

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 773 751**

51 Int. Cl.:

H04N 19/117 (2014.01)
H04N 19/12 (2014.01)
H04N 19/146 (2014.01)
H04N 19/159 (2014.01)
H04N 19/176 (2014.01)
H04N 19/61 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **02.09.2004 PCT/US2004/028969**

87 Fecha y número de publicación internacional: **24.03.2005 WO05027492**

96 Fecha de presentación y número de la solicitud europea: **02.09.2004 E 04783270 (4)**

97 Fecha y número de publicación de la concesión europea: **20.11.2019 EP 1661387**

54 Título: **Transformación solapada condicional**

30 Prioridad:

07.09.2003 US 501081 P
31.08.2004 US 931885

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
14.07.2020

73 Titular/es:

MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
One Microsoft Way
Redmond, WA 98052, US

72 Inventor/es:

HSU, POHSIANG y
SRINIVASAN, SRIDHAR

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 773 751 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

Transformación solapada condicional

Campo técnico

5 La presente invención se refiere a técnicas para codificar, decodificar y procesar digitalmente señales de contenido multimedia (por ejemplo, audio, vídeo, imagen, texto, voz, etc.). Más particularmente, la invención se refiere a códecs de medios digitales basados en transformaciones solapadas.

Antecedentes

10 La codificación de transformación es una técnica de compresión utilizada en muchos sistemas de compresión de audio, imagen y vídeo. Normalmente, la imagen y el vídeo digitales no comprimidos se representan o capturan como muestras de elementos de imagen o colores en ubicaciones en un fotograma de imagen o vídeo dispuesto en una cuadrícula bidimensional. Por ejemplo, un formato típico para las imágenes consiste en una secuencia de muestras de elementos de imagen en color de 24 bits dispuestas como una cuadrícula. Cada muestra es un número que representa componentes de color en una localización de píxel en la cuadrícula dentro de un espacio de color, como RGB o YIQ, entre otros. Diversos sistemas de imagen y vídeo pueden utilizar diferentes resoluciones de muestreo de color, espacio y tiempo.

15 Las señales de imagen y vídeo digitales no comprimidas pueden consumir una capacidad de almacenamiento y transmisión considerable. La codificación de transformación reduce el tamaño de las imágenes digitales y del vídeo al transformar la representación de dominio espacial de la señal en una representación de dominio de frecuencia (u otro dominio de transformación similar), y luego reducir la resolución de ciertos componentes de frecuencia generalmente menos perceptibles de la representación del dominio de transformación. Por lo general, esto produce una degradación mucho menos perceptible de la señal digital en comparación con la reducción de color o resolución espacial de imágenes o vídeo en el dominio espacial.

20 Más específicamente, la técnica 100 típica de codificación de transformación que se muestra en la Figura 1 divide los píxeles de la imagen digital no comprimida en bloques bidimensionales de tamaño fijo, y cada bloque posiblemente se superpone con otros bloques. A cada bloque se le aplica una transformación 110 lineal, que hace un análisis de frecuencia espacial, que convierte las muestras espaciadas dentro del bloque en un conjunto de coeficientes de frecuencia (o transformación) que generalmente representan la intensidad de la señal digital en las bandas de frecuencia correspondientes en el intervalo de bloque. Para la compresión, los coeficientes de transformación pueden cuantificarse selectivamente (es decir, reducirse en resolución, por ejemplo mediante la caída de los bits menos significativos de los valores de los coeficientes o mapeando de otra manera los valores en un número de resolución más alto ajustado a una resolución más baja), y también la entropía o la longitud variable codificada en un flujo de datos comprimido por el cuantificador/codificador 120 de entropía. Después de la descuantificación/ decodificación 130 de entropía, los coeficientes de transformación transformarán inversamente 140 para casi reconstruir la señal de vídeo/imagen muestreada espacial/con color original.

35 Mientras comprime una imagen fija (o un fotograma intra-codificada en una secuencia de vídeo), los estándares más comunes como MPEG-2, MPEG-4 y Windows Media dividen la imagen en mosaicos y aplican una transformación de bloque a cada mosaico de imagen. Los coeficientes de transformación en una partición dada (comúnmente conocida como bloque) solo están influenciados por los componentes de datos en bruto dentro del bloque. Las operaciones irreversibles o con pérdidas del lado del codificador, tales como la cuantificación, hacen que aparezcan artefactos en la imagen decodificada. Estos artefactos son independientes entre bloques y producen un efecto visualmente molesto conocido como el efecto de bloqueo. Del mismo modo, en el caso de los datos de audio, cuando los bloques no superpuestos se codifican con independencia de la transformación los errores de cuantificación producirán discontinuidades de la señal en los límites de los bloques al reconstruirse la señal de audio en el decodificador. En el caso del audio, se oye un efecto de clic periódico.

45 Transformación solapada de dominio espacial

Con el fin de minimizar el efecto de bloqueo, se pueden explotar correlaciones cruzadas de bloques. Una forma de lograr la correlación cruzada de bloques es mediante el uso de una transformación solapada, como se describe en H. Malvar, "Signal Processing with Lapped Transforms", Artech House, Norwood MA, 1992. Una transformación solapada es una transformación cuya entrada abarca, además de los elementos de datos del bloque actual, algunos elementos adyacentes de bloques vecinos. Del mismo modo, en el lado de la reconstrucción, la transformación inversa influye en todos los puntos de datos del bloque actual, así como en algunos puntos de datos de bloques vecinos.

55 En el caso de los datos bidimensionales (2D), la transformación en 2D solapada es una función del bloque actual, junto con elementos seleccionados de bloques de la izquierda, de la parte superior, de la derecha, de la parte inferior y posiblemente de la parte superior izquierda, de la parte superior derecha, de la parte inferior izquierda y de la parte inferior derecha. El número de puntos de datos de bloques vecinos que se utilizan para calcular la transformación actual se conoce como superposición.

La transformación solapada se puede implementar en el dominio de la transformación como un paso que fusiona cantidades de dominio de transformación después de una transformación de bloque convencional. De lo contrario, se puede implementar en el dominio espacial mediante una etapa de preprocesamiento aplicada a los píxeles dentro del intervalo de superposición. Estas dos implementaciones están matemáticamente relacionadas y por lo tanto son equivalentes.

Como se muestra en la Figura 2, la transformación solapada de dominio espacial (SDLT) 200 es una transformación solapada que se implementa como pasos de pre y post-procesamiento correspondientes 210, 220 antes de la transformación 110 de bloque directa y con posterioridad a la transformación 140 de bloque inversa, respectivamente (véase, por ejemplo, Srinivasan et al. "Improvements to the Spatial-Domain Lapped Transform in Digital Media Compression," Solicitud de patente estadounidense nº 10/620,744, presentada el 15 de julio de 2003 [en lo sucesivo, "Solicitud de patente SDLT mejorada"]). La transformación solapada de dominio espacial se utiliza a menudo para readaptar las técnicas de compresión existentes basadas en transformaciones de bloques con el fin de mejorar la eficiencia.

Gangaputra S. et al: "Adaptive pre- and post-filtering for block based systems", 2002, IEEE, International Conference on Acoustics, Speech, and Signal Processing, Orlando, FL, 13 a 17 de mayo de 2002, Proceedings, New York, NY, EE.UU., vol. 4, 13 de mayo de 2002, págs. IV-3297 a IV-3300, que trata sobre el pre y post-filtrado adaptativo para los sistemas basados en bloques.

Compendio

El objeto de la presente invención es proporcionar un método y un sistema para codificar o decodificar un flujo de bits de vídeo utilizando una transformación solapada condicional.

Este objeto se resuelve por el contenido de las reivindicaciones independientes.

Las realizaciones se dan en las reivindicaciones dependientes.

Cuando la SDLT se aplica a todas las partes de una imagen (como un fotograma en una secuencia de vídeo) de otros contenidos de medios digitales codificados de una manera no predicha (es decir, Intra), surgen ciertos problemas limitantes:

1. El filtro previo amplía el intervalo; por lo tanto, ciertas áreas de la imagen pueden saturarse o recortarse después del filtro previo en el lado del codificador. Del mismo modo, puede haber desbordamiento en el lado del decodificador.

2. El filtro previo tiene un efecto de afilado. Por tanto, la eficiencia de codificación de ciertas áreas de una imagen con contenido local de alta frecuencia (especialmente los bordes de imagen alineados con los bordes del bloque) disminuirá como resultado de la transformación solapada.

La transformación solapada condicional que se implementa utilizando técnicas de procesamiento de señales de medios digitales y de codificación/decodificación y sistemas descritos en la presente memoria logra un mejor control del proceso de superposición para abordar estos problemas. Esta transformación solapada condicional proporciona flexibilidad espacial en la elección o aplicación de los filtros previos y filtros posteriores de una transformación solapada. En otras palabras, la transformación solapada se aplica condicionalmente a diferentes áreas del contenido de medios digitales (por ejemplo, bloques separados de una imagen).

Por ejemplo, una implementación de la transformación solapada condicional descrita en la presente memoria utiliza un parámetro de modo espacialmente variable para especificar la aplicación y el alcance del filtro posterior de superposición que se aplicará tras la decodificación. En un códec de vídeo, por ejemplo, este parámetro de modo espacialmente variable especifica a qué bordes de bloque de una imagen debe aplicarse el filtro posterior de la transformación solapada. En una implementación, el parámetro de modo se codifica y transmite de forma más eficiente para los fotogramas intracodificados de una secuencia de vídeo en forma de indicador booleano con granularidad de macrobloque, lo que resulta más eficiente que codificar el parámetro de modo por cada borde de bloque. Además, esta implementación impone una restricción en el modo de superposición condicional basado en un parámetro de cuantificación relacionado con la tasa de bits, con el fin de minimizar el impacto de realizar la transformación solapada condicional con tasas de bits bajas.

Las características y ventajas adicionales de la invención se harán evidentes a partir de la siguiente descripción detallada de las realizaciones que continúa con referencia a los dibujos adjuntos.

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques de una técnica típica de codificación por transformación en el estado de la técnica anterior.

La Figura 2 es un diagrama de bloques de un sistema de codificación/decodificación basado en la transformación solapada de dominio espacial en el estado de la técnica anterior.

La Figura 3 es un diagrama de bloques de un codificador de vídeo que emplea la transformación solapada condicional descrita en la presente memoria.

La Figura 4 es un diagrama de bloques de un decodificador de vídeo que emplea la transformación solapada condicional descrita en la presente memoria.

5 La Figura 5 es un diagrama de bloques de un codificador que implementa la transformación solapada condicional con señalización de modo de superposición por borde de bloque adyacente.

La Figura 6 es un diagrama sintáctico a nivel de secuencia utilizado en un codificador/decodificador de acuerdo con una segunda implementación de la transformación solapada condicional con señalización de modo de superposición basada en la tasa de bits y por macrobloques.

10 La Figura 7 es un diagrama sintáctico a nivel de fotograma utilizado en el codificador/decodificador de acuerdo con una segunda implementación de la transformación solapada condicional con señalización de modo de superposición basada en la tasa de bits y por macrobloques.

La Figura 8 es un diagrama de flujo que ilustra la señalización basada en la tasa de bits y por macrobloques del indicador de modo de superposición en una segunda implementación de la transformación solapada condicional.

15 La Figura 9 es un diagrama que ilustra los bordes de macrobloques adyacentes a los que se aplica el filtro de superposición para un ejemplo de indicadores de modo de superposición señalizados para un conjunto de macrobloques en una imagen.

