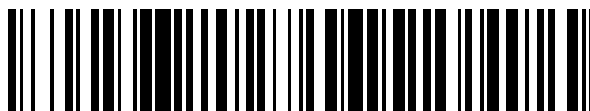


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 610 430**

51 Int. Cl.:

H04N 19/507 (2014.01)

H04N 19/61 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **16.12.2002 PCT/US2002/40208**

87 Fecha y número de publicación internacional: **26.06.2003 WO03053066**

96 Fecha de presentación y número de la solicitud europea: **16.12.2002 E 02787048 (4)**

97 Fecha y número de publicación de la concesión europea: **10.08.2016 EP 1457056**

54 Título: **Codificación por omisión de macrobloques**

30 Prioridad:

17.12.2001 US 341674 P
03.05.2002 US 377712 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
27.04.2017

73 Titular/es:

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

72 Inventor/es:

SRINIVASAN, SRIDHAR y
HSU, POHSIANG

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 610 430 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

Codificación por omisión de macrobloques

5 INFORMACIÓN DE SOLICITUDES RELACIONADAS

La presente solicitud reivindica el beneficio de la Solicitud de Patente Provisional N° de Serie 60/341.674 titulada "Techniques and Tools for Video Encoding and Decoding", ("Técnicas y Herramientas para Codificación y Descodificación de Vídeo"), presentada el 17 de Diciembre de 2001, cuya descripción es incorporada por referencia. La presente solicitud reivindica también el beneficio de la Solicitud de Patente Provisional N° de Serie 60/377.712 titulada "Skip Macroblock Coding" ("Codificación por Omisión de Macrobloques", presentada el 3 de Mayo de 2002, cuya descripción es incorporada por referencia.

CAMPO TÉCNICO

Hay descritas técnicas y herramientas para codificar/descodificar información binaria en aplicaciones de codificación/descodificación de vídeo. Por ejemplo, un codificador de vídeo codifica la información de macrobloques omitidos.

ANTECEDENTES.

El vídeo digital consume grandes cantidades de capacidad de almacenamiento y de transmisión. Una secuencia de vídeo digital en bruto típica incluye 15 o 30 fotogramas por segundo. Cada fotograma puede incluir decenas o cientos de millares de píxeles (también llamados pels). Cada pixel representa un elemento diminuto de la imagen. En forma en bruto, un ordenador representa comúnmente un pixel con 24 bits. Así, el número de bits por segundo, o tasa de bits, de una secuencia de vídeo digital en bruto típica puede ser de 5 millones de bits/segundo o más.

La mayoría de los ordenadores y redes informáticas carecen de recursos para procesar y vídeo digital en bruto. Por esta razón, los ingenieros utilizan la compresión (también llamado codificación o descodificación) para reducir la tasa de bits del vídeo digital. La compresión puede ser sin pérdidas, en la que la calidad del vídeo no sufre pero las disminuciones en tasa de bits están limitadas por la complejidad del vídeo. O, la compresión puede ser con pérdidas, en la que la calidad del vídeo sufre pero las disminuciones en tasa de bits son más dramáticas. La descompresión invierte la compresión.

En general, las técnicas de compresión de vídeo incluyen compresión dentro de fotogramas y compresión entre fotogramas. Las técnicas de compresión dentro de fotogramas comprimen fotogramas individuales, típicamente llamados I- fotogramas, o fotogramas clave. Las técnicas de compresión entre fotogramas comprimen fotogramas con referencia a fotogramas precedentes y/o siguientes, y son llamados típicamente fotogramas predichos, P-fotogramas o B-fotogramas.

El Windows Media Video de la Corporación Microsoft, Versión 7 ["WMV7"] incluye un codificador de vídeo y un descodificador de vídeo. El codificador WMV7 utiliza compresión dentro de fotogramas y entre fotogramas, y el descodificador WMV7 utiliza descompresión dentro de fotogramas y entre fotogramas.

40 A. Compresión dentro de fotogramas en WMV7

La Figura 1 ilustra la compresión (100) dentro de fotogramas a base de bloques de un bloque (105) de píxeles en un fotograma clave en el codificador WMV7. Un bloque es un conjunto de píxeles, por ejemplo, una disposición de píxeles de 8x8. El codificador WMV7 divide un fotograma de vídeo clave en bloques de píxeles de 8x8 y aplica una Transformación de Coseno Discreta ["DCT"] (110) de 8x8 a bloques individuales tales como el bloque (105). Una DCT es un tipo de transformación de frecuencia que convierte el bloque de píxeles de 8x8 (información espacial) en un bloque de coeficientes DCT (115) de 8x8, que son información de frecuencia. La propia operación DCT no tiene pérdidas o casi no tiene pérdidas. Comparado con los valores de píxel originales, sin embargo, los coeficientes de DCT son más eficientes para que el codificador comprima ya que la mayoría de la información significativa está concentrada en coeficientes de baja frecuencia (convencionalmente, la parte superior izquierda del bloque (115)) y muchos de los coeficientes de alta frecuencia (convencionalmente, la parte inferior derecha del bloque (115)) tienen valores de cero o próximos a cero.

El codificador cuantifica a continuación (120) los coeficientes DCT, dando como resultado un bloque de coeficientes DCT cuantificados (125) de 8x8. Por ejemplo, el codificador aplica un tamaño de paso de cuantificación escalar, uniforme a cada coeficiente, que es análogo a dividir cada coeficiente por el mismo valor y redondear. Por ejemplo, si un valor de coeficiente DCT es 163 y el tamaño de paso es 10, el valor de coeficiente DCT cuantificado es 16. La cuantificación es con pérdidas. El valor de coeficiente DCT reconstruido será 160, no 163. Como los coeficientes DCT de baja frecuencia tienden a tener valores más elevados, la cuantificación da como resultado pérdidas de precisión pero no pérdidas completas de información para los coeficientes. Por el otro lado, mientras los coeficientes DCT de alta frecuencia tienden a tener valores de cero o próximos a cero, la cuantificación de los coeficientes de alta frecuencia da como resultado típicamente en regiones contiguas de valores de cero. Además, en algunos casos los coeficientes DCT de alta frecuencia son cuantificados de forma más aproximada que los coeficientes DCT de baja frecuencia, dando como resultado mayores pérdidas de precisión/información para los coeficientes DCT de alta frecuencia.

El codificador prepara a continuación el bloque de coeficientes DCT cuantificados (125) de 8x8 para codificación de entropía, que es una forma de compresión sin pérdidas. El tipo exacto de codificación de entropía puede variar dependiendo de si un coeficiente es un coeficiente DC (frecuencia más baja), un coeficiente AC (otras frecuencias) en la

fila superior o columna izquierda, u otro coeficiente AC.

El codificador codifica el coeficiente DC (126) como un diferencial del coeficiente DC (136) de un bloque de 8x8 vecino, que es un vecino previamente codificado (por ejemplo parte superior o izquierda) del bloque que está siendo codificado. (La Figura 1 muestra un bloque vecino (135) que está situado a la izquierda del bloque que está siendo codificado en el fotograma). La entropía de codificador codifica (140) el diferencial.

El codificador de entropía puede codificar la columna izquierda o la fila superior de los coeficientes AC como un diferencial de una columna o fila correspondiente del bloque de 8x8 vecino. La Figura 1 muestra la columna izquierda (127) de coeficientes AC codificados como un diferencial (147) desde la columna izquierda (137) del bloque (135) vecino (a la izquierda). La codificación diferencial aumenta la posibilidad de que los coeficientes de diferencial tengan valores cero. Los coeficientes AC restantes son del bloque (125) de los coeficientes DCT cuantificados.

El codificador analiza (150) el bloque de 8x8 (145) de coeficientes DCT AC cuantificados, predichos en una agrupación unidimensional (155) y a continuación la entropía codifica los coeficientes AC analizados utilizando una variación de codificación de longitud de recorrido (160). El codificador selecciona un código de entropía de una o más tablas de recorrido/nivel/último (165) y emite el código de entropía.

Un fotograma clave contribuye mucho más a la tasa de bits que un fotograma predicho. En aplicaciones de tasa de bits media o baja, los fotogramas clave son a menudo cuellos de botella críticos para el rendimiento, así que la compresión eficiente de los fotogramas clave es crítica.

La Figura 2 ilustra una desventaja de compresión dentro de fotogramas tal como se ha mostrado en la Figura 1. En particular, la explotación de redundancia entre bloques del fotograma clave está limitada a la predicción de un subconjunto de coeficientes de frecuencia (por ejemplo, el coeficiente DC y la columna izquierda (o fila superior) de coeficientes AC) desde el bloque vecino de la izquierda (220) o de la parte superior (230) de un bloque (210). El coeficiente DC representa la media del bloque, la columna izquierda de coeficientes AC representa los promedios de las filas de un bloque, y la fila superior representa los promedios de las columnas. En efecto, la predicción de coeficientes DC y AC como en WMV7 limita la extrapolación a las señales promedio del modo fila (o de modo columna) del bloque vecino izquierdo (o superior). Para una fila particular (221) en el bloque izquierdo (220), los coeficientes AC en la columna de coeficiente DCT izquierda para el bloque izquierdo (220) son utilizados para predecir la fila correspondiente (211) completa del bloque (210).

B. Compresión Entre Fotogramas en WMV7

La compresión entre fotogramas en el codificador WMV7 utiliza la codificación de predicción compensada de movimiento basado en bloques seguida por codificación de transformación del error residual. Las Figuras 3 y 4 ilustran la compresión entre fotogramas basada en bloques para un fotograma predicho en el codificador WMV7. En particular, la Figura 3 ilustra la estimación de movimiento para un fotograma predicho (310) y la Figura 4 ilustra la compresión de una predicción residual para un bloque de movimiento estimado de un fotograma predicho.

