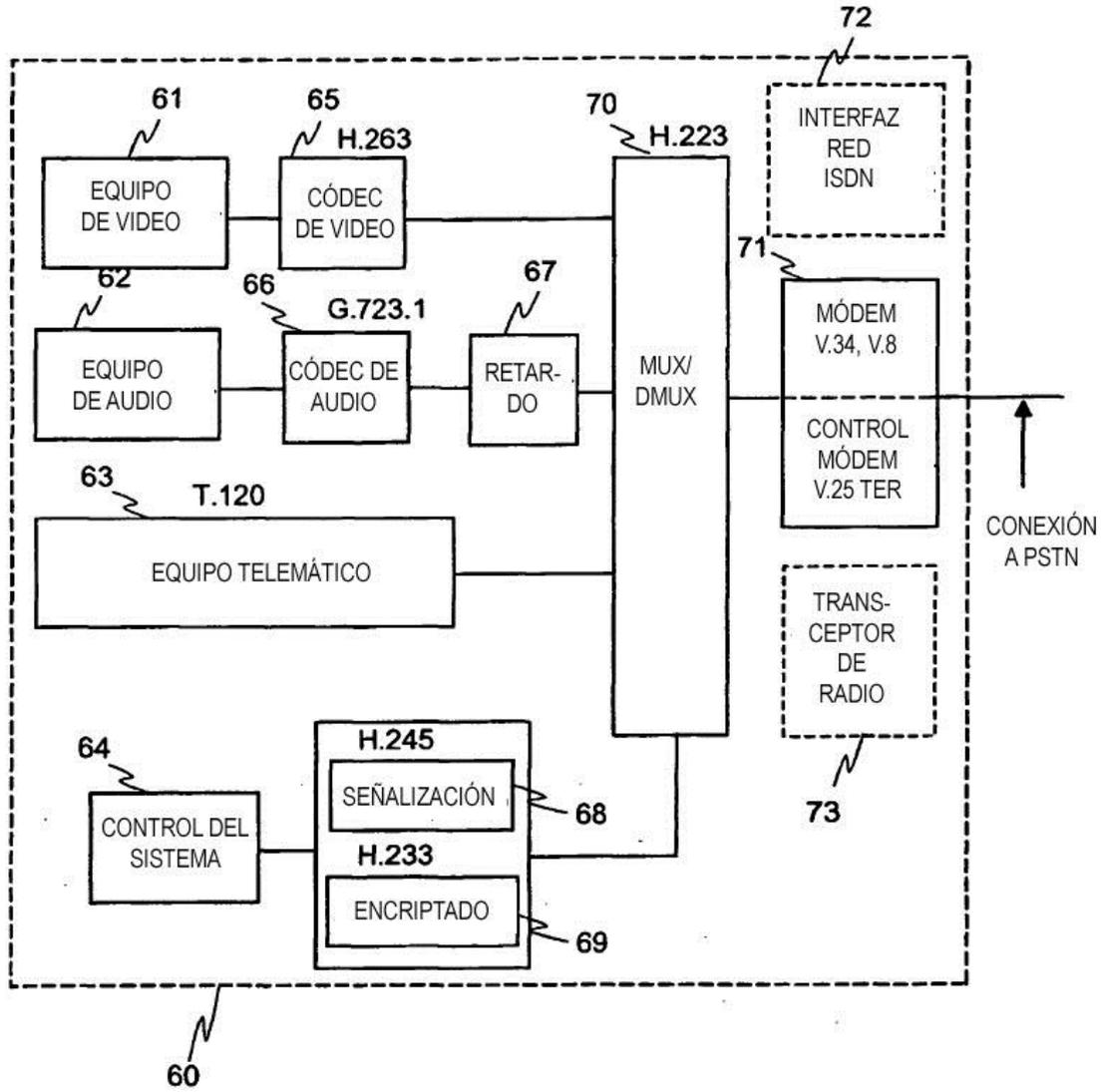


Fig. 6

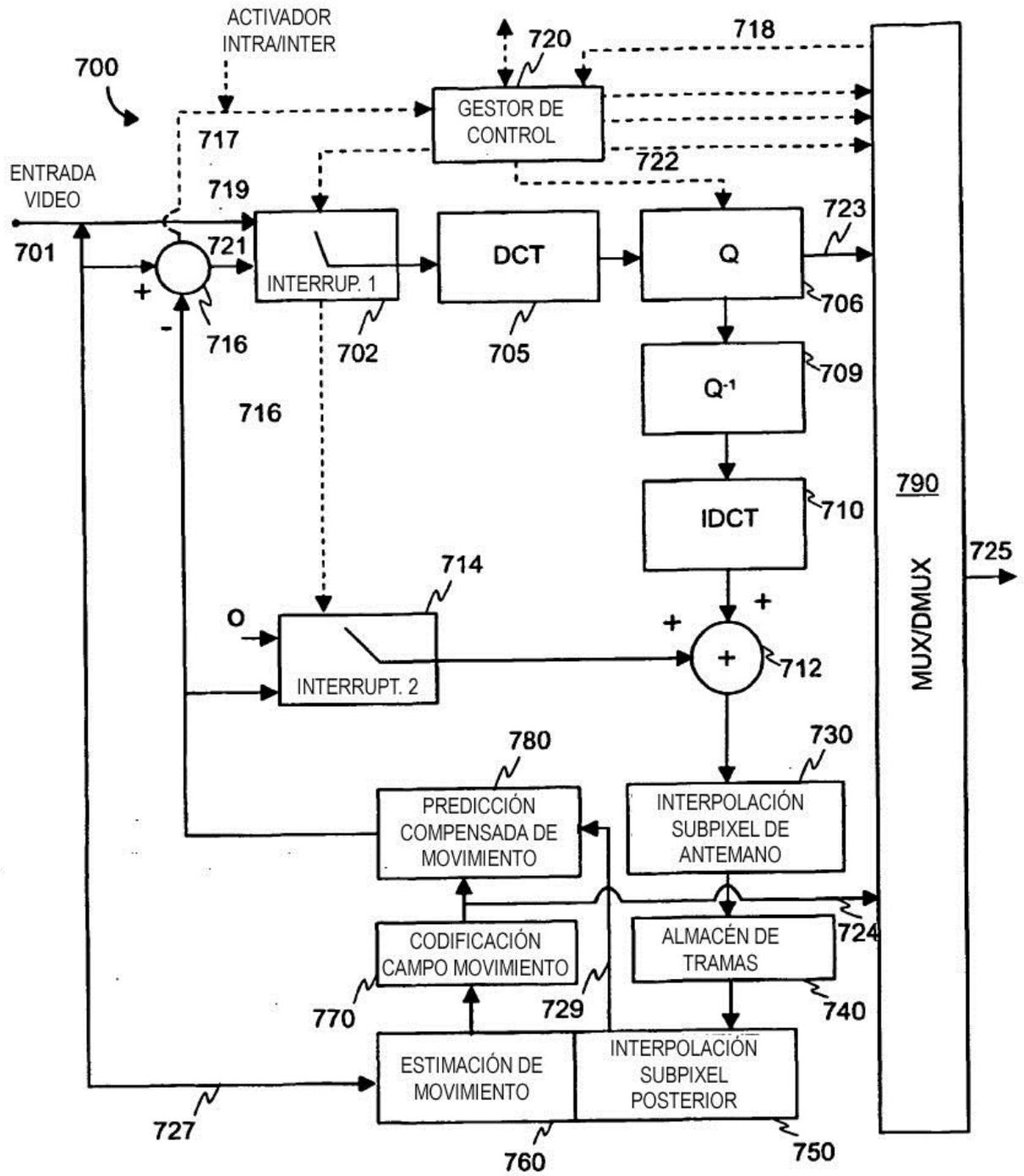


Fig. 7

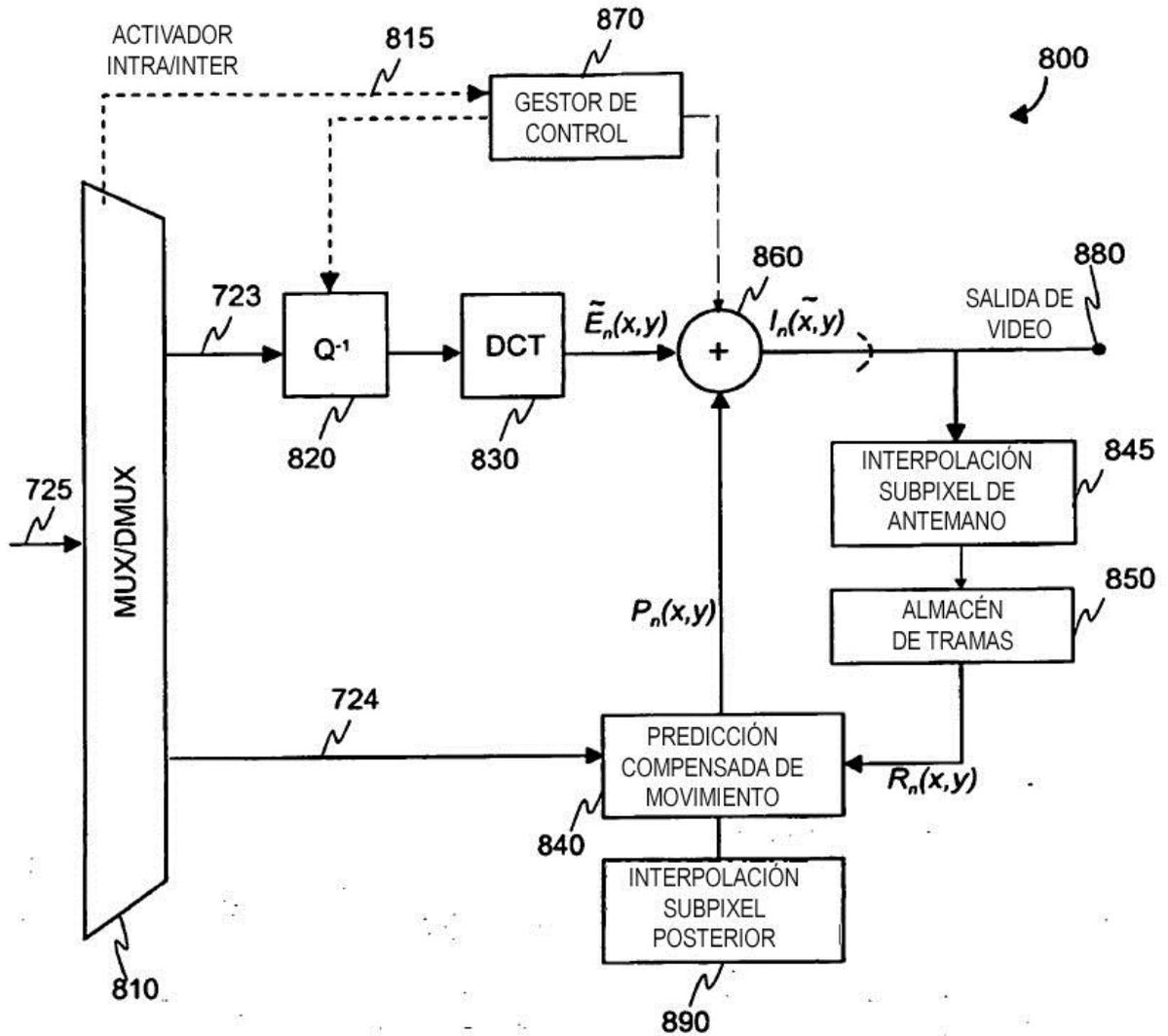


Fig. 8

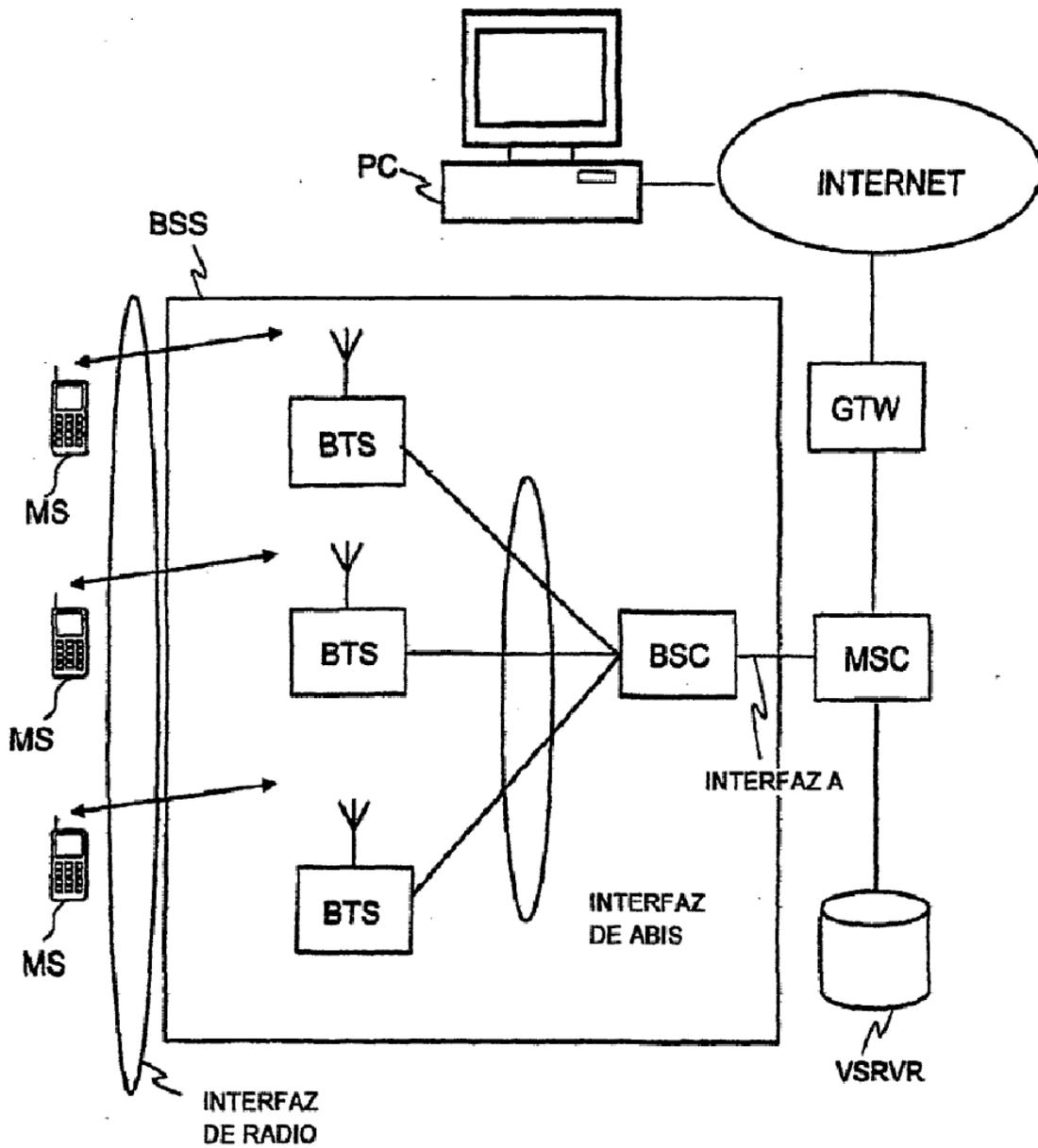