La Figura 10 es un diagrama de bloques de un entorno informático adecuado para el codificador/decodificador de vídeo de las Figuras 3 y 4.

20 Descripción detallada

La siguiente descripción se refiere a las implementaciones de la transformación solapada condicional, en la que los filtros de borde de bloque asociados con la transformación solapada se aplican condicionalmente en una base espacialmente variable sobre el contenido de medios digitales. Un ejemplo de aplicación de la transformación solapada condicional es en un codificador y decodificador de imagen o vídeo. Sin embargo, esta forma condicional de transformación solapada de dominio espacial no se limita a códecs de imagen o vídeo, y se puede aplicar a otros sistemas de codificación, decodificación y/o procesamiento de medios. Por consiguiente, la transformación solapada condicional se describe en el contexto de un codificador y decodificador de imagen o vídeo generalizado, pero alternativamente se puede incorporar en varios tipos de sistemas de codificación, decodificación y/o procesamiento de señales multimedia (por ejemplo, audio, vídeo, imagen, etc.) que emplean filtros de superposición de transformaciones solapadas en una base condicional o espacialmente variable, ya sea en el dominio espacial o dominio de la transformación.

1. Codificador y decodificador de vídeo generalizado

La Figura 3 es un diagrama de bloques de un codificador (300) de vídeo generalizado y la Figura 4 es un diagrama de bloques de un decodificador (400) de vídeo generalizado en el que se pueden incorporar las transformaciones WMV9/VC-9.

Las relaciones mostradas entre módulos dentro del codificador y decodificador indican el flujo principal de información en el codificador y decodificador; no se muestran otras relaciones en aras de la simplicidad. En particular, las Figuras 3 y 4 generalmente no muestran información adicional sobre los ajustes del codificador, modos, tablas, etc. utilizados para una secuencia de vídeo, fotograma, macrobloque, bloque, etc. Dicha información adicional se envía en el flujo de bits de salida, típicamente después de la codificación de entropía de la información adicional. El formato del flujo de bits de salida puede ser un formato de vídeo de Windows Media u otro formato.

El codificador (300) y el decodificador (400) están basados en bloques y utilizan un formato de macrobloques 4:2:0 con cada macrobloque incluyendo 4 bloques de luminancia de luminancia 8x8 (a veces tratados como un macrobloque 16x16) y dos bloques de crominancia 8x8. Alternativamente, el codificador (300) y el decodificador (400) están basados en objetos, utilizan un formato de bloques o macrobloques diferente, o realizan operaciones en conjuntos de píxeles de diferente tamaño o configuración que los bloques de 8x8 y los macrobloques de 16x16.

Dependiendo de la implementación y del tipo de compresión deseada, pueden añadirse módulos del codificador o decodificador, omitirse, dividirse en múltiples módulos, combinarse con otros módulos, y/o reemplazarse por módulos similares. En realizaciones alternativas, los codificadores o decodificadores con diferentes módulos y/u otras configuraciones de módulos realizan una o más de las técnicas descritas.

A. Codificador de vídeo

La Figura 3 es un diagrama de bloques de un sistema (300) codificador de vídeo general. El sistema (300) codificador recibe una secuencia de fotogramas de vídeo que incluye un fotograma (305) actual y produce

información (395) de vídeo comprimido como salida. Las realizaciones particulares de codificadores de vídeo utilizan típicamente una variación o versión suplementada del codificador (300) generalizado.

El sistema (300) codificador comprime los fotogramas predichos y los fotogramas clave. Para facilitar la presentación, la Figura 3 muestra una trayectoria de los fotogramas clave a través del sistema (300) codificador y una trayectoria de los fotogramas predichos a futuro. Muchos de los componentes del sistema (300) codificador se utilizan para comprimir tanto los fotogramas clave como los fotogramas predichos. Las operaciones exactas realizadas por dichos componentes pueden variar en función del tipo de información que se comprime.

Un fotograma predicho [también llamado fotograma p, fotograma b en el caso de la predicción bidireccional o fotograma intercodificado] se representa en términos de predicción (o diferencia) de uno o más fotogramas diferentes. Una predicción residual es la diferencia entre lo que se predijo y el fotograma original. Por el contrario, un fotograma clave [también llamado fotograma i, fotograma intracodificado] se comprime sin referencia a otros fotogramas.

Si el fotograma (305) actual es un fotograma predictivo, un estimador (310) de movimiento estima el movimiento de macrobloques u otros conjuntos de píxeles del fotograma (305) actual con respecto a un fotograma de referencia, que es el fotograma (325) anterior reconstruido almacenado en la memoria (320) de fotograma. En realizaciones alternativas, el fotograma de referencia es un fotograma posterior o bien el fotograma actual se predice bidireccionalmente. El estimador (310) de movimiento produce como información adicional información (315) de movimiento, como por ejemplo vectores de movimiento. Un compensador (330) de movimiento aplica la información (315) de movimiento al fotograma (325) anterior reconstruido para formar un fotograma (335) actual compensado por movimiento. Sin embargo, la predicción rara vez es perfecta y la diferencia entre el fotograma (335) actual compensado por movimiento y el fotograma (305) actual original es el residuo (345) de la predicción. Alternativamente, un estimador de movimiento y un compensador de movimiento aplican otro tipo de estimación/compensación de movimiento.

Un transformador (360) de frecuencia convierte la información de vídeo del dominio espacial en datos del dominio de frecuencia (es decir, espectrales). Para los fotogramas de vídeo basados en bloques, el transformador (360) de frecuencia aplica una transformación, descrita en las siguientes secciones, que tiene propiedades similares a la transformación de coseno discreto ["DCT", por sus siglas en inglés]. En algunas realizaciones, el transformador (360) de frecuencia aplica una transformación de frecuencia a bloques de residuos de predicción espacial para fotogramas clave. El transformador (360) de frecuencia puede aplicar una transformación de frecuencia de 8x8, 8x4, 4x8, o de otro tamaño.

Un cuantificador (370) cuantifica a continuación los bloques de coeficientes de datos espectrales. El cuantificador aplica una cuantificación uniforme y escalar a los datos espectrales con un tamaño de paso que varía fotograma a fotograma o de otra forma. Alternativamente, el cuantificador aplica otro tipo de cuantificación a los coeficientes de datos espectrales, por ejemplo, una cuantificación no uniforme, vectorial o no adaptativa, o bien cuantifica directamente los datos de dominio espacial en un sistema codificador que no utiliza transformaciones de frecuencia. Además de la cuantificación adaptativa, el codificador (300) puede utilizar caídas de fotogramas, filtrado adaptativo u otras técnicas para el control de la tasa.

Cuando se necesita un fotograma actual reconstruido para la estimación/compensación de movimiento subsiguiente, un cuantificador (376) inverso realiza una cuantificación inversa sobre los coeficientes de datos espectrales cuantificados. Un transformador (366) de frecuencia inversa realiza entonces la inversa de las operaciones del transformador (360) de frecuencia, produciendo un residuo de predicción reconstruido (para un fotograma predicho) o un fotograma clave reconstruido. Si el fotograma (305) actual era un fotograma clave, el fotograma clave reconstruido se toma como el fotograma actual reconstruido (no se muestra). Si el fotograma (305) actual era un fotograma predicho, el residuo de predicción reconstruido se añade al fotograma (335) actual compensado por movimiento para formar el fotograma actual reconstruido. La memoria (320) de fotograma almacena el fotograma actual reconstruido para su uso en la predicción del siguiente fotograma. En algunas realizaciones, el codificador aplica un filtro de desbloqueo al fotograma reconstruido para suavizar de forma adaptativa las discontinuidades en los bloques del fotograma.

El codificador (380) de entropía comprime la salida del cuantificador (370) así como cierta información adicional (por ejemplo, información de movimiento (315), tamaño de paso de cuantificación). Entre las técnicas típicas de codificación de entropía se incluyen la codificación aritmética, la codificación diferencial, la codificación de Huffman, la codificación de longitud de recorrido la codificación LZ, la codificación por diccionario y combinaciones de las anteriores. El codificador (380) de entropía utiliza típicamente diferentes técnicas de codificación para diferentes tipos de información (por ejemplo, coeficientes DC, coeficientes AC, diferentes tipos de información adicional), y puede elegir entre múltiples tablas de códigos dentro de una técnica de codificación particular.

El codificador (380) de entropía coloca la información de vídeo comprimida (395) en la memoria intermedia (390). Un indicador de nivel de la memoria intermedia se realimenta a los módulos adaptativos de tasa de bits. La información (395) de vídeo comprimida se agota desde la memoria intermedia (390) con una tasa de bits constante o

relativamente constante y se almacena para su posterior transmisión a esa tasa de bits. Alternativamente, el sistema (300) codificador transmite información de vídeo comprimida inmediatamente después de la compresión.

Antes o después de la memoria intermedia (390), la información (395) de vídeo comprimida puede codificarse por canal para su transmisión a través de la red. La codificación de canal puede aplicar datos de detección y corrección de errores a la información (395) de vídeo comprimida.

B. Decodificador de vídeo

La Figura 4 es un diagrama de bloques de un sistema (400) decodificador de vídeo general. El sistema (400) decodificador recibe información (495) para una secuencia comprimida de fotogramas de vídeo y genera una salida que incluye un fotograma (405) reconstruido. Las realizaciones particulares de decodificadores de vídeo utilizan típicamente una variación o versión suplementada del codificador (400) generalizado.

El sistema (400) decodificador descomprime los fotogramas predichos y los fotogramas clave. Para facilitar la presentación, la Figura 4 muestra una trayectoria de los fotogramas clave a través del sistema (400) decodificador y una trayectoria de los fotogramas predictivos. Muchos de los componentes del sistema (400) decodificador se utilizan para comprimir tanto los fotogramas clave como los fotogramas predichos. Las operaciones exactas realizadas por dichos componentes pueden variar en función del tipo de información que se comprime.

Una memoria intermedia (490) recibe la información (495) para la secuencia de vídeo comprimida y pone la información recibida a disposición del decodificador (480) de entropía. La memoria intermedia (490) típicamente recibe la información a una tasa que es bastante constante con el tiempo, e incluye una memoria intermedia de fluctuaciones para suavizar las variaciones a corto plazo en el ancho de banda o transmisión. La memoria intermedia (490) puede incluir una memoria intermedia de reproducción y también otras memorias intermedias. Alternativamente, la memoria intermedia (490) recibe información a una tasa variable. Antes o después de la memoria intermedia (490), la información de vídeo comprimida puede decodificarse por canal y procesarse para la detección y corrección de errores.

El decodificador (480) de entropía decodifica datos cuantificados codificados por entropía así como información adicional codificada por entropía (por ejemplo, información de movimiento, tamaño de paso de cuantificación), aplicando típicamente la inversa de la codificación de entropía realizada en el codificador. Entre las técnicas de decodificación de entropía se incluyen la decodificación aritmética, la decodificación diferencial, la decodificación de Huffman, la decodificación de longitud de recorrido la decodificación LZ, la decodificación por diccionario y combinaciones de las anteriores. El decodificador (480) de entropía utiliza con frecuencia diferentes técnicas de decodificación para diferentes tipos de información (por ejemplo, coeficientes DC, coeficientes AC, diferentes tipos de información adicional), y puede elegir entre múltiples tablas de códigos dentro de una técnica de decodificación particular.