El codificador WMV7 divide un fotograma predicho en bloques de píxeles de 8x8. Grupos de 4 bloques de 8x8 forman macrobloques. Para cada macrobloque, se realiza un proceso de estimación de movimiento. La estimación de movimiento aproxima el movimiento del macrobloque de píxeles con relación a un fotograma de referencia, por ejemplo, un fotograma precedente, previamente codificado. En la FIGURA 3, el codificador WMV7 calcula un vector de movimiento para un macrobloque (315) en el fotograma predicho (310). Para calcular el vector de movimiento, el codificador busca en un área de búsqueda (335) de un fotograma de referencia (330). Dentro del área de búsqueda (335), el codificador compara el macrobloque (315) procedente del fotograma predicho (310) con distintos macrobloques candidatos con el fin de encontrar que un macrobloque candidato es una buena coincidencia. El codificador puede comprobar macrobloques candidatos cada píxel o cada 1/2 píxel en el área de búsqueda (335), dependiendo de la resolución de estimación de movimiento deseada para el codificador. Otros codificadores de video comprueban en otros incrementos, por ejemplo, cada 1/4 de píxel. Para un macrobloque candidato, el codificador comprueba la diferencia entre el macrobloque (315) del fotograma predicho (310) y el macrobloque candidato y el coste de codificar el vector de movimiento para ese macrobloque. Después de que el codificador encuentra un macrobloque de buena coincidencia, el proceso de coincidencia de bloques termina. El codificador emite el vector de movimiento (codificado de entropía) para el macrobloque coincidente así el descodificador puede encontrar el macrobloque coincidente durante la descodificación. Cuando se descodifica el fotograma predicho (310), un descodificador utiliza el vector de movimiento para calcular un macrobloque predicho para el macrobloque (315) utilizando información procedente del fotograma de referencia (330). La predicción para el macrobloque (315) es raramente perfecta, así el codificador codifica normalmente bloques de 8x8 de diferencias de píxeles (también llamado los bloques de error o residuales) entre el macrobloque predicho y el propio macrobloque (315).

La Figura 4 ilustra el cálculo y codificación de un bloque de error (435) para un bloque de movimiento estimado en el codificador WMV7. El bloque de error (435) es la diferencia entre el bloque predicho (415) y el bloque actual original (425). El codificador aplica una DCT (440) al bloque de error (435), que da como resultado el bloque (445) de 8x8 de coeficientes. Aún más de lo que estaba en el caso con coeficientes DCT para valores de píxel, la información significativa para el bloque de error (435) está concentrada en coeficientes de baja frecuencia (convencionalmente, la parte superior

izquierda del bloque (445)) y muchos de los coeficientes de alta frecuencia tienen valores de cero o próximos a cero (convencionalmente, la parte inferior derecha del bloque (445)).

5 El codificador cuantifica a continuación (450) los coeficientes DCT, que dan como resultado un bloque de coeficientes DCT cuantificados (455) de 8x8. El tamaño del paso de cuantificación es ajustable. Otra vez, como los coeficientes DCT de baja frecuencia tienden a tener valores más elevados, la cuantificación da como resultado pérdida de precisión, pero no pérdida completa de información para los coeficientes. Por otro lado, como los coeficientes DCT de alta frecuencia tienden a tener valores de cero o próximos a cero, la cuantificación de los coeficientes de alta frecuencia da como resultado regiones contiguas de valores de cero. Además, en algunos casos los coeficientes DCT de alta frecuencia son
10 cuantificados de manera más aproximada que los coeficientes DCT de baja frecuencia, lo que da como resultado una mayor pérdida de precisión/información para los coeficientes DCT de alta frecuencia.

15 El codificador prepara a continuación el bloque (455) de coeficientes DCT de 8x8 cuantificados para codificación de entropía. El codificador analiza (460) el bloque (455) de 8x8 en una agrupación unidimensional (465) con 64 elementos, de tal manera que los coeficientes son generalmente ordenados desde la frecuencia más baja a la frecuencia más elevada, lo cual crea típicamente largos recorridos de valores cero.

20 La entropía del codificador codifica los coeficientes analizados utilizando una variación de codificación de longitud de recorrido (470). El codificador selecciona un código de entropía a partir de una o más tablas de ejecución/nivel/último (475) y emite el código de entropía.

25 Cuando el vector de movimiento de un macrobloque es cero (es decir, no hay movimiento) y no es transmitida la información de bloque residual para el macrobloque, el codificador utiliza un indicador de macrobloque omitido de 1 bit para el macrobloque. Para muchos tipos de contenido de video (por ejemplo, video de movimiento lento y/o de baja tasa de bits), esto reduce la tasa de bits evitando la transmisión del vector de movimiento y la información del bloque residual. El codificador pone el indicador de omisión de macrobloque para un macrobloque en la capa de macrobloques en la corriente de bits de salida, junto con otra información para el macrobloque.

30 La Figura 5 muestra el proceso de descodificación (500) para un bloque inter-codificado. Debido a la cuantificación de los coeficientes DCT, el bloque reconstruido (575) no es idéntico al bloque original correspondiente. La compresión es con pérdidas.

35 En resumen de la Figura 5, un descodificador descodifica (510, 520) la información codificada de entropía que representa una predicción residual utilizando descodificación de longitud variable y una o más tablas de ejecución/nivel/último (515). El descodificador inverso escanea (530) una agrupación unidimensional (525) que almacena la información descodificada de entropía en un bloque bidimensional (535). El descodificador inverso cuantifica y transforma el coseno discreto inverso (conjuntamente, 540) los datos, resultantes en un bloque de error reconstruido (545). En un trayecto separado, el descodificador calcula un bloque predicho (565) utilizando información del vector de movimiento (555) para el desplazamiento a partir de un fotograma de referencia. El descodificador combina (570) el bloque predicho (555) con el
40 bloque de error reconstruido (545) para formar el bloque reconstruido (575).

45 Cuando el descodificador recibe un indicador de omisión de macrobloque para un macrobloque, el descodificador omite calcular una predicción y descodificación de una información de bloque residual para el macrobloque. En su lugar, el descodificador utiliza datos de píxel correspondientes procedentes de la ubicación del macrobloque en el fotograma de referencia.

50 La cantidad de cambio entre el fotograma original y reconstruido se denomina la distorsión y el número de bits requeridos para codificar el fotograma se denomina la tasa. La cantidad de distorsión es aproximadamente inversamente proporcional a la tasa. En otras palabras, codificar un fotograma con pocos bits (mayor compresión) dará como resultado una mayor distorsión y viceversa. Uno de los objetivos de un esquema de compresión de video es intentar mejorar la distorsión de tasa - en otras palabras intentar conseguir la misma distorsión utilizando menor número de bits (o los mismos bits y menor distorsión).

55 Aunque el uso de los indicadores de omisión de macrobloques en WMV7 reduce típicamente la tasa de bits para muchos tipos de contenido de video, es menos que óptimo en algunas circunstancias. En muchos casos, se produce un error al explotar la redundancia de indicadores de omisión de macrobloques de macrobloque a macrobloque, por ejemplo, cuando los macrobloques omitidos ocurren en racimos en una imagen. También, WMV7 ignora la predicción de movimiento para macrobloques en fotogramas predichos cuando los macrobloques son omitidos, lo que perjudica la eficiencia de compresión de los fotogramas predichos en algunos casos.

60 C. Normas para Compresión y Descompresión de Video
Aparte de WMV7, varias normas internacionales se refieren a compresión y descompresión de video. Estas normas incluyen las normas 1, 2, y 4 del Grupo de Expertos de Imagen en Movimiento ["MPEG"] y las normas H-261, H.262 y H.263 de la Unión de Telecomunicación Internacional ["ITU"]. Como WMV7, estas normas utilizan una combinación de
65 compresión dentro de fotogramas y entre fotogramas, aunque las normas difieren típicamente del WMV7 en los detalles de las técnicas de compresión utilizadas.

Algunas normas internacionales reconocen la codificación de omisión de macrobloques como una herramienta para ser utilizada en compresión y descompresión de video. Para más detalles acerca de la codificación por omisión de macrobloques en las normas, véanse las propias especificaciones de las normas.

5 La codificación de macrobloques omitidos en las normas anteriores reduce típicamente la tasa de bits para muchos tipos de contenidos de video, pero es menos óptimo en algunas circunstancias. En muchos casos, no logra reducir la redundancia en indicadores por omisión de macrobloques de macrobloque a macrobloque, por ejemplo, cuando los macrobloques omitidos ocurren en racimos en una imagen. También, ignora la predicción de movimiento para
10 macrobloques en macrobloques/imágenes predichos cuando los macrobloques son omitidos, lo que perjudica la eficiencia de compresión de macrobloques/imágenes predichos en algunos casos.

15 Dada la importancia crítica de compresión y descompresión a video digital, no es sorprendente que la compresión y descompresión de video sean campos ampliamente desarrollados. Cualquiera que sea el beneficio de las técnicas de compresión y descompresión de video previas, sin embargo, no tienen las ventajas de las siguientes técnicas y herramientas.