Fig. 11

A	d	b	d	A
e	e	e	e	
c	d	c	d	
e	e	e	f	
A				A

Fig. 12(a)

A₁	A₂	A₃	d	b	d	A₄	A₅	A₆
		e	e	e	e			
		c	d	c	d			
		e	e	e	f			
		A				A		

Figura 12(b)

A₁ **b₁**

A₂ **b₂**

A₃	d	b₃	d	A
e	e	e	e	
c	d	c	d	
e	e	e	f	
A₄		b₄		A

A₅ **b₅**

A₆ **b₆**

Figura 12(c)

A	d	b	d	A
e	g	e	g	
c	d	c	d	
e	g	e	f	
A				A

Fig. 13(a)

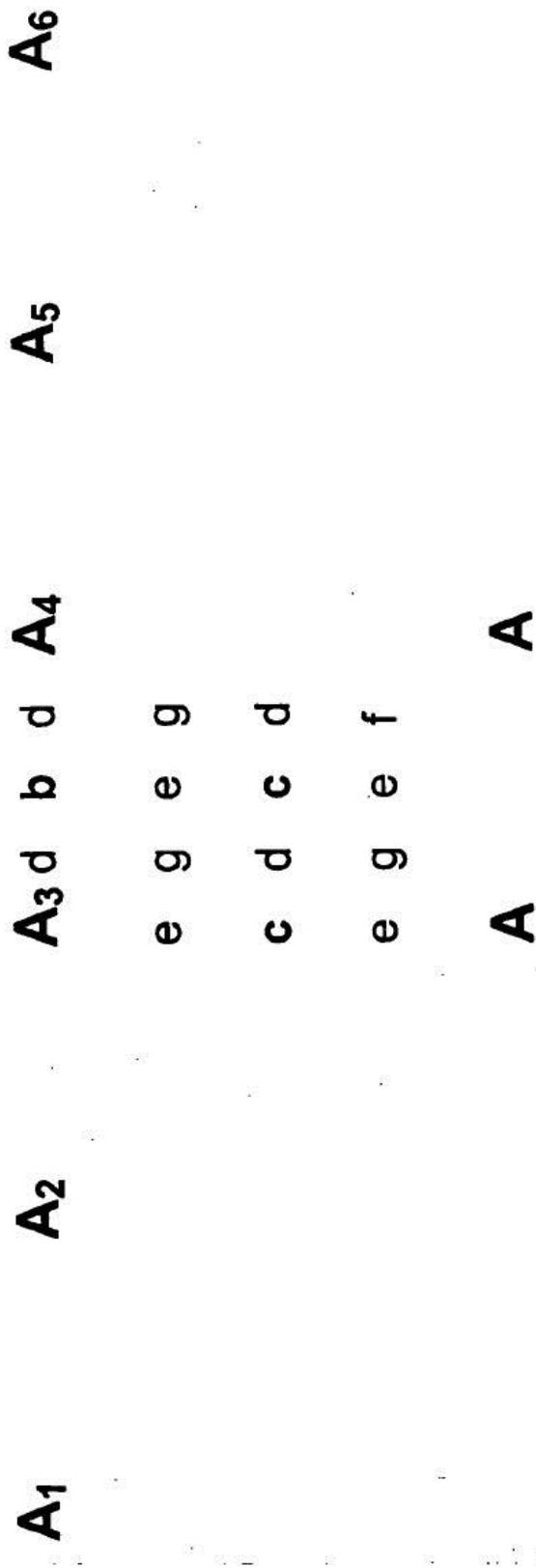


Figura 13(b)

A₁ **b₁**

A₂ **b₂**

A₃	d	b₃	d	A
e	g	e	g	
c	d	c	d	
e	g	e	f	
A₄		b₄		A

A₅ **b₅**

A₆ **b₆**

Figura 13(c)

A	d	b	d	A
e	h	f	h	
b	g	c	g	
e	h	f	i	
A				A

Fig. 14(a)