Si el fotograma (405) a reconstruir es un fotograma predictivo, un compensador (430) de movimiento aplica información (415) de movimiento a un fotograma (425) de referencia para formar una predicción (435) del fotograma (405) que se está reconstruyendo. Por ejemplo, el compensador (430) de movimiento utiliza un vector de movimiento de macrobloque para encontrar un macrobloque en el fotograma (425) de referencia. Una memoria intermedia (420) de fotogramas almacena fotogramas reconstruidos previos para su uso como fotogramas de referencia. Alternativamente, un compensador de movimiento aplica otro tipo de compensación de movimiento. La predicción del compensador de movimiento rara vez es perfecta, por lo que el decodificador (400) también reconstruye los residuos de la predicción.

Cuando el decodificador necesita un fotograma reconstruido para la compensación de movimiento posterior, la memoria (420) de fotograma almacena el fotograma reconstruido para su uso en la predicción del siguiente fotograma. En algunas realizaciones, el codificador aplica un filtro de desbloqueo al fotograma reconstruido para suavizar las discontinuidades adaptativamente en los bloques del fotograma.

Un cuantificador inverso (470) cuantifica inversamente los datos decodificados por entropía. En general, el cuantificador inverso aplica una cuantificación uniforme y escalar inversa a los datos decodificados por entropía con un tamaño de paso que varía fotograma a fotograma o de otra forma. Alternativamente, el cuantificador inverso aplica otro tipo de cuantificación inversa a los datos, por ejemplo, una cuantificación no uniforme, vectorial o no adaptativa, o bien cuantifica de forma inversa directamente los datos de dominio espacial en un sistema decodificador que no utiliza transformaciones de frecuencia inversa.

Un transformador (460) de frecuencia inversa convierte los datos cuantificados del dominio de frecuencia en información de vídeo del dominio espacial. Para los fotogramas de vídeo basados en bloques, el transformador (460) de frecuencia inversa aplica una transformación inversa, que se describe en las siguientes secciones. En algunas realizaciones, el transformador (460) de frecuencia inversa aplica una transformación de frecuencia inversa a bloques de residuos de predicción espacial para fotogramas clave. El transformador (460) de frecuencia inversa puede aplicar una transformación de frecuencia de 8x8, 8x4, 4x8, o de otro tamaño.

2. Transformación solapada condicional

Con referencia nuevamente a la Figura 2, las etapas de filtro de preprocesamiento y post-procesamiento 210, 220 se aplican a todos los bordes de bloques adyacentes de una imagen cuando se realiza en la imagen la transformación solapada de dominio espacial anterior típica mediante un codificador/decodificador 200. Una imagen de tamaño de $M \times N$ píxeles, codificada usando bloques de 8×8 , tiene casi $M \cdot N / 32$ bordes distintos entre bloques adyacentes de 8×8 solo para su canal de luminancia. Cuando el codificador/decodificador 200 de la técnica anterior aplica la transformación solapada de dominio espacial a una imagen, las etapas 210, 220 de filtro se aplican a cada uno de estos bordes de bloques adyacentes de la imagen.

A. Señalización de modo de superposición por borde de bloque adyacente

En una primera implementación de la transformación solapada condicional mostrada en la Figura 5, el codificador envía un símbolo distinto (símbolo 510 de modo de superposición) correspondiente a cada borde de bloque adyacente, definiendo la elección del filtro de superposición (M_n) 520-522 (que puede ser un filtro de paso, es decir, no se aplica filtrado), como se muestra en la Figura 5 para el codificador. En el codificador, la elección del filtro se puede hacer en función de varios criterios, incluyendo si algún resultado del filtro está fuera del intervalo, una prueba de suavidad, un criterio de optimización de distorsión de tasa y combinaciones de estos criterios, entre otros. Este interruptor se conoce como el "modo" de filtro de superposición, y el modo determina una "condición" bajo la cual se aplican los filtros previos y posteriores. La Figura 5 muestra un parámetro de modo de superposición que determina la configuración del interruptor, y que se envía para cada borde del bloque que se filtra.

Una desventaja de permitir que cada filtro previo y posterior se marque independientemente es el aumento de la sobrecarga asociada con el modo de filtrado. Por ejemplo, supongamos que el vídeo de tamaño 320×240 está siendo codificado a 30 f/s. Además, supongamos que se envía un fotograma I (fotograma intracodificado) cada segundo. Por lo tanto, hay alrededor de 3600 bordes intrabloques 8×8 en el vídeo cada segundo. (Este número es aproximado, porque algunos bloques pueden estar en los límites de la imagen, y por lo tanto no son bordes de bloques adyacentes). Incluso cuando se envía un símbolo de modo de superposición de un solo bit (una sobrecarga de 1 bit) para cada borde, esto da como resultado un exceso de tasa de bits de 3,6 kbps en el flujo de bits codificado (información 195 de vídeo comprimida de la Figura 3), lo cual no es insignificante.

B. Señalización de modo de superposición basada en la tasa de bits y por bloques

En otro ejemplo de implementación, el códec logra una menor sobrecarga para señalar el modo de superposición usando un esquema de señalización que se basa en la tasa de bits de codificación y utiliza un símbolo de modo de superposición por bloque. Este esquema de señalización utiliza los siguientes elementos sintácticos: un parámetro de cuantificación (QP) a nivel de fotograma, un indicador de superposición de nivel de fotograma y un indicador de modo de superposición a nivel de macrobloque. En códecs que utilizan otra sintaxis de codificación de vídeo, el modo de transformación de superposición condicional para bordes de bloque individuales, o grupos de bordes de bloque, se puede codificar utilizando otros esquemas y elementos sintácticos. Por ejemplo, se puede utilizar otro elemento sintáctico relacionado con la tasa de bits de codificación o la cuantificación del fotograma.

Con referencia a las Figuras 6 y 7, el flujo 195 de bits de vídeo comprimido (Figura 3) incluye información para una secuencia de fotogramas de vídeo progresivos comprimidos u otras imágenes. El flujo de bits está organizado en varias capas jerárquicas que son decodificadas por un decodificador como el decodificador (400) de la Figura 4. La capa más alta es la capa de secuencia, que tiene información para la secuencia general de fotogramas. Además, cada fotograma de vídeo comprimido está formado por datos estructurados en tres capas jerárquicas. De arriba a abajo las capas son: imagen, macrobloque y bloque.

La Figura 6 es un diagrama sintáctico para la capa 600 de secuencia, que incluye un encabezado 610 de secuencia seguido de datos para la capa de imagen (véase la Figura 7). El encabezado 610 de secuencia incluye varios elementos a nivel de secuencia procesados por el decodificador y utilizados para decodificar la secuencia, incluyendo un elemento 620 de cuantificación de macrobloques (DQUANT), un elemento 630 de especificación de cuantificador (QUANTIZER) y un elemento 640 de indicador de transformación superpuesta (OVERLAP). DQUANT 620 es un campo de 2 bits que indica si el tamaño del paso de cuantificación puede variar o no dentro de un fotograma. Existen tres valores posibles para DQUANT. Si $DQUANT = 0$, entonces solo se puede utilizar un único tamaño de paso de cuantificación (es decir, el tamaño del paso de cuantificación de fotogramas) por fotograma. Si $DQUANT = 1$ o 2 , entonces es posible cuantificar de forma diferente cada uno de los macrobloques en el fotograma.

El QUANTIZER 630 es un campo de código de longitud fija ["FLC"] de 2 bits que indica el cuantificador utilizado para la secuencia. Los tipos de cuantificador se codifican de acuerdo con la siguiente Tabla 1.

Tabla 1: Especificación del cuantificador

FLC	Especificación del cuantificador
00	Cuantificador especificado implícitamente a nivel de fotograma

01	Cuantificador especificado explícitamente a nivel de fotograma
10	Cuantificador de zonas muertas 5 QP utilizado para todos los fotogramas
11	Cuantificador de zonas muertas 3 QP utilizado para todos los fotogramas

Indicador de transformación superpuesta (OVERLAP) (1 bit)

El OVERLAP 640 es un indicador de 1 bit que indica si se utilizan transformaciones superpuestas, como se discute más adelante. Si OVERLAP = 1, entonces se utilizan transformaciones superpuestas, de lo contrario no se utilizan.

5 La Figura 7 es un diagrama sintáctico para la capa 700 de imagen para un fotograma intra-codificado entrelazado ["fotograma I entrelazado"]. Los diagramas sintácticos para otras imágenes, como los fotogramas I progresivos, las imágenes P y los fotogramas B tienen muchos elementos sintácticos similares. La capa 700 de imagen incluye un encabezado 710 de imagen seguido de datos para la capa de macrobloque. El encabezado 710 de imagen incluye
10 varios elementos a nivel de imagen procesados por el decodificador y utilizados para decodificar el fotograma correspondiente. Algunos de esos elementos solo están presentes si su existencia está señalada o implicada por un elemento de nivel de secuencia o un elemento de nivel de imagen anterior.

En la sintaxis ilustrada del flujo de bits, el parámetro de cuantificación (QP) a nivel de fotograma se presenta en forma de escala cuantificadora de imágenes (PQUANT), la cual puede ser señalada implícita o explícitamente en la sintaxis de la capa de imagen, según lo especificado por el elemento sintáctico a nivel de secuencia, el CUANTIFICADOR (QUANTIZER) descrito anteriormente. En cualquier caso, la escala cuantificadora de imágenes (PQUANT) se traduce a partir de un elemento 720 del índice del cuantificador de imágenes (PQINDEX). PQINDEX 720 es un campo de 5 bits que indica el índice de escala cuantificadora para todo el fotograma. Está presente en todos los tipos de imágenes. Si se utiliza el cuantificador implícito, entonces PQINDEX especifica tanto la escala cuantificadora de imágenes (PQUANT) como el cuantificador (zona muerta 3QP o 5QP) utilizado para el fotograma.
15 La Tabla 2 muestra cómo PQINDEX se traduce a PQUANT y el cuantificador para el modo implícito.
20

Tabla 2: Traducción de PQINDEX a PQUANT/cuantificador de zonas muertas (cuantificador implícito)

PQINDEX	PQUANT	Cuantificador de zonas muertas	PQINDEX	PQUANT	Cuantificador de zonas muertas
0	NA	NA	16	13	5 QP
1	1	3 QP	17	14	5 QP
2	2	3 QP	18	15	5 QP
3	3	3 QP	19	16	5 QP
4	4	3 QP	20	17	5 QP
5	5	3 QP	21	18	5 QP
6	6	3 QP	22	19	5 QP
7	7	3 QP	23	20	5 QP
8	8	3 QP	24	21	5 QP
9	6	5 QP	25	22	5 QP
10	7	5 QP	26	23	5 QP
11	8	5 QP	27	24	5 QP
12	9	5 QP	28	25	5 QP
13	10	5 QP	29	27	5 QP
14	11	5 QP	30	29	5 QP
15	12	5 QP	31	31	5 QP

5 Si el cuantificador se señala explícitamente a nivel de secuencia o fotograma, entonces PQINDEX se traduce al tamaño del paso del cuantificador de imágenes PQUANT, como se indica en la Tabla 3.