El documento US 5.552.832 se refiere al tratamiento de la señal de video para generar señales de video comprimidas que pueden ser transmitidas bajo las restricciones de ancho de banda de las conexiones de red de telefonía conmutada pública y a continuación descomprimidos en tiempo real por un sistema de PC para proporcionar una conferencia de audio y video satisfactoria. Las señales de imagen de video son codificadas basadas en bloques de píxeles de 8x8 donde cada bloque puede ser codificado como bloques, omitidos intra-bloques o entre bloques. Un bloque es omitido si la suma de los valores absolutos de las diferencias de coeficiente DCT cuantificado para un bloque dado es menor que un umbral de banda neutra especificado. Si la composición del bloque está cambiando como resultado de una fuente de información, como el movimiento, a continuación la suma excederá el umbral de banda neutra dentro de uno o dos intervalos de tiempo. Si un bloque es omitido, el descodificador sabrá a continuación copiar el bloque correspondiente (8x8) de coeficientes DCT desde el fotograma descodificado previo. Además, un procesador anfitrión mantiene una memoria tampón del fotograma de coeficiente DCT único. Para un bloque omitido, el bloque correspondiente de la memoria tampón del fotograma es dejado sin cambios desde su estado para el fotograma previo. Dentro de una sintaxis de corriente de bits, un encabezamiento de fotograma incluye un código de tipo fotograma (FTC) que indica si el fotograma es un fotograma mezclado (con intra-bloques, entre bloques y bloques omitidos), un fotograma predicho (con bloques solamente entre bloques y omitidos) o un fotograma de referencia (con intra-bloques solamente). Para fotogramas mezclados, un cuerpo de fotograma consiste de bits que definen el tipo de bloque (es decir, intra-bloque, entre bloques, o bloque omitido) y los coeficientes DCT cuantificados (o diferencias de coeficiente DCT cuantificado) para cada bloque (8x8) en el fotograma actual. Además, los tipos de bloques son codificados en longitud de recorrido mediante escaneado siguiendo un diseño de escaneado de bloque específico. Un diseño escaneado puede ser codificado en el codificador y descodificador como tablas de búsqueda, o el codificador y descodificador pueden tener una variedad de diseños de escaneado predefinidos para cada tamaño de fotograma.

40 Hsu P. y col., "A low bit-rate video codec based on two-dimensional mesh motion compensation with adaptive interpolation" ("Un codec de video de tasa de bits bajo en compensación de movimiento de malla bidimensional con interpolación adaptativa"), Transacciones de IEEE en circuitos y sistemas para tecnología de video, Volumen 11, nº 1, 2001 págs. 111 -117 se refiere a codificación de video basada en estimación y compensación de movimiento de malla tridimensional. En particular, para cada fotograma que ha de ser codificado, el fotograma y su fotograma previo es cubierto con una malla cuadrilateral regular. Los nodos de la malla están separados 16 píxeles en ambas direcciones. El desplazamiento de los nodos de un fotograma y su fotograma previo, es decir, el vector del movimiento del nodo, será utilizado para generar un campo de movimiento completo parche por parche a través de interpolación bilineal. Además de la interpolación bilineal por defecto, 32 nuevos diseños de interpolación son introducidos, en que los diseños de interpolación son clasificados en dos tipos de configuraciones. En la primera configuración, el diseño de interpolación consiste de dos regiones donde cada región contiene dos vectores de movimiento del nodo, por lo cual la segunda configuración describe una región que contiene tres vectores de movimiento del nodo y una región que contiene solamente un vector de movimiento de nodo. Además, una matriz de clasificación sirve para el propósito de indicar las ubicaciones donde han sido utilizados nuevos diseños de interpolación y una cadena de índices para notificar al descodificador de los diseños de interpolación real introducidos por el codificador. Para obtener un orden de visita para los parches interpolados adaptativos, es empleado un escaneado de trama fija de la matriz de clasificación en el codificador y descodificador.

COMPENDIO

60 Es el objeto del invento tener una mejor compresión de imágenes de video.

Este objeto es resuelto por el invento como se ha reivindicado en las reivindicaciones independientes.

Las realizaciones preferidas son definidas por las reivindicaciones dependientes.

65 En resumen, la descripción detallada está dirigida a distintas técnicas y herramientas para codificar y descodificar (por ejemplo, en un codificador/descodificador de video) información binaria. La información binaria puede comprender bits

que indican si un codificador o decodificador de video omite ciertos macrobloques en un fotograma de video. O, la información binaria puede comprender bits que indican la resolución del vector de movimiento para macrobloques (por ejemplo 1-MV o 4-MV), modo entrelazado (por ejemplo, campo o fotograma), o alguna otra información. La información binaria puede ser codificada sobre una base de fotograma por fotograma o en algunas otras bases.

5 En algunas realizaciones, la información binaria está dispuesta en un plano de bits. Por ejemplo, el plano de bits es codificado en la capa de imagen/fotograma. Alternativamente, la información binaria está dispuesta de alguna otra manera y/o codificada en una capa diferente. El codificador y decodificador procesan la información binaria. La información binaria puede comprender la información de nivel de macrobloques. Alternativamente, el codificador y
10 decodificador procesan los planos de bits del nivel de bloque, nivel de sub-bloque, o información de nivel de pixel.

En algunas realizaciones, el codificador y decodificador conmutan modos de codificación. Por ejemplo, el codificador y decodificador usan modo normal, de omisión de fila, o de omisión de columna. Los diferentes modos permiten al codificador y decodificador explotar la redundancia en la información binaria. Alternativamente, el codificador y decodificador utilizan otros modos y/o modos adicionales tales como modos diferenciales. Para aumentar la eficiencia, el codificador y decodificador puede utilizar una técnica de inversión de plano de bits en algunos modos.

En algunas realizaciones, el codificador y decodificador definen un macrobloque omitido como un macrobloque predicho cuyo movimiento es igual a su movimiento causalmente predicho y que tiene error residual cero. Alternativamente, el codificador y decodificador definen un macrobloque omitido como un macrobloque predicho con movimiento cero y error residual cero.

En algunas realizaciones, el codificador y decodificador utilizan un modo de codificación en bruto para permitir aplicaciones de latencia baja. Por ejemplo, en el modo de codificación en bruto, los macrobloques codificados pueden ser transmitidos al decodificador inmediatamente, sin tener que esperar a que todos los macrobloques en el fotograma/imagen sean codificados. El codificador y decodificador pueden conmutar entre el modo de codificación en bruto y otros modos.

Las distintas técnicas y herramientas pueden ser utilizadas en combinación o independientemente. En particular, la solicitud describe dos implementaciones de codificación y decodificación de macrobloques omitidos, junto con sintaxis de corriente de bits correspondiente. Diferentes realizaciones implementan una o más de las técnicas y herramientas descritas.

Características y ventajas adicionales serán puestas en evidencia a partir de la siguiente descripción detallada de diferentes realizaciones que prosigue con referencia a los dibujos adjuntos.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La Figura 1 es un diagrama que muestra compresión dentro de fotogramas basado en bloques de un bloque de píxeles de 8x8 de acuerdo con la técnica anterior.

La Figura 2 es un diagrama que muestra la predicción de coeficientes de frecuencia de acuerdo con la técnica anterior.

La Figura 3 es un diagrama que muestra la estimación de movimiento en un codificador de video de acuerdo con la técnica anterior.

La Figura 4 es un diagrama que muestra compresión entre fotogramas basado en bloques para un bloque de 8x8 de predicciones residuales en un codificador de video de acuerdo con la técnica anterior.

La Figura 5 es un diagrama que muestra descompresión dentro de fotogramas basado en bloques para un bloque de 8x8 de predicciones residuales de acuerdo con la técnica anterior.

La Figura 6 es un diagrama de bloques de un entorno de computación adecuado en el cual pueden ser implementadas varias realizaciones descritas.

La Figura 7 es un diagrama de bloques de un sistema codificador de video generalizado utilizados en varias realizaciones descritas.

La Figura 8 es un diagrama de bloques de un sistema decodificador de video generalizado utilizado en varias realizaciones descritas.

La Figura 9 es un gráfico que muestra los elementos de corriente de bits que componen la capa de imagen P de acuerdo con la primera implementación.

La Figura 10 es un diagrama de flujo que muestra una técnica para codificar información de macrobloque omitido en un codificador de video que tiene varios modos de codificación por omisión de macrobloques.

La Figura 11 es un diagrama de flujo que muestra una técnica para decodificar información de macrobloque omitido codificado por un codificador de video que tiene varios modos de codificación por omisión de macrobloque.

La Figura 12 muestra un ejemplo de un fotograma de codificación de macrobloque omitido.

La Figura 13 es un diagrama de flujo que muestra una técnica para codificar en modo de codificación normal por omisión de macrobloque.

La Figura 14 es un diagrama de flujo que muestra una técnica para codificar en un modo de codificación por omisión de macrobloque de predicción de fila.

La Figura 15 es un listado de código que muestra pseudo-código para la decodificación de predicción de fila de

información de macrobloque omitido.

La Figura 16 es un diagrama de flujo que muestra una técnica para codificar en un modo de codificación por omisión de macrobloque de predicción de columna.

La Figura 17 es un listado de código que muestra pseudo-código para descodificación de predicción de columna de información de macrobloque omitido.

La Figura 18 es un diagrama de flujo que muestra una técnica para determinar si omitir la codificación de ciertos macrobloques en un codificador de video.

La Figura 19 es un diagrama de flujo que muestra una técnica para codificar información binaria en un plano de bits en un modo de codificación de omisión de fila.

La Figura 20 es un diagrama de flujo que muestra una técnica para codificar información binaria en un plano de bits en un modo de codificación de omisión de columna.

La Figura 21 es un diagrama de flujo que muestra una técnica para codificar información binaria en un plano de bits en un modo de codificación normal 2.

Las Figuras 22, 23 y 24 muestran ejemplos de fotogramas de información binaria en mosaico en modo normal 6.

La Figura 25 es un diagrama de flujo que muestra una técnica para codificar información binaria en un plano de bits en un modo de codificación normal 6.

La Figura 26 es un diagrama de flujo que muestra una técnica para codificar información binaria en un modo de codificación diferencial.

La Figura 27 es un diagrama de flujo que muestra una técnica para descodificar información binaria codificada en un modo de codificación diferencial.

La Figura 28 es un diagrama de flujo que muestra una técnica para codificar de manera selectiva información binaria en modo de codificación en bruto para aplicaciones de latencia baja.