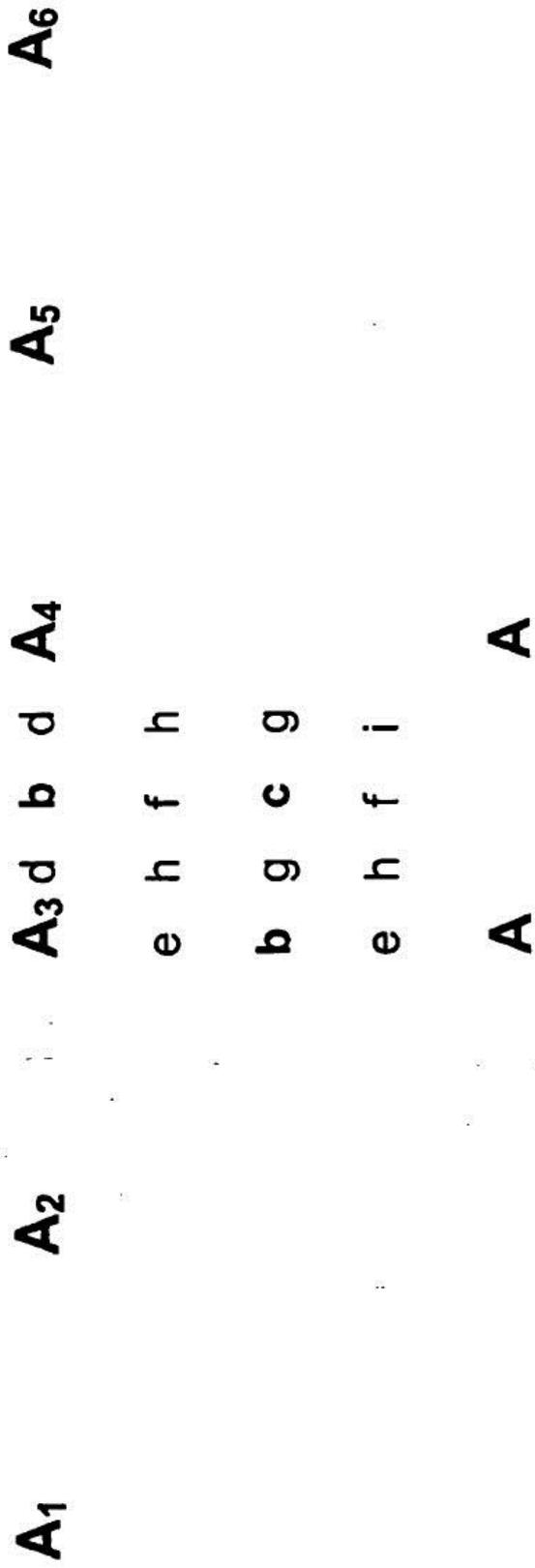


Figura 14(b)

A₁

A₂

A₃	d	b	d	A
	e	h	f	
	b	g	c	
	e	h	f	
A₄				A

A₅

A₆

Figura 14(c)