Tabla 3: Traducción de PQINDEX a PQUANT (cuantificador explícito)

PQINDEX	PQUANT 3QP Zona muerta	PQUANT 5QP Zona muerta	PQINDEX	PQUANT 3QP Zona muerta	PQUANT 5QP Zona muerta
0	NA	NA	16	16	14
1	1	1	17	17	15
2	2	1	18	18	16
3	3	1	19	19	17
4	4	2	20	20	18
5	5	3	21	21	19
6	6	4	22	22	20
7	7	5	23	23	21
8	8	6	24	24	22
9	9	7	25	25	23

PQINDEX	PQUANT 3QP Zona muerta	PQUANT 5QP Zona muerta	PQINDEX	PQUANT 3QP Zona muerta	PQUANT 5QP Zona muerta
10	10	8	26	26	24
11	11	9	27	27	25
12	12	10	28	28	26
13	13	11	29	29	27
14	14	12	30	30	29
15	15	13	31	31	31

Alternativamente, en lugar de la traducción mostrada en la Tabla 3, PQUANT es igual a PQINDEX para todos los valores de PQINDEX del 1 al 31 cuando el cuantificador se señala explícitamente a nivel de secuencia o fotograma.

5 La cabecera de imagen incluye además un elemento 730 de indicador de superposición condicional (CONDOVER) y un elemento 740 de indicadores de patrón de macrobloque de superposición condicional (OVERFLAGS). Este elemento de indicador de solapamiento condicional solo está presente en imágenes I, y solo cuando OVERLAP está activado y PQUANT está dentro de un cierto intervalo, como se discute más adelante. El elemento sintáctico SOVERFLAGS solo está presente en las imágenes I, y solo cuando CONDOVER tiene el valor binario 11.
10 OVERFLAGS está codificado como un plano de bits, que en modo en bruto requiere que cada macrobloque lleve su información local, OVERFLAGMB.

Reglas del modo de superposición

15 En este segundo ejemplo de implementación de la transformación solapada condicional, el codificador (Figura 5) limita la sobrecarga colocando algunas restricciones en el indicador de modo de superposición en entre fotogramas como se indica a continuación:

1. El modo de superposición está configurado para ser una cantidad binaria. El valor FALSO (FALSE) indica que no se aplica el filtrado de superposición (es decir, los filtros previos y filtros posteriores son filtros de paso) y VERDADERO (TRUE) indica que las etapas de filtrado de superposición se aplican tal como se definen para la transformación solapada de dominio espacial en la solicitud de patente de SDLT mejorada mencionada anteriormente. En otras implementaciones, la cantidad de modos de filtro de superposición puede ser mayor como se ilustra en la Figura 5, a expensas de una sobrecarga adicional por macrobloque.

2. Para determinados valores de un parámetro de cuantificación (QP) basado en fotogramas (por ejemplo, la escala cuantificadora de imágenes (PQUANT) determinada a partir de la tabla 2 o 3) mayores que un umbral de cuantificación (QTH) o $QP > QTH$ (es decir, para tasas de bits de bajas a muy bajas), la condición de superposición se establece en VERDADERO (TRUE) para todos los bordes intrabloques. En la implementación ilustrada, se utiliza un umbral de cuantificación de 8, pero las implementaciones alternativas pueden utilizar un umbral de cuantificación diferente.

3. De lo contrario, para ciertos valores de parámetros de cuantificación basados en fotogramas $QP \leq QTH$ (es decir, para tasas de bits de altas a muy altas), cada intramacrobloques de un fotograma se asocia con un indicador de modo de superposición binaria. Las reglas de filtro de superposición asociadas con este indicador se enumeran más adelante.

Para los intrabloques y macrobloques en fotogramas con codificación predictiva (fotogramas P), el modo de superposición se deriva implícitamente a partir del parámetro de cuantificación (QP) únicamente. Esto es razonable, ya que los fotogramas P permiten un mayor grado de libertad en la elección del modo de codificación que los fotogramas I.

Como se muestra en el diagrama de flujo de la Figura 8, el indicador de modo de superposición se codifica y decodifica en esta segunda implementación de la siguiente manera:

1. No se envía ningún modo de superposición y no se realiza ninguna superposición (850) si el indicador de superposición a nivel de secuencia (OVERLAP 640 en la Figura 6) del primer bloque 810 de decisión se establece en FALSO (FALSE).

2. El indicador de modo de superposición es implícitamente VERDADERO (TRUE) cuando $QP > QTH$ en el bloque de decisión 820, y el indicador de superposición de nivel de secuencia (OVERLAP 640) es TRUE en el bloque 810

de decisión. En este caso, la superposición se realiza para todos los bordes de bloque adyacentes en la imagen (870).

3. Cuando $QP \leq QTH$ en el bloque 820 de decisión, se envía un indicador a nivel de fotograma (CONDOVER 730 en la Figura 7) que indica una de las tres posibilidades (a, b o c) y las operaciones se llevan a cabo tal como se indica:

- 5 a. Sin superposición para el fotograma (850)
- b. Todos los macrobloques superpuestos en el fotograma (870)
- c. "Superposición condicional" - por indicadores de modo de superposición de macrobloques (OVERFLAGS 740 en la Figura 7) enviados (860).

Esto se divide en dos decisiones binarias en el diagrama de flujo para facilitar su comprensión.

- 10 4. Cuando se indica superposición condicional, el indicador de modo de superposición es un indicador binario que se envía para cada intramacrobloques de un intrafotograma (es decir, el elemento sintáctico OVERFLAGS).

Superposición condicional

15 Dado que el modo de superposición se señala por macrobloque, se hace referencia a múltiples bordes por cada indicador de superposición condicional. En la implementación ilustrada, el tamaño del macrobloque es de 16x16 píxeles (para el canal de luminancia), y el tamaño del bloque de transformación es de 8x8 píxeles. El canal de crominancia tiene la mitad de resolución espacial para el tamaño de bloque de 8x8. Por consiguiente, los bordes pueden ser internos a un macrobloque (es decir, los bordes entre cada uno de los bloques de transformación en un macrobloque para el canal de luminancia), o a lo largo de dos macrobloques (para alrededor de la mitad de los bordes del canal de luminancia, y todos los bordes del canal de crominancia). Tal como se ilustra mediante el ejemplo de los indicadores de modo de superposición señalizados para los macrobloques ilustrados en la Figura 9, a 20 continuación se enumeran las reglas para determinar si se debe aplicar o no el filtro previo/posterior a un borde:

- 1. No se aplica ningún filtrado previo o posterior a los bordes de macrobloque internos de un macrobloque cuyo indicador de modo de superposición es FALSO (FALSE), o de un macrobloque que no sea intra-codificado.
- 25 2. Todos los bordes internos intrabloques de 8x8 de un macrobloque se filtran antes y después si el macrobloque es intra-codificado, y el indicador de modo de superposición correspondiente es VERDADERO (TRUE).
- 3. Los bordes de bloque de 8x8 a lo largo de dos macrobloques se filtran solo si
 - a. Los dos bordes de bloque 8x8 son intrabloques, y
 - b. Ambos indicadores de modo de superposición de los dos macrobloques son VERDADEROS (TRUE).

Señalización de modo de superposición

30 Todos los macrobloques de intrafotogramas en esta segunda implementación ilustrada de la transformación solapada condicional son intra. Dado que el modo de superposición se envía en función del parámetro de cuantificación QP de nivel de fotograma, el indicador de modo de superposición se envía para todos los macrobloques, o para ningún macrobloque. Cuando se envía para todos los macrobloques, los indicadores de modo de superposición constituyen un plano de bits de tamaño $(M/16) \times (N/16)$ para una imagen MxN (se asume que M, N son múltiplos de 16).

40 La implementación de transformación solapada condicional ilustrada utiliza un mecanismo de codificación eficiente (denominado codificación de "plano de bits", descrito a continuación) para enviar los indicadores de modo de superposición por macrobloques. Esto se basa en el supuesto de que la representación del plano de bits de los indicadores de modo de superposición tiene un alto grado de correlación espacial, es decir, es muy probable que los bits adyacentes en la dirección vertical u horizontal tengan el mismo valor. Esta suposición es cierta para el indicador de superposición - es razonable esperar que para las imágenes y el vídeo del mundo real, haya una continuidad espacial suficiente que también sea aparente en los indicadores de modo de superposición.

45 El mecanismo de codificación del plano de bits (descrito más detalladamente a continuación) se utiliza para codificar la señalización en modo de superposición para intrafotogramas. Los indicadores de modo de superposición para todo el fotograma se codifican conjuntamente a nivel de fotograma utilizando uno de una colección de mosaicos y tablas de códigos, excepto en el caso en que están codificados "en bruto" en forma de un bit por macrobloque enviado junto con el encabezado de macrobloque.

Codificación de plano de bits

50 En la codificación de plano de bits, la información binaria específica de un macrobloque, como los indicadores de superposición condicional por macrobloque, puede codificarse en un símbolo binario por macrobloque. En estos

casos, el estado de todos los macrobloques en un campo o fotograma puede codificarse como un plano de bits y transmitirse en el encabezado de campo o de fotograma. Una excepción a esta regla es que el modo de codificación del plano de bits se establezca en modo en Bruto en cuyo caso el estado de cada macrobloque se codifica como un bit por símbolo y se transmite junto con otros elementos sintácticos de nivel de macrobloque a nivel de macrobloque.

- 5 La codificación de plano de bits a nivel de campo/fotograma se utiliza para codificar matrices binarias bidimensionales. El tamaño de cada matriz es rowMB x colMB, donde rowMB y colMB son el número de filas y columnas de macrobloque, respectivamente, en el campo o fotograma en cuestión. Dentro del flujo de bits, cada matriz está codificada como un conjunto de bits consecutivos. Se utiliza uno de siete modos para codificar cada matriz. Los siete modos son:
- 10 1. Modo en Bruto : información codificada como un bit por símbolo y transmitida como parte de la sintaxis de nivel MB;
2. Modo normal-2: dos símbolos codificados conjuntamente;
3. Modo diferencial-2 - codificación diferencial del plano de bits, seguida de la codificación conjunta de dos símbolos residuales;
- 15 4. Modo normal-6: seis símbolos codificados conjuntamente;
5. Modo diferencial-6 - codificación diferencial del plano de bits, seguida de la codificación conjunta de seis símbolos residuales;
6. Modo salto de fila - salto de un bit a las filas de señal sin bits configurados; y
7. Modo salto de columna : salto de un bit a las columnas de señal sin bits configurado.

20 Los elementos sintácticos para un plano de bits a nivel de campo o de fotograma siguen la siguiente secuencia: INVERT, IMODE y DATABITS.

Indicador de inversión (INVERT)

25 El elemento sintáctico INVERT es un valor de 1 bit, que si se establece indica que el plano de bits tiene más bits configurados que bits cero. Según INVERT y el modo, el decodificador invertirá el plano de bits interpretado para recrear el original. Téngase en cuenta que el valor de este bit se ignorará cuando se utilice el modo en bruto . A continuación se proporciona una descripción de cómo se utiliza el valor INVERT en la decodificación del plano de bits.

Modo de codificación (IMODE)

30 El elemento sintáctico IMODE es un valor de longitud variable que indica el modo de codificación utilizado para codificar el plano de bits. La Tabla 4 muestra la tabla de códigos utilizada para codificar el elemento sintáctico IMODE. A continuación se proporciona una descripción de cómo se utiliza el valor IMODE en la decodificación del plano de bits.