DESCRIPCIÓN DETALLADA

Las realizaciones descritas se refieren a técnicas y herramientas para codificar y descodificar (por ejemplo, en un codificador/descodificador de video) información binaria. La información binaria puede comprender bits que indican si un codificador o descodificador de video omite ciertos macrobloques en un fotograma de video. O, la información binaria puede comprender bits que indican la resolución del vector de movimiento para macrobloques (por ejemplo 1-MV o 4-MV), modo entrelazado (por ejemplo, campo o fotograma), o alguna otra información. La información binaria puede ser codificada sobre una base fotograma por fotograma o sobre alguna otra base.

En algunas realizaciones, la información binaria está dispuesta en un plano de bits. El plano de bits es codificado en la capa de imagen/fotograma. Alternativamente, la información binaria está dispuesta de algún otro modo y/o codificada en una capa diferente.

En algunas realizaciones, el codificador y descodificador conmutan modos de codificación. Por ejemplo, el codificador y descodificador utilizan modos normal, de omisión de fila, o de omisión de columna. Los diferentes modos permiten al codificador y descodificador explotar la redundancia en la información binaria. Alternativamente, el codificador y descodificador utilizan otros modos y/o modos adicionales.

En algunas realizaciones, el codificador y descodificador definen un macrobloque omitido como un macrobloque predicho cuyo movimiento es igual a su movimiento predicho causalmente y que tiene error residual cero. Alternativamente, el codificador y descodificador definen un macrobloque omitido como un macrobloque predicho con movimiento cero y error residual cero.

En algunas realizaciones, en vez de codificación de fotograma/nivel de imagen eficiente, un modo de codificación en bruto es permitido para permitir aplicaciones de baja latencia. En el modo de codificación en bruto, los macrobloques codificados pueden ser transmitidos al descodificador inmediatamente, sin tener que esperar a que todos los macrobloques en el fotograma/imagen sean codificados.

En algunas realizaciones, el codificador y descodificador procesan los planos de bits de información de nivel de macrobloque. Alternativamente, el codificador y descodificador procesan planos de bits de información de bloque, sub-bloque, o nivel de píxel.

Las distintas técnicas y herramientas pueden ser utilizadas en combinación o independientemente. En particular, la solicitud describe dos implementaciones de codificación y descodificación de macrobloque omitido, junto con la sintaxis de corriente de bits correspondiente. Diferentes realizaciones implementan una o más de las técnicas y herramientas descritas.

En las realizaciones descritas, el codificador y descodificador de video realizan distintas técnicas. Aunque las operaciones para estas técnicas son descritas típicamente en un orden particular, secuencial por el bien de la presentación, debería entenderse que esta manera de descripción abarca reordenamientos menores en el orden de las operaciones, a no ser que sea requerida una ordenación particular. Por ejemplo, las operaciones descritas secuencialmente pueden en algunos casos ser reordenadas o realizadas concurrentemente. Además, con objeto de simplicidad, los diagramas de flujo típicamente no muestran los distintos modos en los que pueden ser utilizadas las

técnicas particulares en unión con otras técnicas.

En las realizaciones descritas, el codificador y descodificador de video utilizan distintos indicadores y señales en una corriente de bits. Aunque se han descrito indicadores y señales específicos, debería entenderse que esta manera de descripción abarca diferentes convenciones (por ejemplo, 0 en vez de 1) para los indicadores y señales.

I. Entorno informático

La Figura 6 ilustra un ejemplo generalizado de un entorno informático (600) en el cual distintas realizaciones descritas pueden ser implementadas. El entorno informático (600) no pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad, ya que las técnicas y herramientas puede ser implementadas en diversos entornos informáticos de propósito general o de propósito especial.

Con referencia a la Figura 6, el entorno informático (600) incluye al menos una unidad de tratamiento (610) y la memoria (620). En la Figura 6, la configuración más básica (630) es incluida dentro de una línea discontinua. La unidad de tratamiento (610) ejecuta instrucciones ejecutables por ordenador y puede ser un procesador real o virtual. En un sistema multitratamiento, múltiples unidades de tratamiento ejecutan instrucciones ejecutables por ordenador para aumentar la potencia del tratamiento. La memoria (620) puede ser memoria volátil (por ejemplo, registros, caché, RAM), memoria no volátil (por ejemplo ROM, EEPROM, memoria flash, etc.) o alguna combinación de las dos. La memoria (620) almacena software (680) que implementa un codificador o descodificador, tal como un codificador o descodificador de video.

Un entorno informático puede tener características adicionales. Por ejemplo, el entorno informático (600) incluye almacenamiento (640), uno o más dispositivos de entrada (650), uno o más dispositivos de salida (660), y una o más conexiones de comunicación (670). Un mecanismo de interconexión (no mostrado) tal como un bus, controlador, o red interconecta los componentes del entorno informático (600). Típicamente, el software de sistema operativo (no mostrado) proporciona un entorno informático para otro software que funciona en el entorno informático (600), y coordina actividades de los componentes del entorno informático (600).

El almacenamiento (640) puede ser extraíble o no extraíble, e incluye discos magnéticos, cintas magnéticas o casetes, CD ROM, DVD, o cualquier otro medio que puede ser utilizado para almacenar información y que puede ser accedido dentro del entorno informático (600). El almacenamiento (640) almacena instrucciones para el software (680) que implementa el codificador y descodificador.

El o los dispositivos de entrada (650) puede ser un dispositivo de entrada táctil tal como un teclado, ratón, bolígrafo o "trackball" ("bola de seguimiento"), un dispositivo de entrada de voz, un dispositivo de escaneado, u otro dispositivo que proporciona entrada al entorno informático (600). Para codificación de audio o video, el o los dispositivos (650) puede ser una tarjeta de sonido, una tarjeta de video, una tarjeta sintonizadora de TV, o dispositivo similar que acepta entrada de audio o video en forma analógica o digital, o un CD-ROM o CD-RW que lee muestras de audio o video en el entorno informático (600). El o los dispositivos de salida (660) puede ser una pantalla de presentación, impresora, altavoz, escritor de CD u otro dispositivo que proporciona salida desde el entorno informático (600).

La conexión o conexiones de comunicación (670) habilitan la comunicación sobre un medio de comunicación a otra entidad informática. El medio de comunicación transporta información tal como instrucciones ejecutables por ordenador, entrada o salida de audio o video, 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 adaptadas o cambiadas de tal manera que codifica la información en la señal. A modo de ejemplo, y no de limitación, el medio de comunicación incluye técnicas con cable o inalámbricas implementadas con un portador eléctrico, óptico, de RF, de infrarrojos, acústico, u otro portador.

Las técnicas y herramientas pueden ser descritas en el contexto general de medio legible por ordenador. Medios legibles por ordenador son cualquier medio disponible que puede ser accedido dentro de un entorno informático. A modo de ejemplo, y no de limitación, con el entorno informático (600), el medio legible por ordenador incluye memoria (620), almacenamiento (640), medios de comunicación, y combinaciones de cualquiera de los anteriores.

Las técnicas y herramientas pueden ser descritas en el contexto general de instrucciones ejecutables por ordenador, tales como las incluidas en módulos de programas, que son ejecutados en un entorno informático sobre un procesador objetivo real o virtual. Generalmente, los módulos de programa incluyen rutinas, programas, bibliotecas, objetos, clases, componentes, estructuras de datos, etc., que realizan tareas particulares o implementan tipos de datos abstractos particulares. La funcionalidad de los módulos de programa puede ser combinada o dividida entre módulos de programa como se ha descrito en distinta realizaciones. Las instrucciones ejecutables por ordenador para módulos de programa pueden ser ejecutadas dentro de un entorno informático local o distribuido.

Con el propósito de la presentación, la descripción detallada utiliza términos como "determinar", "seleccionar", "reconstruir", e "informar", para describir operaciones de ordenador en un entorno informático. Estos términos son abstracciones de alto nivel para operaciones realizadas por un ordenador, y no deberían ser confundidas con acciones que son realizadas por un ser humano. Las operaciones de ordenador reales correspondientes a estos términos varían dependiendo de la implementación.

II. Codificador y Descodificador de Video Generalizado

La Figura 7 es un diagrama de bloques de un codificador de video generalizado (700) y la Figura 8 es un diagrama de bloques de un decodificador de video generalizado (800).

5 La relación mostrada entre los módulos dentro del codificador y decodificador indica el flujo principal de información en el codificador y decodificador; otras relaciones no han sido mostradas con objeto de simplicidad. En particular, las Figuras 7 y 8 no muestran normalmente información lateral que indica los conjuntos, modos, tablas, etc., de codificador utilizados para una secuencia, fotograma, macrobloque, bloque, etc., de video. Tal información lateral es enviada en la corriente de bits de salida, típicamente después de codificación de entropía de la información lateral. El formato de la corriente de bits de salida puede ser el formato de Windows Media Video versión 8 u otro formato.

10 El codificador (700) y el decodificador (800) están basados en bloques y utilizan un formato de macrobloque 4:2:0 con cada macrobloque que incluye 4 bloques de luminancia 8x8 (a veces tratado como un macrobloque de 16x16) y dos bloques de crominancia 8x8. Alternativamente, el codificador (700) y el decodificador (800) están basados en objetos, utilizan un formato de macrobloque o de bloque diferente, o realizan operaciones sobre conjuntos de píxeles de diferente tamaño o configuración que los bloques de 8x8 y los macrobloques de 16x16.

15 Dependiendo de la implementación y del tipo de compresión deseada, los módulos del codificador o decodificador pueden ser añadidos, omitidos, divididos en múltiples módulos, combinados con otros módulos, y/o reemplazados con módulos similares. En realizaciones alternativas, el codificador o decodificador con diferentes módulos y/u otras configuraciones de módulo realizan una o más de las técnicas descritas.