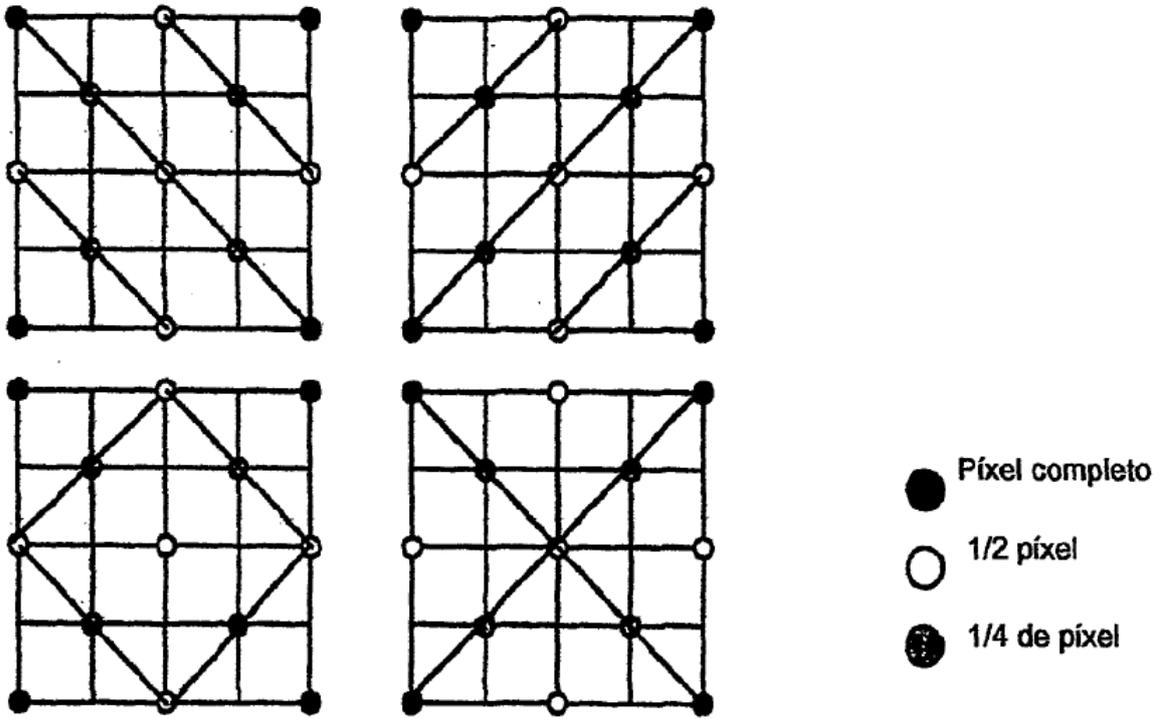


Figura 15

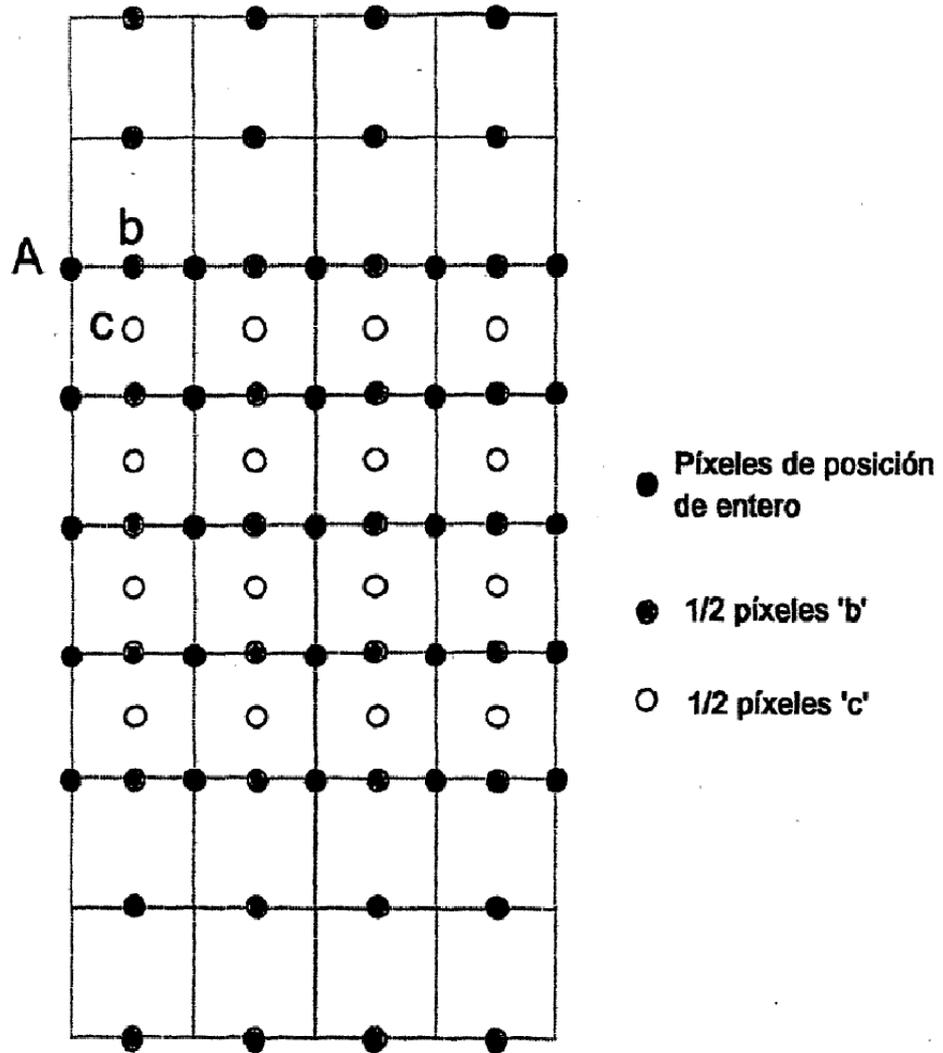


Fig. 16

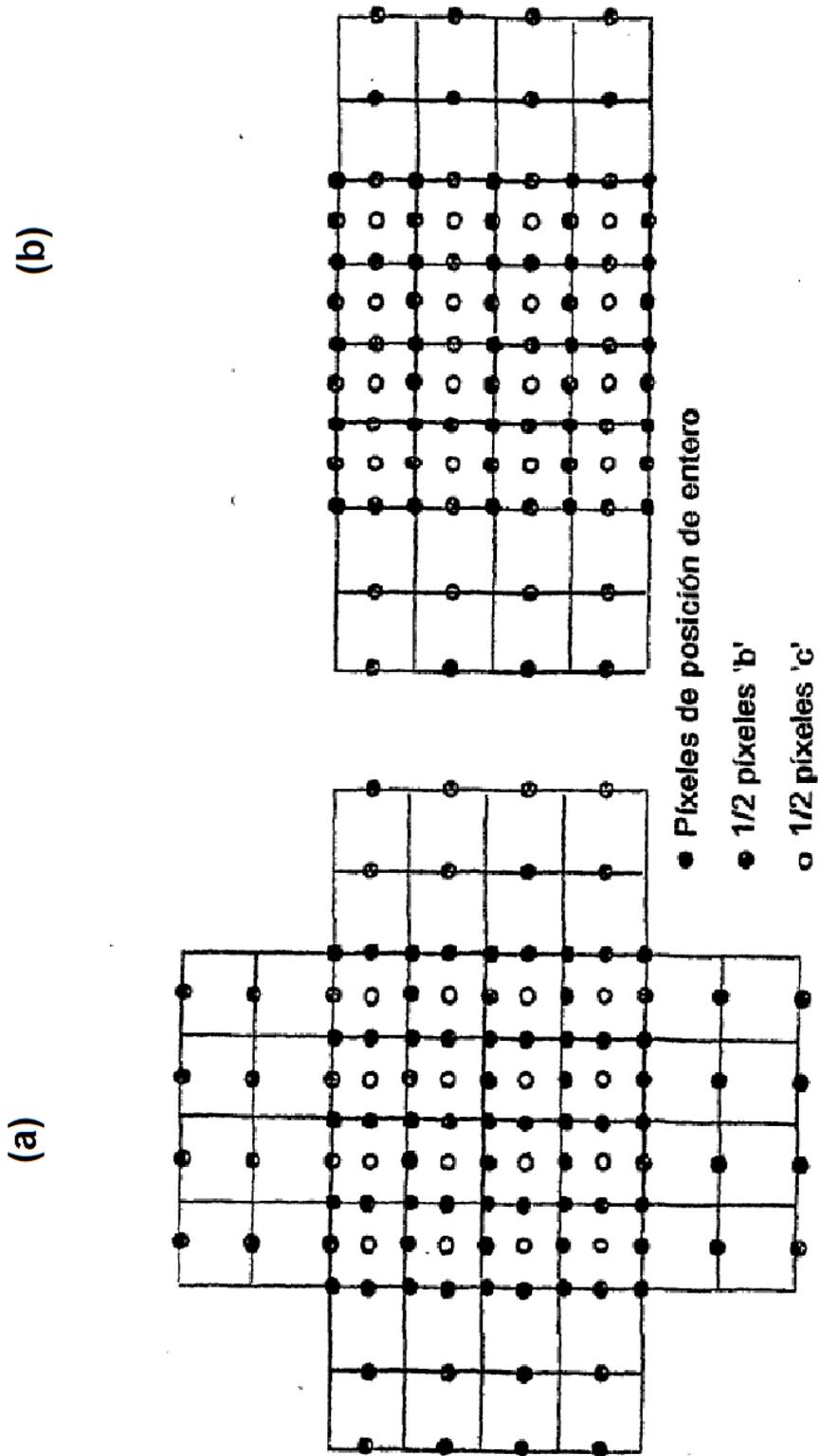


Fig. 17

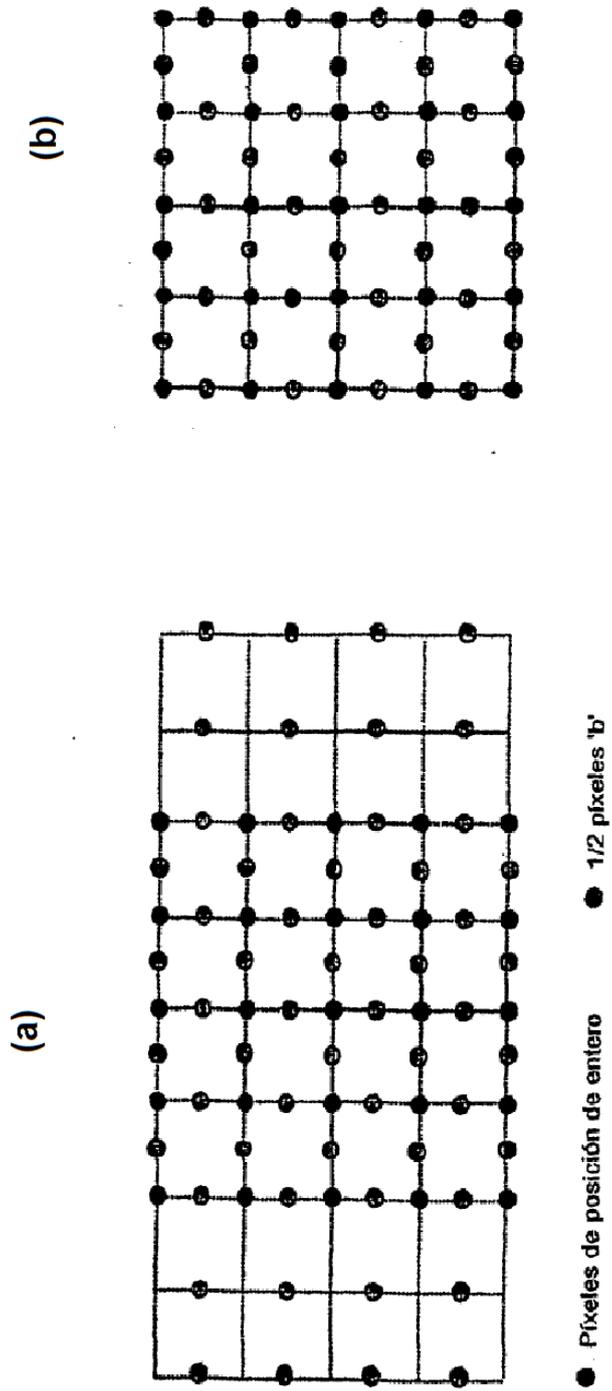