Tabla 4: Tabla de códigos IMODE VLC

IMODE VLC	Modo de codificación
10	Norm-2
11	Norm-6
010	Salto de fila
011	Salto de columna
001	Diff-2
0001	Diff-6
0000	En bruto

35 Bits de codificación de plano de bits (DATABITS)

El elemento sintáctico DATABITS es un elemento sintáctico de tamaño variable que codifica el flujo de símbolos para el plano de bits. El método utilizado para codificar el plano de bits viene determinado por el valor de IMODE. Los siete modos de codificación están descritos en las siguientes secciones

Modo en bruto

5 En este modo, el plano de bits se codifica como un bit por símbolo explorado en el orden de exploración por trama de los macrobloques, y se envía como parte de la capa de macrobloques. Alternativamente, la información se codifica en modo en bruto a nivel de campo o de fotograma y DATABITS tiene una longitud de rowMB x colMB bits.

Modo Normal-2

10 Si rowMB x colMB es impar, el primer símbolo se codifica en bruto. Los símbolos posteriores se codifican por pares, en el orden natural de la exploración. La tabla 5 VLC binaria en la Tabla 5 se utiliza para codificar pares de símbolos.

Tabla 5: Tabla de códigos Norm-2/Diff-2

Símbolo 2n	Símbolo 2n + 1	Palabra-código
0	0	0
1	0	100
0	1	101
1	1	11

Modo Diff-2

15 El método Normal-2 se utiliza para producir el plano de bits como se describió anteriormente, y luego la operación Diff⁻¹ se aplica al plano de bits como se describe a continuación.

Modo Normal-6

20 En los modos Norm-6 y Diff-6, el plano de bits se codifica en grupos de seis píxeles. Estos píxeles se agrupan en mosaicos de 2x3 o 3x2. El plano de bits está revestido al máximo con mosaicos usando un conjunto de reglas, y los píxeles restantes se codifican usando una variante de los modos salto de fila y salto de columna. Los mosaicos "verticales" 2x3 se utilizan si y solo si rowMB es un múltiplo de 3 y colMB no lo es. De lo contrario, se utilizan mosaicos "horizontales" de 3x2. En el caso de un plano revestido con mosaicos lineales a lo largo de los bordes superior e izquierdo de la imagen, el orden de codificación de los mosaicos sigue el siguiente patrón. Los mosaicos de 6 elementos se codifican primero, y a continuación los mosaicos lineales codificados de salto de columna y salto de fila. Si el tamaño de la matriz es un múltiplo de 2x3 o de 3x2, estos últimos mosaicos lineales no existen y el plano de bits estará perfectamente revestido de mosaicos.

25 Los mosaicos rectangulares de 6 elementos se codifican usando un código de prefijo incompleto, es decir, un código de prefijo que no utiliza todos los nodos finales para la codificación. Supongamos que N es el número de bits establecidos en el mosaico, es decir, $0 \leq N \leq 6$. Para $N < 3$, se utiliza un VLC para codificar el mosaico. Para $N = 3$, un escape de longitud fija es seguido por un código de longitud fija de 5 bits, y para $N > 3$, un escape de longitud fija es seguido por el código del complemento del mosaico.

30 El mosaico rectangular contiene 6 bits de información. Supongamos que k sea el código asociado con el mosaico, donde $k = b_i 2^i$, b_i es el valor binario del i-ésimo bit en orden natural de exploración dentro del mosaico. Por lo tanto $0 \leq k < 64$. Se utiliza una combinación de VLC y códigos de escape más códigos de longitud fija para señalar k.

Modo Diff-6

35 El método Normal-6 se utiliza para producir el plano de bits como se describió anteriormente, y luego la operación Diff⁻¹ se aplica al plano de bits como se describe a continuación.

Modo salto de fila

40 En el modo de codificación de salto de fila todas las filas de cero se saltan con una sobrecarga de un bit. La sintaxis es la siguiente: para cada fila, un solo bit ROWSKIP indica si se omite la fila; si se omite la fila, el bit ROWSKIP para la siguiente fila es el siguiente; de lo contrario (la fila no se omite), los bits ROWBITS (un bit para cada macrobloque de la fila) son los siguientes. Por lo tanto, si toda la fila es cero, se envía un bit cero como símbolo de ROWSKIP y se omite ROWBITS. Si hay un bit configurado en la fila, ROWSKIP se configura en 1 y toda la fila se

envía en bruto (ROWBITS). Las filas se exploran desde la parte superior hasta la parte inferior del campo o fotograma.

Modo de salto de columna

5 Salto de columna es la transposición de salto de fila . Las columnas se exploran desde la parte izquierda hasta la parte derecha del campo o fotograma.

Diff-1:: Decodificación diferencial inversa

10 Si se utiliza cualquiera de los modos diferenciales (Diff-2 o Diff-6), primero se decodifica un plano de bits de "bits diferenciales" utilizando los modos normales correspondientes (Norm-2 o Norm-6, respectivamente). Los bits diferenciales se utilizan para regenerar el plano de bits original. El proceso de regeneración es un 2-D DPCM en un alfabeto binario. Para regenerar el bit en la ubicación (i, j), el predictor $b_p(i, j)$ se genera de la siguiente manera (a partir de los bits $b(i, j)$ en las posiciones (i, j)):

$$b_p(i, j) = \begin{cases} A & i = j = 0, \text{ or } b(i, j - 1) \neq b(i - 1, j) \\ b(0, j - 1) & i == 0 \\ b(i - 1, j) & \text{de otra manera} \end{cases}$$

15 Para el modo de codificación diferencial, no se realiza el proceso de inversión por bits basado en INVERT. Sin embargo, el indicador INVERT se utiliza de forma diferente para indicar el valor del símbolo A para la derivación del predictor mostrado anteriormente. Más específicamente, A es igual a 0 si INVERT es igual a 0 y A es igual a 1 si INVERT es igual a 1. El valor real del plano de bits se obtiene mediante una operación XOR del predictor con el valor del bit diferencial decodificado. En la ecuación anterior, $b(i, j)$ es el bit en la posición ij-ésima después de la decodificación final (es decir, después de hacer Norm-2/Norm-6, seguido del xor diferencial con su predictor).

Ejemplo de filtro de superposición

20 En un ejemplo de modo de filtro de superposición utilizado en la segunda implementación de la transformación solapada condicional, el suavizado de superposición se lleva a cabo en la reconstrucción de 16 bits sin fijar en el decodificador cuando se señala. Esto es necesario porque el proceso de avance asociado con el suavizado de la superposición puede dar como resultado una expansión del intervalo más allá del intervalo permisible de 9 bits para los valores de píxeles. El resultado del suavizado de superposición se fija a 9 bits, en línea con el resto de los píxeles no tocados por el suavizado de superposición.

25 Los bordes verticales (píxeles a0, a1, b1, b0 en el ejemplo anterior) se filtran primero, seguidos de los bordes horizontales (píxeles p0, p1, q1, q0). El resultado intermedio después de la primera etapa de filtrado (suavizado de bordes vertical) se almacena en 16 bits. A continuación se muestran los filtros centrales aplicados a los cuatro píxeles que se encuentran en ambos bordes:

$$30 \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \left(\begin{pmatrix} 7 & 0 & 0 & 1 \\ -1 & 7 & 1 & 1 \\ 1 & 1 & 7 & -1 \\ 1 & 0 & 0 & 7 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} r_0 \\ r_1 \\ r_0 \\ r_1 \end{pmatrix} \right) \gg 3$$

Los píxeles originales que se filtran son (x0, x1, x2, x3). r0 y r1 son parámetros de redondeo, que toman valores alternos de 3 y 4 para garantizar un redondeo sin sesgos estadísticos. Los valores originales son filtrados por la matriz con entradas que son claramente fáciles de implementar. Estos valores, después de añadir los factores de redondeo, se desplazan tres bits para dar la salida filtrada (y0, y1, y2, y3).

35 Tanto para los filtros de borde horizontal como para los filtros de borde vertical, los valores de redondeo son r0 = 4, r1 = 3 para las columnas y filas con índices impares respectivamente, suponiendo que la numeración dentro de un bloque comience en 1. Para las columnas / filas con índice par, r0 = 3 y r1 = 4. El filtrado se define como una operación in situ de 16 bits - por lo tanto, los píxeles originales se sobrescriben después del suavizado. Para el filtrado de bordes verticales, los píxeles (a0, a1, b1, b0) corresponden a (x0, x1, x2, x3), que a su vez se filtran a (y0, y1, y2, y3). Del mismo modo, para el filtrado de bordes horizontales, la correspondencia es con (p0, p1, q1, q0) respectivamente.

45 Los píxeles en la esquina 2x2 de un bloque se filtran en ambas direcciones. El orden de filtrado determina sus valores finales, y por lo tanto es importante mantener el orden - filtrado de bordes verticales seguido de filtrado de bordes horizontales - para una mayor precisión de bits. Conceptualmente, la sujeción se debe realizar después de las dos etapas de filtrado direccional, en todos los píxeles que se filtran. Sin embargo, puede haber alguna ventaja

computacional al combinar la sujeción con el filtrado – esta es una cuestión de implementación siempre y cuando se haga con cuidado para generar la salida correcta.

3. Entorno informático

Las implementaciones descritas anteriormente de la transformación solapada condicional se pueden realizar en cualquiera de los diversos dispositivos en los que se realiza el procesamiento de señales de imagen y vídeo, incluyendo, entre otros ejemplos, ordenadores; equipos de grabación, transmisión y recepción de imágenes y vídeo; reproductores de vídeo portátiles; videoconferencias; aplicaciones de transmisión de vídeo en la web, etc. Las técnicas de codificación de imagen y vídeo se pueden implementar en circuitos de hardware (por ejemplo, en circuitos de un ASIC, FPGA, etc.), así como en software de procesamiento de imagen y vídeo que se ejecute en un ordenador u otro entorno informático (ya sea que se ejecute en la unidad central de procesamiento (CPU), o en un procesador gráfico dedicado, tarjeta de vídeo o similar), tal como se muestra en la Figura 10.

La Figura 10 ilustra un ejemplo generalizado de un entorno (1000) informático adecuado en el que se puede implementar la transformación solapada condicional descrita. El entorno (1000) informático no pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad de la invención, ya que la presente invención puede implementarse en diversos entornos informáticos de propósito general o especial.

Con referencia a la Figura 10, el entorno (1000) informático incluye al menos una unidad (1010) de procesamiento y memoria (1020). En la Figura 10, esta configuración (1030) más básica se encuentra incluida dentro de una línea discontinua. La unidad (1010) de procesamiento ejecuta instrucciones ejecutables por ordenador y puede tratarse de un procesador real o virtual. En un sistema multiproceso, varias unidades de procesamiento ejecutan instrucciones ejecutables por ordenador para aumentar la potencia de procesamiento. La memoria (1020) puede ser volátil (por ejemplo, registros, caché, RAM), no volátil (por ejemplo, ROM, EEPROM, memoria flash, etc.) o una combinación de ambas. La memoria (1020) almacena software (1080) que implementa la transformación solapada condicional descrita.

Un entorno informático puede tener características adicionales. Por ejemplo, el entorno (1000) informático incluye almacenamiento (1040), uno o más dispositivos (1050) de entrada, uno o más dispositivos (1060) de salida y una o más conexiones (1070) de comunicación. Un mecanismo de interconexión (no se muestra) como un bus, controlador o red interconecta los componentes del entorno (1000) informático. Normalmente, el software del sistema operativo (no se muestra) proporciona un entorno operativo para otro software que se ejecuta en el entorno (1000) informático y coordina las actividades de los componentes del entorno (1000) informático.