A. Codificador de Video

20 La Figura 7 es un diagrama de bloques de un sistema codificador (700) de video general. El sistema codificador (700) recibe una secuencia de fotogramas de video que incluyen un fotograma actual (705), y produce información de video comprimida (795) como salida. Las realizaciones particulares de codificadores de video utilizan típicamente una variación o versión complementada del codificador generalizado (700).

25 El sistema codificador (700) comprime los fotogramas predichos y fotogramas clave. Con el propósito de la presentación, la Figura 7 muestra un trayecto para fotogramas clave a través del sistema codificador (700) y un trayecto para fotogramas predichos progresivos. Muchos de los componentes del sistema codificador (700) son utilizados para comprimir tanto fotogramas clave como fotogramas predichos. Las operaciones exactas realizadas por los componentes pueden variar dependiendo del tipo de información que es comprimida.

30 Un fotograma predicho [también llamado p-fotograma, b-fotograma para fotograma de predicción o inter-codificado bidireccional] es representado en términos de predicción (o diferencia) de uno o más de los otros fotogramas. Un residuo de predicción es la diferencia entre lo que se ha predicho y el fotograma original. En contraste, un fotograma clave [también llamado fotograma i, fotograma intra-codificado] es comprimido sin referencia a otros fotogramas.

35 Si el fotograma actual (705) es un fotograma predicho progresivo, un estimador de movimiento (710) estima el movimiento de macrobloques u otros conjuntos de píxeles del fotograma actual (705) con respecto a un fotograma de referencia, que es el fotograma previo reconstruido (725) almacenado temporalmente en el almacenamiento de fotograma (720). En realizaciones alternativas, el fotograma de referencia es un fotograma posterior o el fotograma actual es predicho de manera bidireccional. El estimador de movimiento (710) puede estimar el movimiento por pixel, 1/2 pixel, 1/4 de pixel, u otros incrementos, y puede conmutar la resolución de la estimación de movimiento sobre una base fotograma por fotograma u otra base. La resolución de la estimación de movimiento puede ser la misma o diferente horizontal y verticalmente. El estimador de movimiento (710) emite una información de movimiento de información lateral (715) tal como vectores de movimiento. Un compensador de movimiento (730) aplica la información de movimiento (715) al fotograma previo reconstruido (725) para formar un fotograma actual de movimiento compensado (735). La predicción es raramente perfecta, sin embargo, y la diferencia entre el fotograma actual de movimiento compensado (735) y el fotograma actual original (705) es el residuo de predicción (745). Alternativamente, un estimador de movimiento y compensador de movimiento aplica otro tipo de estimación/compensación de movimiento.

40 Un transformador de frecuencia (760) convierte la información de video de dominio espacial en datos de dominio de frecuencia (es decir, espectral). Para fotogramas de video basado en bloques, el transformador de frecuencia (760) aplica una transformada de coseno discreta ["DCT"] o variante de DCT para bloques de datos de pixel o datos de residuo de predicción, produciendo bloques de coeficientes DCT. Alternativamente, el transformador de frecuencia (760) aplica otra transformada de frecuencia convencional tal como una transformada de Fourier o utiliza análisis de "ondículas" o de sub-banda. En realizaciones en que el codificador utiliza extrapolación espacial (no mostrado en la Figura 7) para codificar bloques de fotogramas clave, el transformador de frecuencia (760) puede aplicar una transformada de frecuencia reorientada tal como una DCT sesgada para bloques de residuos de predicción para el fotograma clave. En otras realizaciones, el transformador de frecuencia (760) aplica unas transformadas de tamaño de frecuencia de 8x8, 8x4, 4x8 o de otro tamaño (por ejemplo, DCT) para residuos de predicción para fotogramas predichos.

65 Un cuantificador (770) cuantifica entonces los bloques de coeficientes de datos espectrales. El cuantificador aplica una cuantificación escalar, uniforme a los datos espectrales con un tamaño de paso que varía en una base de fotograma por

fotograma o en otra base. 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 directamente cuantifica los datos de dominio espacial en un sistema codificador que no utiliza transformadas de frecuencia. Además de la cuantificación adaptable, el codificador (700) puede utilizar disminución de fotograma, filtrado adaptativo, u otras técnicas para el control de tasa.

Si un macrobloque dado en un fotograma predicho no tiene información de ciertos tipos (por ejemplo no tiene información para el macrobloque y e información no residual), el codificador (700) puede codificar el macrobloque como un macrobloque omitido. Si es así, el codificador señala el macrobloque omitido en la corriente de bits de salida de la información de video comprimida (795).

Cuando un fotograma actual reconstruido es necesario para la estimación/compensación del movimiento subsiguiente, un cuantificador inverso (776) realiza una cuantificación inversa sobre los coeficientes de datos espectrales cuantificados. Un transformador de frecuencia inversa (766) realiza entonces la inversa de las operaciones del transformador de frecuencia (760), produciendo un residuo de predicción reconstruido (para un fotograma predicho) o un fotograma clave reconstruido. Si el fotograma actual (705) era un fotograma clave, el fotograma clave reconstruido es tomado como el fotograma actual reconstruido (no mostrado). Si el fotograma actual (705) era un fotograma predicho, el residuo de predicción reconstruido es añadido al fotograma actual de movimiento compensado (735) para formar el fotograma actual reconstruido. El almacenamiento (720) de fotogramas almacena temporalmente el fotograma actual reconstruido para utilizar 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 (780) de entropía comprime la salida del cuantificador (770) así como cierta información lateral (por ejemplo información de movimiento (715), modos de extrapolación espacial, tamaño del paso de cuantificación). Las técnicas de codificación de entropía típicas incluyen codificación aritmética, codificación diferencial, codificación de Huffman, codificación de longitud de recorrido, codificación de LZ, codificación de diccionario, y combinaciones de las anteriores. El codificador (780) de entropía utiliza típicamente diferentes técnicas de codificación para diferentes tipos de información (por ejemplo, coeficientes de DC, coeficientes de AC, diferentes tipos de información lateral), y puede elegir de entre múltiples tablas de códigos dentro de una técnica de codificación particular.

El codificador (780) de entropía pone la información (795) de video comprimida en la memoria tampón (790). Un indicador de nivel de memoria tampón es alimentado positivamente a los módulos adaptables de tasa de bits.

La información (795) de video comprimida es omitida de la memoria tampón (790) a una tasa de bits constante o relativamente constante y almacenada durante la subsiguiente retransmisión a esa tasa de bits. Por ello, el nivel de la memoria tampón (790) es fundamentalmente una función de la entropía de la información de video filtrada, cuantificada, que afecta a la eficiencia de la codificación de entropía. Alternativamente, el sistema codificador (700) retransmite la información de video comprimida inmediatamente después de la compresión, y el nivel de la memoria tampón (790) también depende de la tasa a la que la información es vaciada desde la memoria tampón (790) para su transmisión.

Antes o después de la memoria tampón (790), la información (795) de video comprimida puede ser codificada en canal para transmisión sobre la red. La codificación en canal puede aplicar detección de error y datos de corrección a la información (795) o de video comprimida.

B. Decodificador de video

La Figura 8 es un diagrama de bloques de un sistema decodificador (800) de video general. El sistema decodificador (800) recibe información (895) durante una secuencia comprimida de fotogramas de video y produce una salida que incluye un fotograma reconstruido (805). Realizaciones particulares de decodificadores de video utilizan típicamente una variación o versión complementada del decodificador generalizado (800).

El sistema decodificador (800) descomprime los fotogramas predichos y los fotogramas clave. Con propósito de presentación, la Figura 8 muestra un trayecto para los fotogramas clave a través del sistema decodificador (800) y un trayecto para los fotogramas predichos previamente. Muchos de los componentes del sistema decodificador (800) son utilizados para comprimir tanto fotogramas clave como fotogramas predichos. Las operaciones exactas realizadas por aquellos componentes pueden variar dependiendo del tipo de información que es comprimida.

Una memoria tampón (890) recibe la información (895) para la secuencia de video comprimida y hace que la información recibida esté disponible para el decodificador de entropía (880). La memoria tampón (890) recibe típicamente la información a una tasa que es casi constante a lo largo del tiempo, e incluye una memoria tampón fluctuante para suavizar las variaciones a corto plazo en el ancho de banda o en la transmisión. La memoria tampón (890) puede incluir una memoria tampón de reproducción y otras memorias tampón también. Alternativamente, la memoria tampón (890) recibe información a una tasa variable. Antes o después de la memoria tampón (890), la información de video comprimida puede ser descodificada en canal y procesada para detección de error y corrección.

El decodificador de entropía (880) descodifica la entropía de datos cuantificados codificados de entropía así como la

información lateral codificada de entropía (por ejemplo, información de movimiento (815), modos de extrapolación espacial, tamaño del paso de cuantificación), aplicando típicamente la inversa de la codificación de entropía realizada en el codificador. Las técnicas de descodificación de entropía incluyen descodificación aritmética, descodificación diferencial, descodificación de Huffman, descodificación de longitud de recorrido, descodificación LZ, descodificación de diccionario, y combinaciones de las anteriores. El descodificador de entropía (880) utiliza frecuentemente diferentes técnicas de descodificación para diferentes tipos de información (por ejemplo, coeficientes de DC, coeficientes de AC, diferentes tipos de información lateral), y puede elegir de entre múltiples tablas de códigos dentro de una técnica de descodificación particular.

Si el fotograma (805) que ha de ser reconstruido es un fotograma predicho previamente, un compensador de movimiento (830) aplica información de movimiento (815) a un fotograma de referencia (825) para formar una predicción (835) del fotograma (805) que está siendo reconstruido. Por ejemplo, el compensador de movimiento (830) utiliza un vector de movimiento de macrobloque para encontrar un macrobloque en el fotograma de referencia (825). Una memoria tampón (820) del fotograma almacena fotogramas reconstruidos previos para utilizar como fotogramas de referencia. El compensador de movimiento (830) puede compensar el movimiento en un píxel, 1/2 píxel, 1/4 de píxel, u otros incrementos, y puede conmutar la resolución de la compensación de movimiento sobre una base de fotograma por fotograma o sobre otra base. La resolución de la compensación de movimiento puede ser la misma o diferente horizontal y verticalmente. Alternativamente, un compensador de movimiento aplica otro tipo de compensación de movimiento. La predicción por el compensador de movimiento es raramente perfecta, así el descodificador (800) también reconstruye residuos de predicción.