Fig. 18

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig. 19

A	d	b ¹	d	b ²	d	b ³	d	A
d	e	d	f	d	f	d	e	
b ¹	d	c ¹¹	d	c ¹²	d	c ¹³	d	
d	f	d	g	d	g	d	f	
b ²	d	c ²¹	d	c ²²	d	c ²³	d	
d	f	d	g	d	g	d	e	
b ³	d	c ³¹	d	c ³²	d	c ³³	d	
d	e	d	f	d	f	d	e	
A								

Figura 20

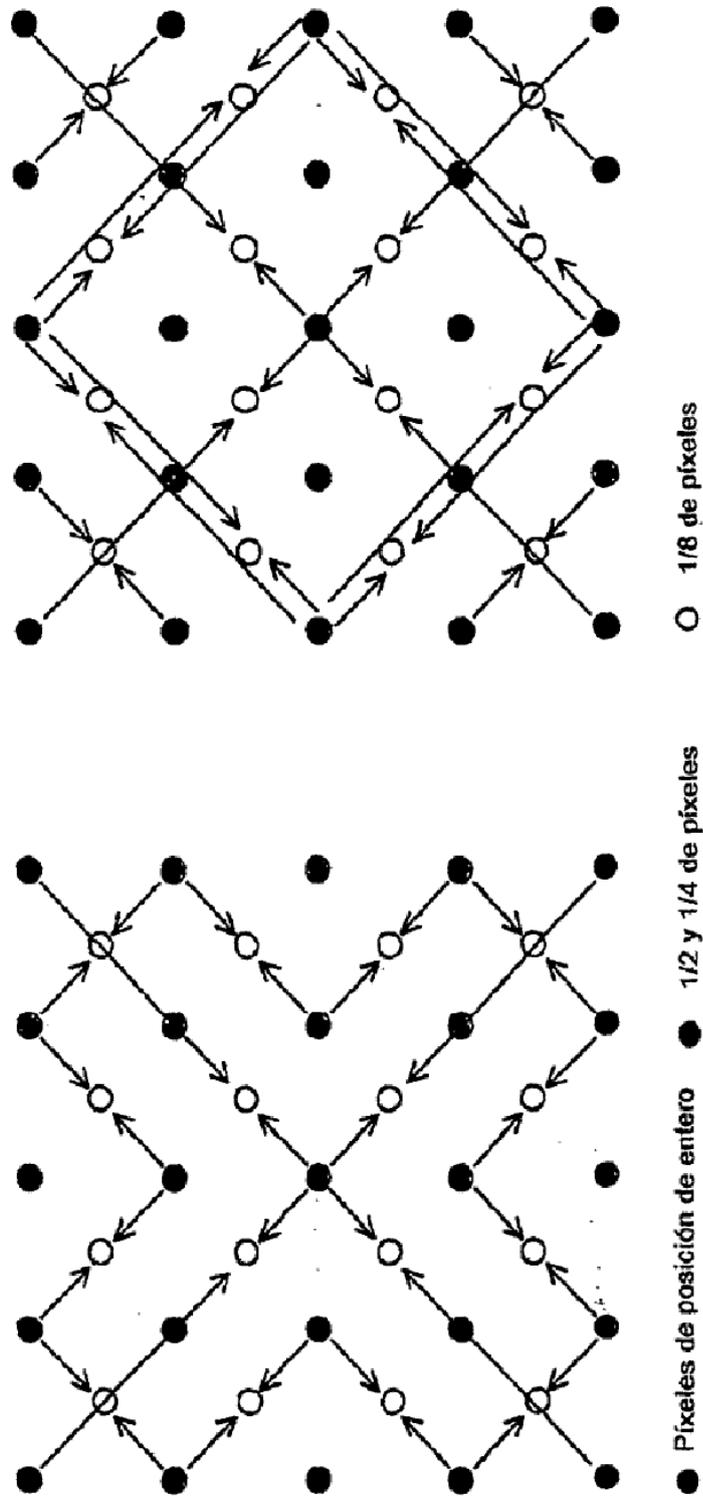


Fig. 21

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Fig. 22