El almacenamiento (1040) puede ser extraíble o no extraíble e incluye discos magnéticos, cintas magnéticas o casetes, CD-ROM, CD-RW, DVD o cualquier otro medio que se pueda utilizar para almacenar información y al que se pueda acceder dentro del entorno (1000) informático. El almacenamiento (1040) almacena instrucciones para el software (1080) que implementa el codificador de audio que genera y comprime las matrices de cuantificación.

Los dispositivos (1050) de entrada pueden ser un dispositivo de entrada táctil como un teclado, ratón, lápiz o bola rastreadora, un dispositivo de entrada de voz, un dispositivo de escaneo u otro dispositivo que proporcione una entrada al entorno (1000) informático. Para el audio, los dispositivos (1050) de entrada pueden ser una tarjeta de sonido o dispositivo similar que acepte entrada de audio en forma analógica o digital, o un lector de CD-ROM que proporcione muestras de audio al entorno informático. Los dispositivos (1060) de salida pueden ser una pantalla, una impresora, un altavoz, una grabadora de CD u otro dispositivo que proporcione salida desde el entorno (1000) informático.

Las conexiones (1070) de comunicación permiten la comunicación a través de un medio de comunicación con otra entidad informática. El medio de comunicación transmite información como instrucciones ejecutables por computadora, información de audio o vídeo comprimida u otros datos en una señal de datos modulada. Una señal de datos modulada es una señal que tiene una o más de sus características configuradas o cambiadas de tal manera que codifican información en la señal. A modo de ejemplo, y no como limitación, los medios de comunicación incluyen técnicas con o inalámbricas implementadas con un portador eléctrico, óptico, de RF, infrarrojo, acústico o de otro tipo.

Las técnicas de transformación y codificación/decodificación de la presente memoria se pueden describir en el contexto general de los medios legibles por ordenador. Los medios legibles por ordenador son cualquier medio disponible al que se puede acceder dentro de un entorno informático. A modo de ejemplo, y no como limitación, con el entorno (1000) informático, entre los medios legibles por ordenador se incluyen la memoria (1020), el almacenamiento (1040), los medios de comunicación y combinaciones de cualquiera de los anteriores.

La transformación solapada condicional de la presente memoria se puede describir en el contexto general de las instrucciones ejecutables por ordenador, tales como las incluidas en los módulos de programa, que se ejecutan en el procesador real o virtual objetivo de un entorno informático. Generalmente, los módulos de programa incluyen rutinas, programas, bibliotecas, objetos, clases, componentes, estructuras de datos, etc. que realizan tareas específicas o implementan tipos de datos abstractos específicos. Las funcionalidades de los módulos de programa pueden combinarse o dividirse entre módulos de programa según se desee en varias realizaciones. Las

instrucciones ejecutables por ordenador para módulos de programa pueden ejecutarse dentro de un entorno informático local o distribuido.

5 Para facilitar la presentación, la descripción detallada utiliza términos como "determinar", "generar", "ajustar" y "aplicar" para describir las operaciones informáticas en un entorno informático. Estos términos son abstracciones de alto nivel para operaciones realizadas por un ordenador, y no deben confundirse con actos realizados por un ser humano. Las operaciones informáticas reales correspondientes a estos términos varían en función de la implementación.

10 En vista de las muchas realizaciones posibles a las que se pueden aplicar los principios de nuestra invención, reivindicamos como nuestra invención todas las realizaciones que pueden estar dentro del alcance de las siguientes reivindicaciones y las equivalentes a estas.

REIVINDICACIONES

1. Un método para codificar un flujo de bits de vídeo utilizando una transformación solapada condicional, en donde el método comprende:

5 la señalización de un modo de filtro de superposición del codificador al decodificador del flujo de bits de vídeo macrobloque, en donde la señalización del modo de superposición es un conjunto de indicadores (740) de modo de superposición por macrobloque;

la aplicación condicional de un filtro de superposición para una transformación solapada de dominio espacial, SDLT, a un subconjunto de bordes entre bloques de transformación en una base espacialmente variable en función de la condición de superposición, en donde la condición de superposición tiene al menos un estado sin superposición en el que no se aplica ningún filtro de superposición a los bordes de bloques adyacentes y un estado de superposición en el que el filtro de superposición se aplica a los bordes de bloques adyacentes, y en donde la condición de superposición varía espacialmente con la granularidad de los macrobloques, y

en donde la aplicación condicional del filtro de superposición comprende

15 solo si se señala que la condición de superposición de un intramacrobloque está en el estado de superposición, aplicar el filtro de superposición a todos los bordes internos del bloque del intramacrobloque; y

solo si se señala que la condición de superposición está en el estado de superposición para ambos intramacrobloques adyacentes, aplicar el filtro de superposición a los bordes de los bloques entre los dos intramacrobloques adyacentes; y de lo contrario no aplicar ningún filtro de superposición a los bordes o entre macrobloques.

2. El método de la reivindicación 1, en donde la aplicación del filtro de superposición depende además de un parámetro relacionado con la tasa de bits.

3. El método de la reivindicación 2, en donde el parámetro relacionado con la tasa de bits es un parámetro de cuantificación a nivel de fotograma.

25 4. Un método de decodificación de un flujo de bits de vídeo codificado utilizando una transformación solapada condicional mediante una transformación solapada de dominio espacial, SDLT, en donde el método comprende:

la decodificación de datos de bloques de una pluralidad de bloques codificados de un fotograma en el flujo de bits de vídeo codificado;

la realización de la descuantificación de los datos de bloques decodificados;

30 la realización de una transformación inversa de bloques de transformación de los datos de bloques decodificados;

la determinación, después de la transformación inversa, de una condición de superposición para grupos de más de un borde de bloques de transformación adyacentes de al menos un elemento sintáctico del flujo de bits de vídeo codificado, en donde la condición de superposición tiene al menos un estado sin superposición en el que no se aplica ningún filtro de superposición a los bordes de bloques adyacentes y un estado de superposición en el que el filtro de superposición se aplica a los bordes de bloques adyacentes, y

en donde al menos un elemento sintáctico comprende un conjunto de indicadores de modo de superposición que tienen granularidad de macrobloques,

siendo señalada la condición de superposición en el flujo de bits de vídeo codificado mediante indicadores (740) de modo de superposición por macrobloque del conjunto de indicadores de modo de superposición; y

40 la aplicación condicional del filtro de superposición a un subconjunto de bordes entre bloques de transformación de manera espacialmente variable en función de la condición de superposición, incluyendo:

solo si se señala que la condición de superposición de un intramacrobloques está en el estado de superposición, la aplicación del filtro de superposición a todos los bordes de bloque internos del intramacrobloque;

45 solo si se señala que la condición de superposición está en el estado de superposición para ambos intramacrobloques adyacentes, la aplicación del filtro de superposición a los bordes de los bloques entre los dos intramacrobloques adyacentes; y

de lo contrario la no aplicación de ningún filtro de superposición a los bordes o entre macrobloques.

5. El método de la reivindicación 4, en donde el conjunto de indicadores de modo de superposición se codifica usando codificación de plano de bits.

6. El método de la reivindicación 4 o la reivindicación 5, en donde al menos un elemento sintáctico comprende además un parámetro (730) de superposición a nivel de fotograma y un parámetro (QP) relacionado con la tasa de bits basado en fotograma, en donde el método comprende además:

5 si el parámetro relacionado con la tasa de bits indica una tasa (820) de bits baja, la determinación de aplicar el filtro de superposición a todos los bordes de los bloques de transformación adyacentes dentro del fotograma respectivo;

de lo contrario si el parámetro de superposición a nivel de fotograma no especifica superposición (850), la determinación de no aplicar ningún filtro de superposición a los bordes de los bloques de transformación adyacentes dentro del fotograma respectivo;

10 de lo contrario, si el parámetro de superposición a nivel de fotograma especifica superposición (860) condicional, la determinación de aplicar el filtro de superposición según lo especificado por el conjunto de indicadores de modo de superposición; y

de lo contrario (870), la determinación de aplicar el filtro de superposición a todos los bordes de los bloques de transformación adyacentes dentro del fotograma respectivo.

15 7. El método de la reivindicación 4 o la reivindicación 5, en donde al menos un elemento sintáctico comprende además un parámetro de superposición a nivel de secuencia, y el método comprende además:

si el parámetro de superposición a nivel de secuencia no especifica ninguna superposición, la determinación de no aplicar ningún filtro de superposición a los bordes de los bloques de transformación adyacentes dentro de todos los fotogramas de la secuencia.

20 8. El método de la reivindicación 4 o la reivindicación 5, en donde al menos un elemento sintáctico comprende además otro elemento sintáctico relativo a una tasa de bits del flujo de bits de vídeo codificado.

9. El método de la reivindicación 8, en donde el elemento sintáctico relacionado con una tasa de bits del flujo de bits de vídeo codificado es un parámetro de cuantificación a nivel de fotograma.

25 10. Al menos un programa legible por ordenador que lleva el medio que tiene un módulo de software en el mismo que se puede ejecutar mediante una unidad de procesamiento para realizar el método de una de las reivindicaciones 1 a 9.