Cuando el descodificador necesita un fotograma reconstruido para la subsiguiente compensación de movimiento, el almacenamiento (820) de fotogramas almacena temporalmente el fotograma reconstruido para utilizar en la predicción del siguiente fotograma. En algunas realizaciones, el codificador aplica un filtro de desbloqueo al fotograma reconstruido para suavizar adaptativamente discontinuidades en los bloques del fotograma.

Un cuantificador inverso (870) cuantifica de manera inversa los datos descodificados de entropía. En general, el cuantificador inverso aplica una cuantificación inversa uniforme, escalar a los datos descodificados de entropía con un tamaño de paso que varía sobre una base de fotograma por fotograma o sobre otra base. Alternativamente, el cuantificador inverso aplica otro tipo de cuantificación inversa a los datos, por ejemplo, una cuantificación no uniforme, vectorial, o no adaptativa, o cuantifica de manera inversa directamente los datos de dominio espacial en un sistema descodificador que no utiliza transformaciones de frecuencia inversas.

Un transformador (860) de frecuencia inversa convierte los datos de dominio de frecuencia cuantificados en información de video de dominio espacial. Para fotogramas de video a base de bloques el transformador (860) de frecuencia inversa aplica una DCT ["IDCT"] o variante de IDCT a los bloques de los coeficientes de DCT, produciendo datos de píxel o datos residuales de predicción para fotogramas clave o fotogramas predichos, respectivamente. Alternativamente, el transformador (860) de frecuencia aplica otra transformada de frecuencia inversa convencional tal como una transformación de Fourier o utiliza síntesis de ondículas o de sub-banda. En realizaciones en las que el descodificador utiliza extrapolación espacial (no mostrada en la Figura 8) para descodificar bloques de fotogramas clave, el transformador (860) de frecuencia inversa puede aplicar una transformación de frecuencia inversa reorientada tal como una IDCT sesgada a bloques de residuos de predicción para el fotograma clave. En otras realizaciones, el transformador (860) de frecuencia inversa aplica una transformada de 8×8 , 8×4 , 4×8 , o transformadas de frecuencia inversa de otros tamaños (por ejemplo IDCT) para residuos de predicción para fotogramas predichos.

Cuando un macrobloque omitido es señalado en la corriente de bits de información (895) para una secuencia comprimida de fotogramas de video, el descodificador (800) reconstruye el macrobloque omitido sin utilizar la información (por ejemplo información de movimiento y/o información residual) normalmente incluida en la corriente de bits para macrobloques no omitidos.

III. Primera implementación

En una primera implementación, un codificador y descodificador de video codifica y descodifica respectivamente, información de macrobloques omitidos con una eficiencia mejorada. La información de macrobloques omitidos es señalada en la capa de imágenes en la corriente de bits de video, lo que permite al codificador explotar la redundancia en la información de macrobloques omitidos. También, el codificador y el descodificador seleccionan entre múltiples modos de codificación para codificar y descodificar la información de macrobloques omitidos.

A. Codificación de Capa de Imágenes de Información de Macrobloques Omitidos

En la primera implementación, una secuencia de video comprimida es construida a partir de datos estructurados en cuatro capas jerárquicas. Desde la superior a la inferior son: 1) capa de secuencia; 2) capa de imágenes; 3) capa de macrobloques; y 4) capa de bloques. En la capa de imágenes, los datos para cada imagen consisten de un encabezamiento de imagen seguido por datos para la capa de macrobloques. (Similarmemente, en la capa de macrobloques, los datos para cada macrobloque consisten de un encabezamiento de macrobloque seguido por la capa de bloques). Aunque algunos de los elementos de la corriente de bits para imágenes I e imágenes P son idénticos, otros aparecen solamente en las imágenes P, y viceversa.

La Figura 9 muestra los elementos de corriente de bits que constituyen la capa (900) de imagen P. La Tabla 1 describe brevemente los elementos de corriente de bits de la capa (900) de imagen P.

Tabla 1: elementos de corriente de bits de la capa de imagen P en la primera implementación

5

Campo	Descripción
PTYPE (910)	Tipo de imagen
PQUANT (912)	Escala cuantificadora de imagen
SMBC (920)	Código de macrobloques omitidos
SMB (930)	Campo de macrobloques omitidos
CPBTAB (940)	Tabla de diseño de bloque codificado
MVRES (942)	Resolución de vector de movimiento
TTMBF (944)	Indicador de tipo de transformada de nivel de macrobloque
TTFRM (946)	Tipo de transformada de nivel de fotograma
DCTACMBF (948)	Indicador de ajuste de codificación DCT AC de nivel de macrobloque
DCTACFRM (950)	Índice de ajuste de codificación DCT AC de nivel de fotograma
DCTDCTAB (952)	Tabla de Intra DCT AC
MVTAB (954)	Tabla de vector de movimiento
MB LAYER (960)	Capa de macrobloques

En particular, la capa (900) de imagen P incluye un campo ("SMB") (930) de Macrobloques Omitidos para los macrobloques en la imagen P así como un campo (920) de código de macrobloques omitidos ("SMBC") que señala el modo de codificación para el campo (930) de macrobloques omitidos. El campo (920) SBMC está presente sólo en los encabezamientos de imagen P. El SBMC (920) es un valor de 2 bits que señala uno de cuatro modos utilizados para indicar los macrobloques omitidos en el fotograma. En la primera implementación, los códigos de longitud fija ("FLC") para los modos de codificación de macrobloques omitidos son como sigue:

15

Tabla 2: Tabla de códigos de modo de codificación de macrobloques en primera implementación

SMBC FLC	Modo de Codificación de Bit Omitido
00	Codificación de bits no omitidos
01	Codificación de bits omitido normal
10	Codificación de bits omitidos Predicción de fila (u, "omisión de fila")
11	Codificación de bits omitidos de predicción de columna (u, "omisión de columna")

Si el modo de codificación es normal, predicción de fila, o predicción de columna, entonces el siguiente campo en la corriente de bits es el campo SMB (930) que contiene la información de macrobloques omitidos. Así, el campo SMB está presente solamente en los encabezamientos de imagen P y solamente si SMBC señala codificación de macrobloques omitidos normal, de predicción de fila, o de predicción de columna. Si SMBC señala codificación normal, entonces el tamaño del campo SMB es igual al número de macrobloques en el fotograma. Si SMBC señala predicción de fila o predicción de columna, entonces el tamaño del SMB es variable como se ha descrito a continuación.

La información de macrobloques omitidos informa al descodificador en cuanto a qué macrobloques en el fotograma no están presentes en la capa de macrobloques. Para estos macrobloques, el descodificador copiará los datos de píxel de macrobloque correspondientes a partir del fotograma de referencia cuando se reconstruya ese macrobloque.

B. Conmutación de Modos de Codificación para Información de Macrobloques Omitidos

Como se ha descrito anteriormente, el campo (920) SMBC señala el modo de codificación para el campo (930) de macrobloques omitidos. Más generalmente, la Figura 10 muestra una técnica (1000) para codificar información de macrobloques omitidos en un codificador de video que tiene múltiples modos de modificación de macrobloques omitidos. La Figura 11 muestra una técnica correspondiente (1100) para descodificar información de macrobloques omitidos codificada por un codificador de video que tiene varios modos de codificación de macrobloques omitidos.

Con referencia a la Figura 10, el codificador selecciona un modo de codificación por omisión de macrobloques para codificar información (1010) de macrobloques omitidos. Por ejemplo, en la primera implementación, los modos de codificación por omisión de macrobloques incluyen un modo en el que no se han omitido macrobloques, un modo normal, un modo de predicción de fila (u, "omisión de fila"), y un modo de predicción de columna (u "omisión de columna"). Después de que se haya seleccionado el modo de codificación, el codificador codifica la información (1020) de macrobloques omitidos. El codificador selecciona modos de codificación sobre una base de imagen por imagen. Alternativamente, el codificador selecciona modos de codificación en alguna otra base (por ejemplo, a nivel de secuencia). Cuando el codificador ha hecho la codificación de la información (1030) de macrobloques omitidos, la codificación termina.

Con referencia a la Figura 11, el descodificador determina el modo de codificación de macrobloques omitidos utilizado

por el codificador para codificar la información (1110) de macrobloques omitidos. El descodificador descodifica entonces la información (1120) de macrobloques omitidos. El descodificador determina modos de codificación sobre una base de imagen por imagen. Alternativamente, el descodificador determina modos de codificación en alguna otra base (por ejemplo, al nivel de secuencia). Cuando el descodificador ha hecho la descodificación de la información (1130) de macrobloques omitidos, la descodificación termina.

C. Modos de Codificación

En la primera implementación, los modos de codificación de macrobloques omitidos incluyen un modo en el que no son omitidos macrobloques. Un modo normal, un modo de predicción de fila (u, "omisión de fila"), y un modo de predicción de columna (u "omisión de columna"). Las siguientes secciones describen como es codificada la información de macrobloques omitidos en cada modo con referencia a la Figura 12, que muestra un ejemplo (1200) de un fotograma de codificación de macrobloques omitidos.

1. Modo de Codificación Normal de Macrobloques Omitidos

En modo normal, el estatus de omitido/no omitido de cada macrobloque está representado con un bit. Por ello, el tamaño del campo SMB en bits es igual al número de macrobloques en el fotograma. La posición de bit dentro del campo SMB corresponde al orden de escaneado de trazo de los macrobloques dentro del fotograma comenzando con el macrobloque superior izquierdo. Un valor de bit de 0 indica que el macrobloque correspondiente no es omitido; un valor de bit de 1 indica que el macrobloque correspondiente es omitido.

La Figura 13 muestra una técnica (1300) para codificar en un modo de codificación normal de macrobloques omitidos. En primer lugar, el codificador comprueba si la codificación de un macrobloque será omitida (1310). Si es así, el codificador añade un valor de bit de 1 al campo SMB para indicar que el macrobloque correspondiente es omitido (1320). En caso contrario, el codificador añade un valor de bit de 0 al campo SMB para indicar que el macrobloque correspondiente no es omitido (1330). Cuando el codificador ha hecho la adición de bits al campo SMB (1340) la codificación de macrobloques omitidos termina.

Como ejemplo, utilizando codificación de modo normal, el campo SMB para el fotograma ejemplar (1200) en la Figura 12 sería codificado como: 01001011111111111010010.

2. Modo de Codificación de Macrobloques Omitidos de Predicción de Fila

En modo de predicción de fila, el estatus de cada fila de macrobloques (desde la parte superior a la inferior) está indicado con un bit. Si el bit es 1, entonces la fila contiene todos los macrobloques omitidos y el estatus para la siguiente fila continúa. Si el bit es igual a 0, entonces el estatus omitido/no omitido para cada macrobloque en esa fila es señalado con un bit. Por ello, un campo de bits es de longitud igual al número de macrobloques en una fila continua. Los bits en el campo de bits representan los macrobloques en orden de izquierda a derecha. De nuevo, un valor de 0 indica que el macrobloque correspondiente no es omitido; un valor de 1 indica que el macrobloque correspondiente es omitido.

La Figura 14 muestra una técnica (1400) para codificar en modo de codificación de macrobloques de predicción de fila (u, "omisión de fila"). En primer lugar, el codificador comprueba si una fila contiene todos los macrobloques omitidos (1410). Si es así, el codificador añade un bit indicador de 1 al campo SMB (1420) y el estatus para la siguiente fila continúa. Si la fila no contiene todos los macrobloques omitidos, el codificador añade un bit indicador de 0 al campo SMB, y el estatus omitido/no omitido para cada macrobloque en esa fila es señalado con un bit (1430). Cuando el codificador lo ha hecho con todas las filas en el fotograma (1440), la codificación de predicción de fila termina.

Como para la descodificación, la Figura 15 muestra un pseudo-código (1500) que ilustra la descodificación de predicción de fila de la información de macrobloques omitidos. En el segundo código (1500), la función `get_bits(n)` lee n bits de la corriente de bits y devuelve el valor.

Como ejemplo, utilizando la codificación de modo de predicción de fila, el campo SMB para el fotograma ejemplar (1200) en la Figura 12 sería codificado como: 0010010110010010.

3. Modo de Codificación de Macrobloques Omitidos de Predicción de Columna.

En modo de predicción de columna, el estatus de cada columna de macrobloques (de izquierda a derecha) está indicado con un bit. Si el bit es 1, entonces la columna contiene todos los macrobloques omitidos y el estatus para la siguiente columna continúa. Si el bit es igual a 0, entonces el estatus omitido/no omitido para cada macrobloque en esa columna es señalado con un bit. Por ello, un campo de bits de longitud igual al número de macrobloques en esa columna continúa. Los bits en el campo de bits representan los macrobloques en orden de superior a inferior. De nuevo, un valor de 0 indica que el macrobloque correspondiente no es omitido; un valor de 1 indica que el macrobloque correspondiente es omitido.

La Figura 16 muestra una técnica (1600) para codificar en modo de codificación de macrobloques de predicción de columna (u, "omisión de columna"). En primer lugar, el codificador comprueba si la columna contiene todos los macrobloques omitidos (1610). Si es así, el codificador añade un bit indicador de 1 al campo SMB (1620) y el estatus para la siguiente columna continúa. Si la columna no contiene todos los macrobloques omitidos, el codificador añade un bit indicador de 0 al campo SMB, y el estatus omitido/no omitido para cada macrobloque en esa columna es señalado

con un bit (1630). Cuando el codificador lo ha hecho con todas las columnas en el fotograma (1640), la codificación de predicción de columna termina.

5 Como para la descodificación, la Figura 17 muestra un pseudo-código (1700) que ilustra la descodificación de predicción de columna fila de la información de macrobloques omitidos.

Como ejemplo, utilizando la codificación de modo de predicción de columna, el campo SMB para el fotograma ejemplar (1200) en la Figura 12 sería codificado como: 0011010011000110100110.

10 IV. Segunda Implementación

En una segunda implementación, un codificador y descodificador de video codifica y descodifica respectivamente, información de macrobloques omitidos y/u otros datos binarios de 2D con una eficiencia mejorada. El codificador y descodificador definen un macrobloque omitido como el que tiene un movimiento por defecto (no necesariamente un movimiento cero), que permite que el codificador y el descodificador borren más macrobloques en muchos casos. La codificación eficiente a nivel de fotograma de planos de bits indica la información de macrobloque omitido y/u otros datos binarios de 2D. También, el codificador y descodificador pueden utilizar una opción de codificación en bruto (nivel MB) de macrobloques omitidos para aplicaciones de latencia baja.

20 A. Definición de Omisión de Bit (Definición de Macrobloque Omitido)

La segunda implementación, incluye una nueva definición del concepto de un macrobloque omitido. "Omitido" se refiere a un estado en una corriente de bits en el que no se necesita que más información sea transmitida a ese nivel de granularidad. Un macrobloque (bloque) omitido es un macrobloque (bloque) que tiene un tipo por defecto, movimiento por defecto, y error residual por defecto. (En comparación, en otras implementaciones y normas, los macrobloques omitidos son macrobloques predichos con movimiento cero y residuos cero).

25 La nueva definición de macrobloque omitido es un macrobloque predicho cuyo movimiento es igual a su movimiento predicho causalmente, y que tiene un error residual cero. (El punto de diferencia de la otra definición es que el movimiento por defecto es igual al predictor de movimiento, y éste puede no necesariamente ser cero).

30 Por ejemplo, en algunas realizaciones, los vectores de movimiento predicho para un macrobloque actual son tomados del macrobloque directamente situado por encima o directamente a la izquierda del macrobloque actual. Ahora bien, los componentes horizontal y vertical del predictor son generados a partir de medianas de modo componente horizontal y vertical de los macrobloques a la izquierda, parte superior y parte superior derecha del macrobloque actual.

35 Los vectores de movimiento de un macrobloque omitido con cuatro vectores de movimiento (4MV) vienen dados por sus predicciones realizadas secuencialmente en el orden de escaneado natural. Como con el caso de vector de un movimiento (1MV), los residuos de error son cero.

40 La Figura 18 muestra una técnica (1800) para determinar si hay que omitir la codificación de macrobloques particulares en un codificador de video de acuerdo a la nueva definición de macrobloques omitidos. En primer lugar, el codificador comprueba si el fotograma actual es un fotograma I o un fotograma P (1810). Si el fotograma actual es un fotograma I, no hay macrobloques omitidos en el fotograma actual (1820), y la codificación de macrobloque omitido para el fotograma termina.

45 Por otro lado, si el fotograma actual es un fotograma P, el codificador comprueba los macrobloques en el fotograma corriente que pueden ser omitidos. Para un macrobloque dado, el codificador comprueba si el vector de movimiento para el macrobloque es igual al vector de movimiento predicho causalmente para el macrobloque (por ejemplo, si el vector de movimiento diferencial para el macrobloque es igual a cero) (1830). Si el movimiento para un macrobloque no es igual al movimiento predicho causalmente, el codificador no omite el macrobloque (1840). De otro modo, el codificador comprueba si hay algún residuo que ha de ser codificado para el macrobloque (1850). Si hay un residuo que ha de ser codificado, el codificador no omite el macrobloque (1860). Si no hay residuo para el macrobloque, sin embargo, el codificador omite el macrobloque (1870). El codificador continúa codificando u omitiendo macrobloques hasta que se ha hecho la codificación (1880).

55 B. Codificación de Plano de Bits

En la segunda implementación, cierta información específica de macrobloque (incluyendo la señalización de macrobloques omitidos) puede ser codificada en un bit por macrobloque. El estatus para todos los macrobloques en un fotograma puede ser codificado conjuntamente como un plano de bits y transmitido en el encabezamiento del fotograma.

60 En la segunda implementación, el codificador utiliza la codificación en plano de bits en tres casos para señalar la información acerca de los macrobloques en un fotograma. Los tres casos son: 1) señalar macrobloques omitidos, 2) señalar campo o modo de macrobloque de fotograma, y 3) señalar modo de vector de movimiento de 1 MV o de 4 MV para cada macrobloque. Esta sección describe la codificación de plano de bits para cualquiera de los tres casos y la descodificación correspondiente.

65 La codificación de plano de bits a nivel de fotograma es utilizada para codificar agrupaciones binarias bidimensionales. El

tamaño de cada agrupación es rowMB x colMB, donde rowMB y colMB son el número de filas y columnas del macrobloque, respectivamente. Dentro de la corriente de bits, cada agrupación es codificada como un conjunto de bits consecutivos. Uno de los siete modos es utilizado para codificar cada agrupación, como se ha enumerado en la Tabla 3 y descrito a continuación.

5

Tabla 3: Modos de codificación en segunda implementación

Modo de Codificación	Descripción
En bruto	Codificado como un bit por símbolo
Normal-2	Dos símbolos codificados conjuntamente
Diff-2	Codificación diferencial de plano de bits, seguido por codificación de dos símbolos residuales conjuntamente
Normal-6	Seis símbolos codificados conjuntamente
Diff-6	Codificación diferencial de plano de bits, seguido por codificación de seis símbolos residuales conjuntamente
Omisión de fila	Omisión de un bit para señalar filas sin bits establecidos
Omisión de columna	Omisión de un bit para señalar columnas sin bits establecidos

10

En la segunda implementación, el codificador utiliza tres elementos de sintaxis para integrar la información en un plano de bits: MODE, INVERT y DATABITS.