Figura 1

Técnica anterior

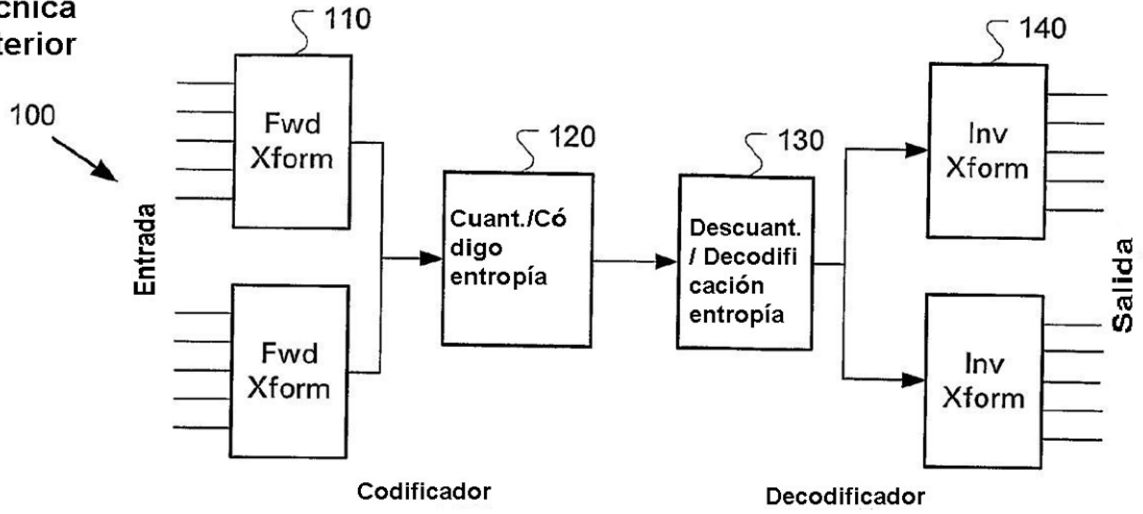


Figura 2

Técnica anterior

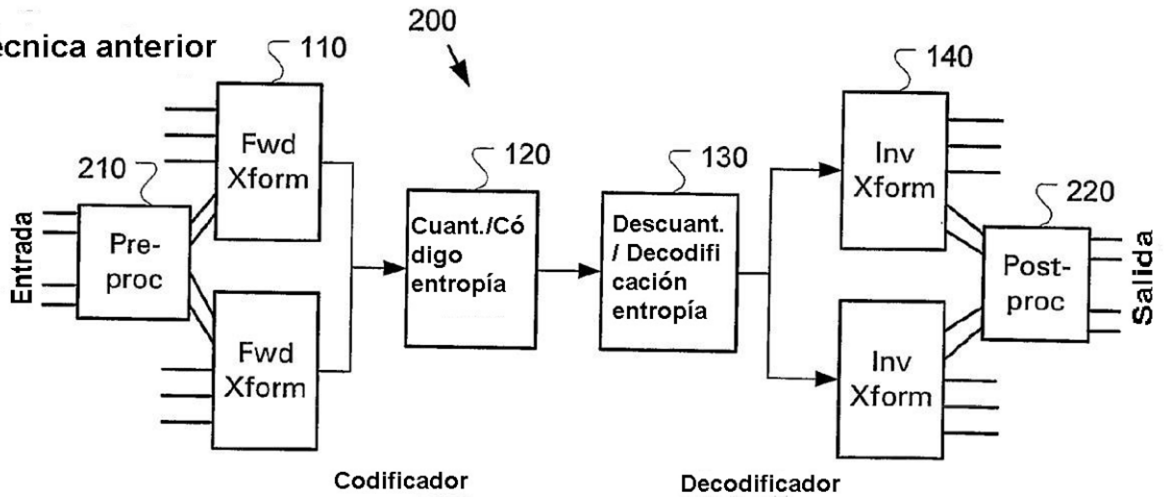


Figura 3

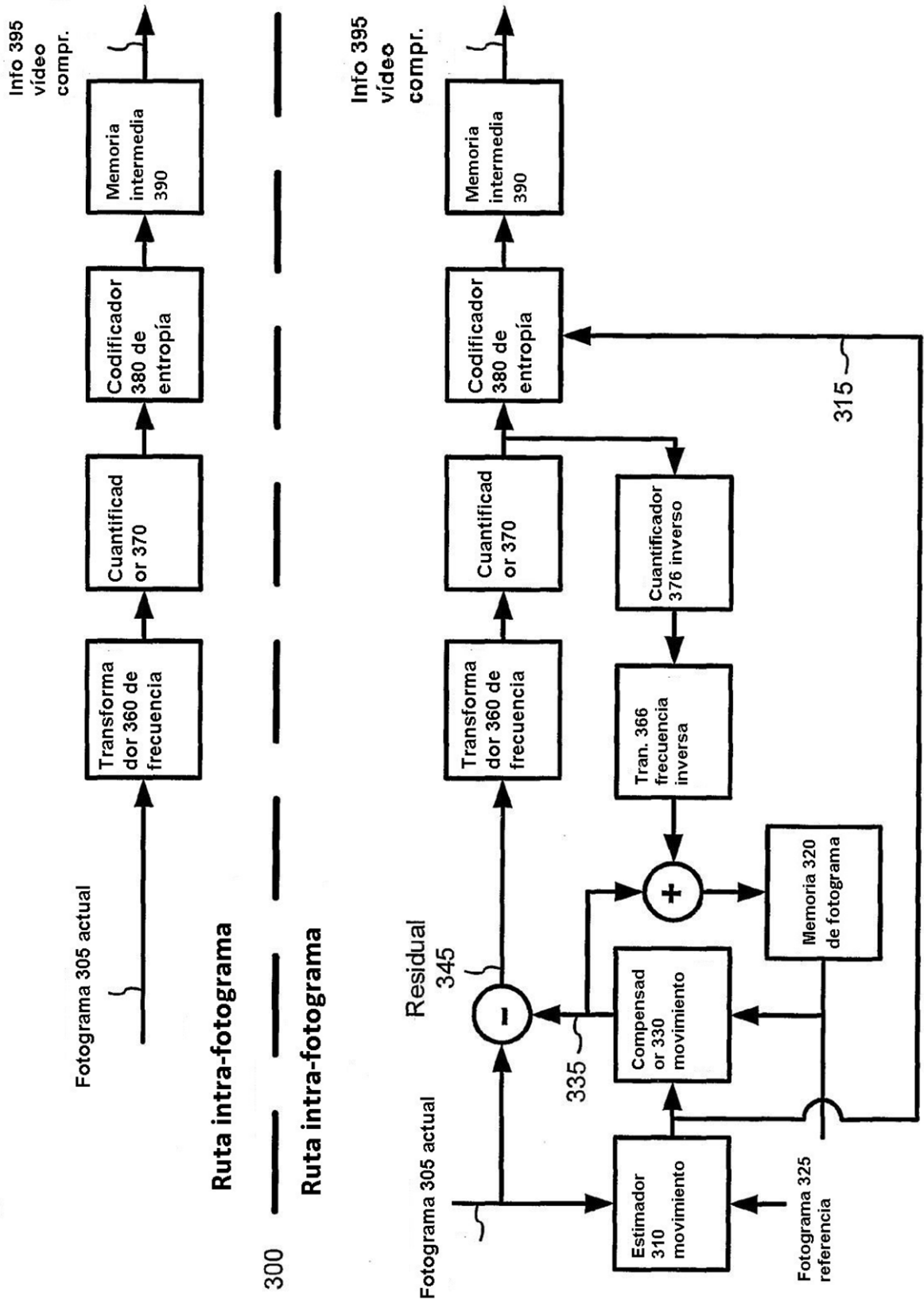


Figura 4

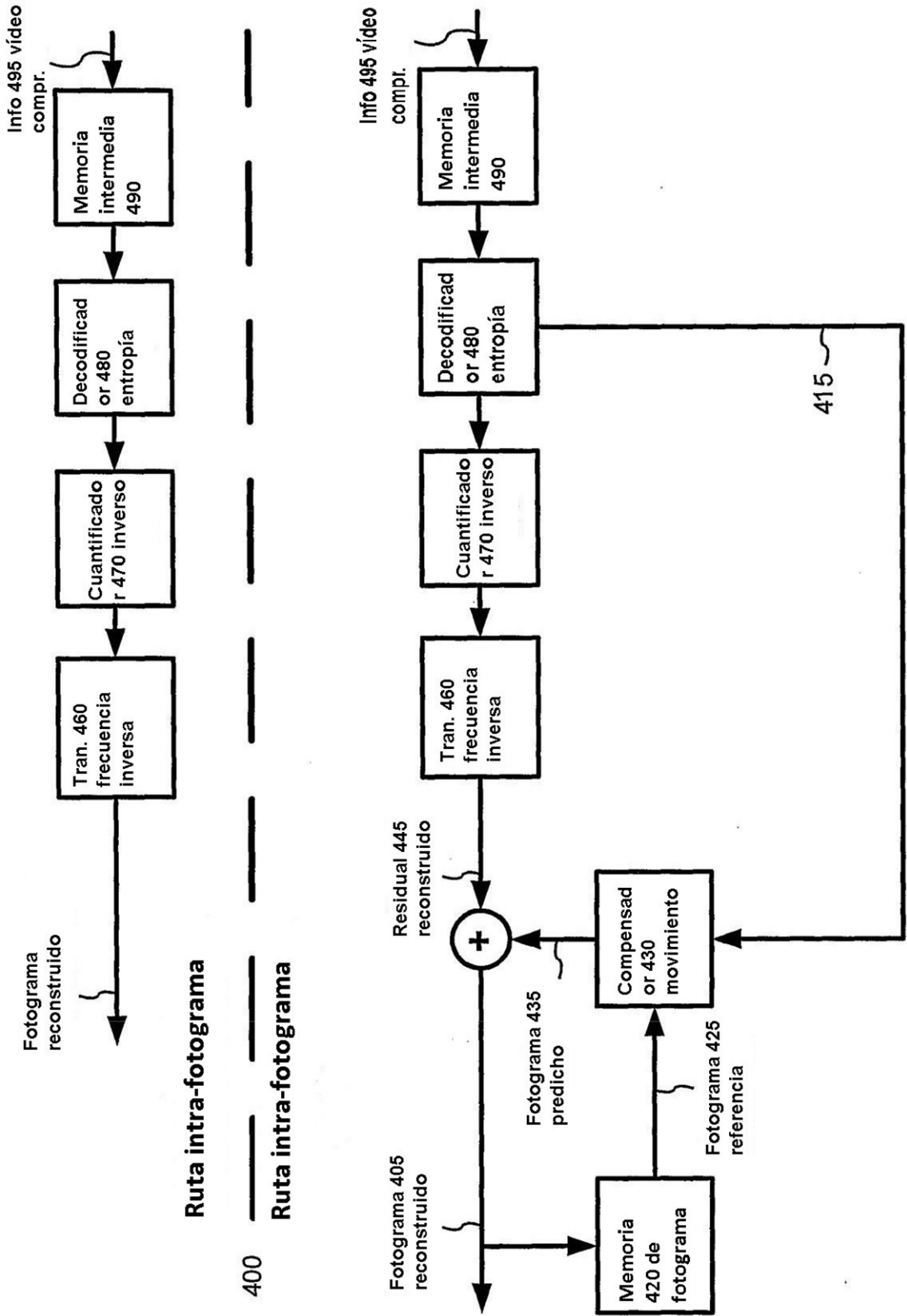


Figura 5

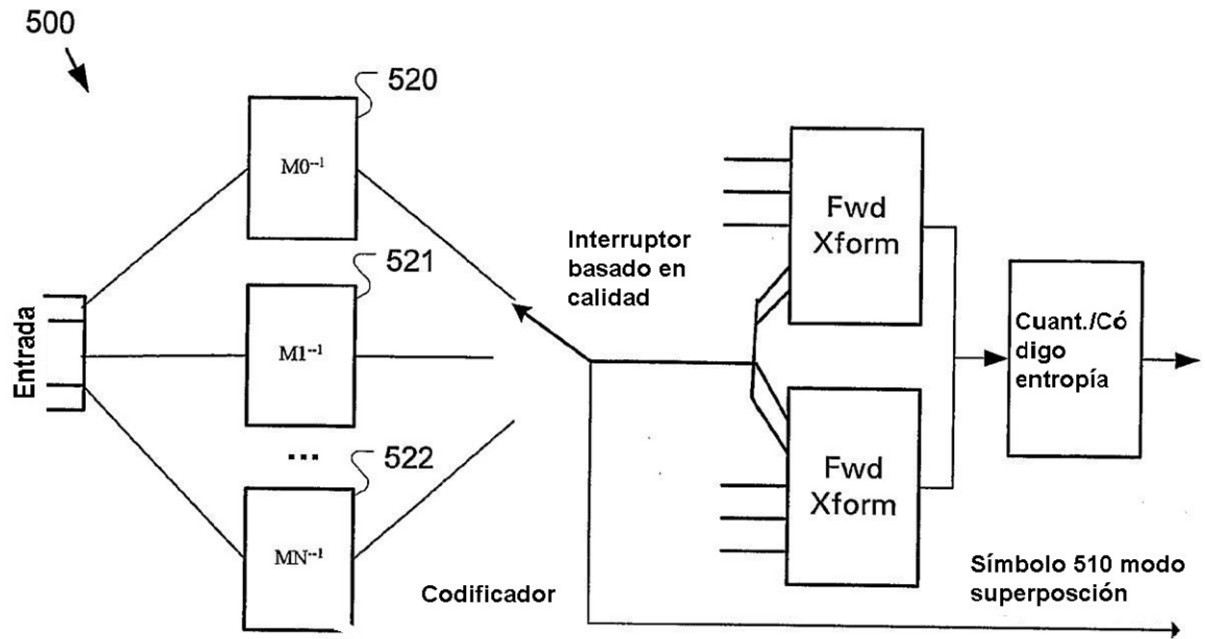


Figura 10

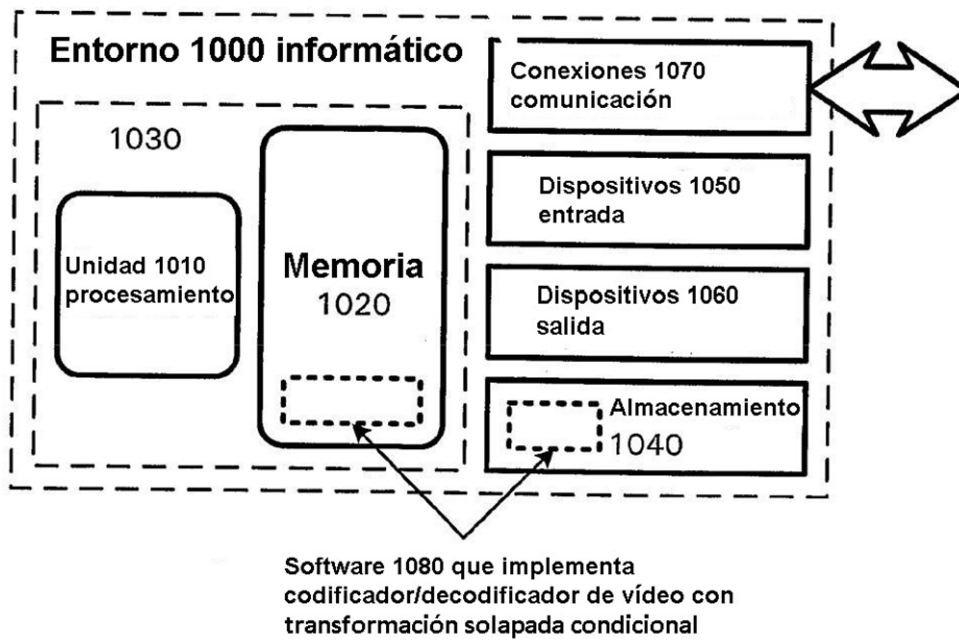
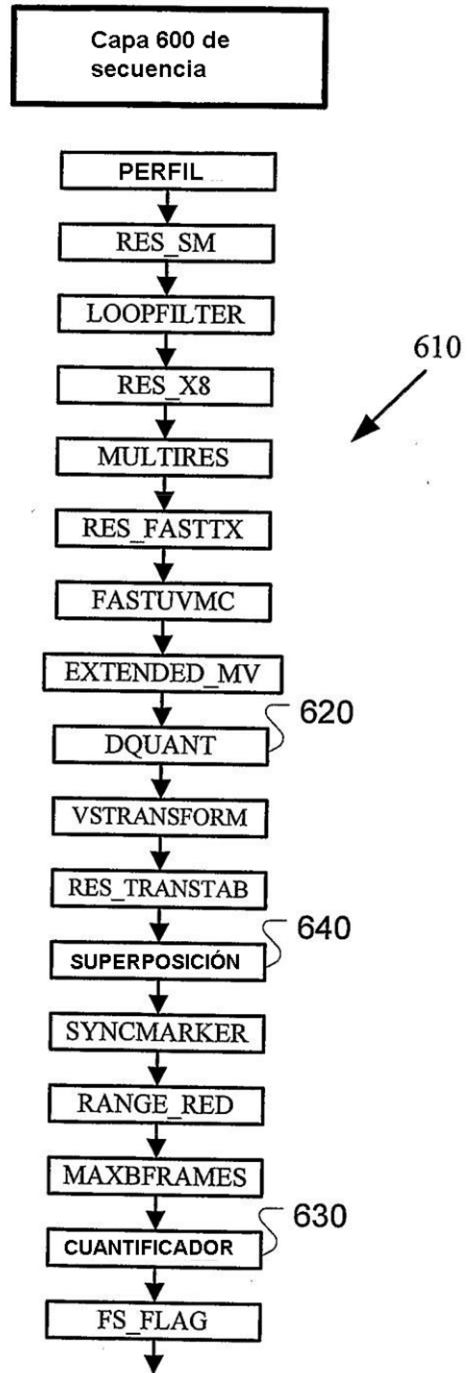


Figura 6



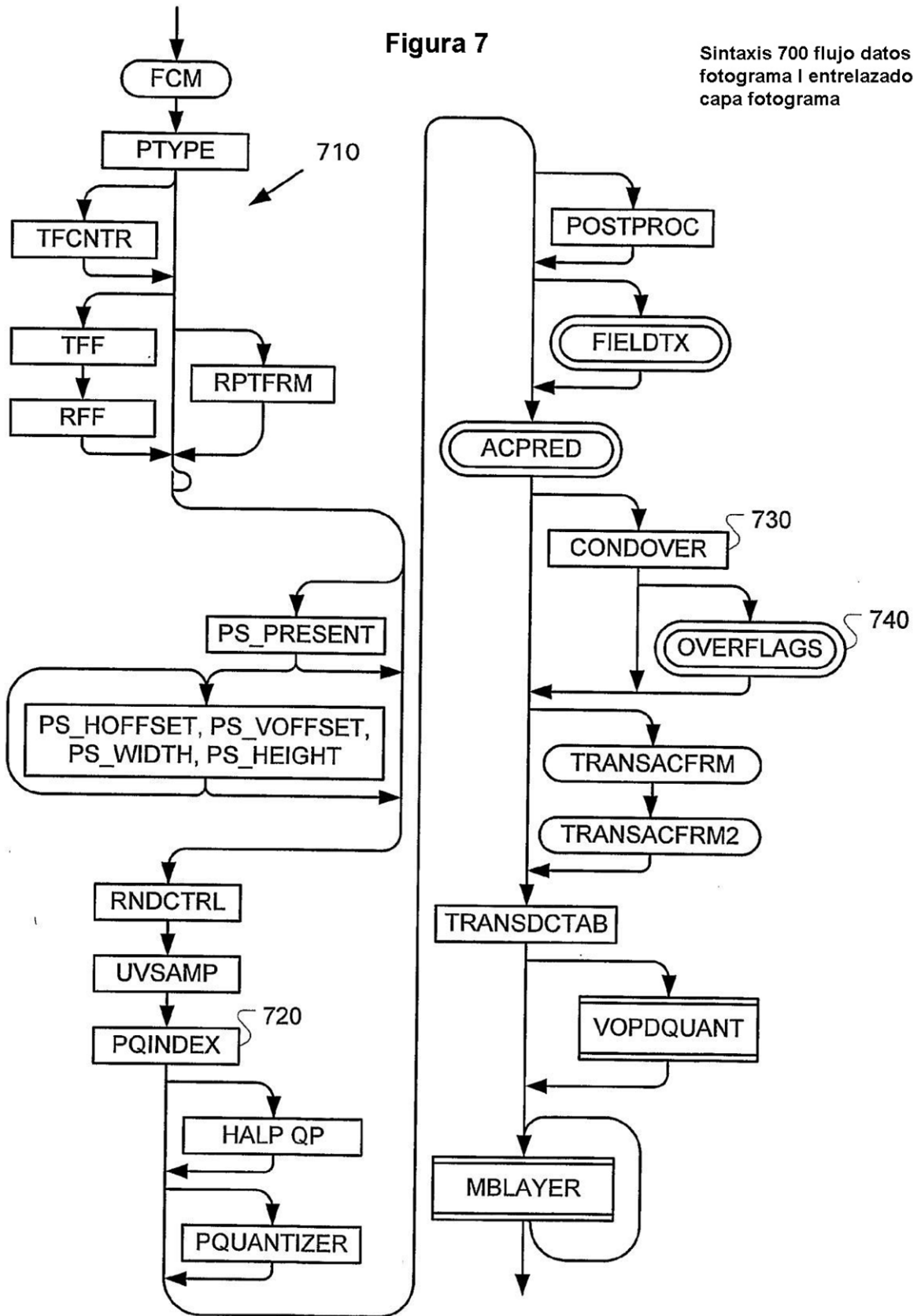


Figura 8

800

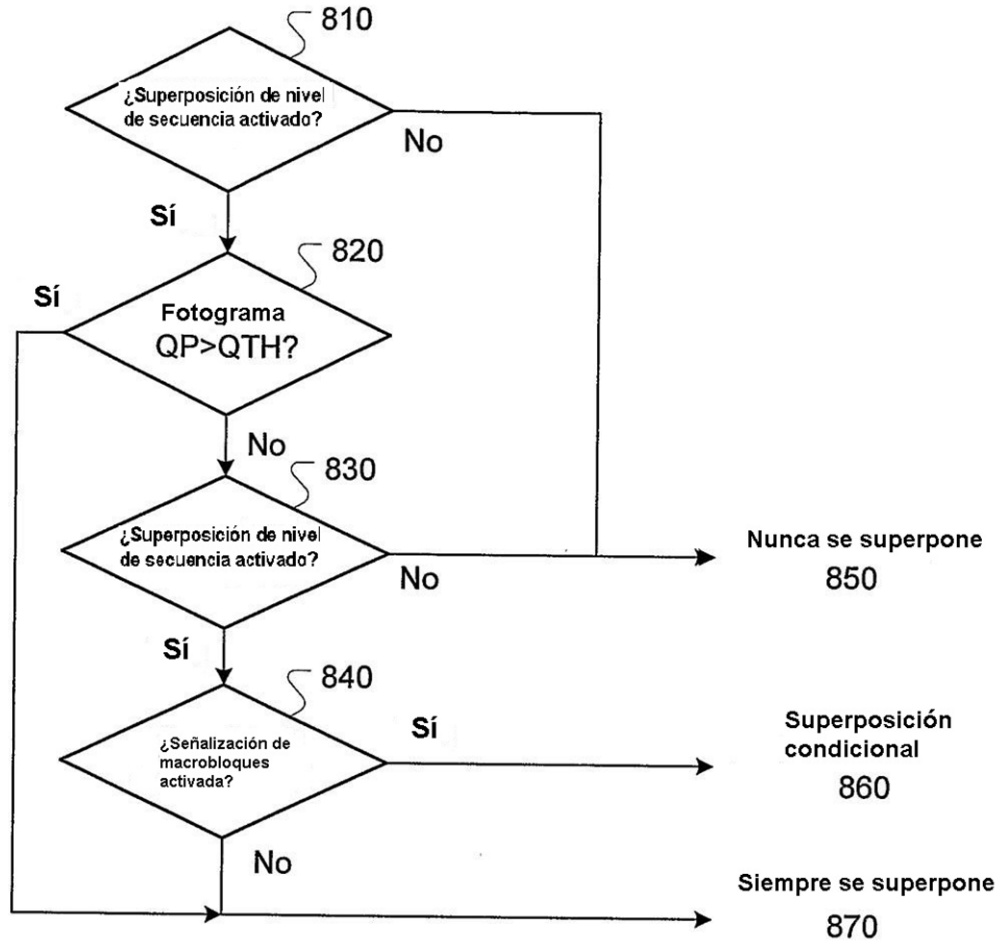


Figura 9

900