15

El campo MODE es un código de longitud variable ("VLC") que codifica el modo de codificación para el plano de bits. Por ejemplo, el VLC en el campo MODE representa cualquiera de los siete modos de codificación enumerados en la Tabla 3. Para ahorrar bits, el codificador puede asignar códigos más cortos a los modos de codificación más probables y códigos más largos a los modos de codificación menos probables. Como se ha observado anteriormente, el campo MODE es transmitido en el encabezamiento del fotograma.

20

El codificador y descodificador conmutarán entre modos de codificación sobre una base de fotograma por fotograma. Por ejemplo, el codificador y el descodificador conmutan entre modos de codificación de modo similar a como el codificador y descodificador de la primera implementación conmutan entre modos de codificación de macrobloque omitido en las Figuras 10 y 11, respectivamente. Alternativamente, el codificador y descodificador conmutan utilizando alguna otra técnica y/o sobre alguna otra base.

25

Si el modo no es un modo en bruto, el campo INVERT de un bit es enviado. En varios modos de codificación donde la inversión condicional puede ser realizada, el campo INVERT indica si los bits en el plano de bits han de ser invertidos antes de que la codificación tenga lugar en el codificador y si la salida de descodificación en el descodificador ha de ser invertida. El campo INVERT es 1 cuando la mayor parte de los bits en el plano de bits es igual a 1, y 0 cuando la mayor parte de los bits en el plano de bits es igual a 0. El codificador emplea varios modos de codificación (tales como normal 2 y normal 6) que consumen menos bits cuando hay presentes más 0. Si el plano de bits que ha de ser codificado tiene más 1 que 0, el codificador puede invertir el plano de bits para aumentar la proporción de 0 en el plano de bits y aumentar el potencial para ahorrar bits. Otros modos (tales como diff-2 y diff-6) utilizan el valor del INVERT para calcular un plano de bits de predictor. Por ello, en algunos modos de codificación, el plano de bits final reconstruido en el descodificador depende de INVERT.

35

El campo DATABITS es una corriente codificada de entropía de símbolos VLC que contienen la información necesaria para reconstruir el plano de bits, dados los campos MODE e INVERT.

40

C. Modos de Codificación

En la segunda implementación, el codificador codifica información binaria (por ejemplo, información de macrobloques omitidos) en cualquiera de siete modos de codificación diferentes: modo de omisión de filas, modo de omisión de columnas, modo normal 2, modo normal 6, modo diff-2, modo diff-6, y modo en bruto. Un descodificador realiza la descodificación correspondiente para cualquiera de los siete modos de codificación. Cada modo que está descrito en detalle a continuación.

45

Alternativamente, el codificador y el descodificador utilizan otros modos de codificación y/o modos adicionales.

1. Modos de Omisión de fila y de Omisión de columna

50

El modo de codificación de omisión de fila ahorra bits representando una fila en un plano de bits con un único bit si cada símbolo binario en la fila es de un cierto valor. Por ejemplo, el codificador representa un macrobloque omitido con un 0 en un plano de bits, y utiliza un modo de codificación de omisión de fila que representa una fila de todo 0 con un único bit. El codificador ahorra por ello bits cuando se omiten filas completas de macrobloques. El descodificador realiza la descodificación correspondiente.

55

En la segunda implementación, las filas con todo ceros son indicadas utilizando un bit establecido a 0. Cuando la fila no es toda de ceros, el indicador de un bit es establecido a 1, y esto va seguido por los bits colMB que contienen la fila de plano de bits en orden. Las filas son escaneadas en el orden natural.

5 De modo similar, para el modo de omisión de columna, si la fila completa es cero, es enviado un bit 0. Si no, es enviado un 1, seguido por los bits rowMB que contienen la columna completa, en orden. Las columnas son escaneadas en el orden natural.

10 Para la codificación de filas y/o columnas sobrantes en modos diff-6 y normal 6 (descritos a continuación), se aplica la misma lógica. Un indicador de un bit indica si la fila o la columna es de todo ceros. Si no, la fila o columna completa es transmitida utilizando un bit por símbolo.

15 Cuando el codificador codifica un plano de bits consistente fundamentalmente de 1, la codificación de omisión de fila y de omisión de columna son usualmente menos eficientes, a causa de la menor probabilidad de que las filas/columnas consistan totalmente de 0. Sin embargo, el codificador puede realizar una inversión en el plano de bits en tal situación para aumentar la proporción de 0 y potencialmente aumentar los ahorros de bits. Así, cuando una inversión condicional es indicada a través del bit INVERT, el codificador invierte previamente el plano de bits antes de que el plano de bits sea puesto en mosaico y codificado. En el lado del descodificador, la inversión condicional es implementada tomando la inversa de la salida final. (Esto no es realizado para el modo diff-2 y diff-6).

20 La Figura 19 muestra una técnica (1900) para codificar información binaria en un plano de bits en un modo de codificación de omisión de fila. El codificador comprueba en primer lugar si es apropiada la inversión del plano de bits, y, si lo es, realiza la inversión (1910). El codificador comprueba entonces una fila en el plano de bits para ver si cada bit en la fila es igual a 0 (1920). Si lo es, el codificador establece el bit indicador para la fila a 0 (1930). Si cualquiera de los bits en la fila no es 0, el codificador establece el bit indicador para la fila a 1 y codifica cada bit en la fila con un bit (1940). Cuando el codificador ha hecho la codificación de todas las filas en el plano de bits (1950), la codificación del plano de bits termina.

25 Un descodificador realiza la descodificación correspondiente para el modo de codificación de omisión de fila.

30 La Figura 20 muestra una técnica para codificar información binaria en un modo de codificación de omisión de columna. El codificador comprueba en primer lugar si es apropiada la inversión del plano de bits, y, si lo es, realiza la inversión (2010). El codificador comprueba entonces una columna en el plano de bits para ver si cada bit en la columna es igual a 0 (2020). Si lo es, el codificador establece el bit indicador para la columna a 0 (2030). Si cualquiera de los bits en la columna no es 0, el codificador establece el bit indicador para la columna en 1 y codifica cada bit en la columna con un bit (1940). Cuando el codificador ha hecho la codificación de todas las columnas en el plano de bits (1950), la codificación del plano de bits termina.

35 Un descodificador realiza la descodificación correspondiente para el modo de codificación de omisión de columna.

40 2. Modo Normal 2

El codificador utiliza el modo normal 2 para codificar conjuntamente varios símbolos binarios en un plano de bits (por ejemplo, utilizando un esquema de codificación de Huffman vectorial u otro de longitud variable). El codificador codifica pares de símbolos binarios con códigos de longitud variable. El descodificador realiza la descodificación correspondiente.

45 Si rowMB x colMB es impar, el primer símbolo es codificado como un solo bit. Los símbolos subsiguientes son codificados en modo de par, en orden de escaneado natural. Una tabla VLC es utilizada para codificar los pares de símbolos para reducir la entropía total.

50 Cuando la inversión condicional está indicada a través del bit INVERTER, el codificador invierte previamente el plano de bits antes de que el plano de bits sea codificado en modo de par. En el lado del descodificador, la inversión condicional es implementada tomando la inversa de la salida final. (Cuando se utiliza el modo diff-2, la inversión condicional no es realizada en esta operación).

55 La Figura 21 muestra una técnica (2100) para codificar información binaria en modo normal 2. El codificador realiza una comprobación inicial para determinar si es apropiada la inversión del plano de bits para mejorar la eficiencia de la codificación y, si lo es, realiza la inversión (2110). El codificador determina a continuación si el plano de bits que es codificado tiene un número impar de símbolos binarios (2120). Si es así, el codificador codifica el primer símbolo con un único bit (2130). El codificador codifica entonces pares de símbolos con códigos de longitud variable, utilizando códigos más cortos para representar los pares más probables y códigos más largos para representar los pares menos probables (2140). Cuando la codificación de los pares de símbolos es hecha (2150), la codificación termina.

60 Un descodificador realiza la descodificación correspondiente para el modo de codificación normal 2.

65 3. Modo Normal 6

El codificador también utiliza el modo normal 6 para codificar conjuntamente varios símbolos binarios en un plano de bits

(por ejemplo, utilizando un esquema de codificación de Huffman vectorial u otro de longitud variable). El codificador pone en mosaico grupos de seis símbolos binarios y representa cada grupo con un código de longitud variable. El decodificador realiza la decodificación correspondiente.

5 En el modo normal 6 (y en el modo diff-6), el plano de bits es codificado en grupos de seis píxeles. Estos píxeles son agrupados en mosaicos bien de 2×3 o bien de 3×2. El plano de bits es puesto en mosaico máximamente utilizando un conjunto de reglas, y los píxeles restantes son codificados utilizando variantes de modos de omisión de fila y omisión de columna.

10 En la segunda implementación, se utilizan mosaicos "verticales" de 3×2 si, y solamente si, rowMB es un múltiplo de 3 y si colMB no es un múltiplo de 3. De otro modo, se utilizan mosaicos "horizontales" de 2×3. Las Figuras 22, 23 y 24 muestran ejemplos de fotogramas en mosaico en el modo de codificación normal 6. La Figura 22 muestra un fotograma (2200) con mosaicos verticales de 3×2 y un resto amplio de 1 símbolo (mostrado como un área sombreada o rayada) para ser codificada en modo de omisión de columna. La Figura 23 muestra un fotograma (2300) con mosaicos horizontales de 2×3 y un resto amplio de 1 símbolo para ser codificado en un modo de omisión de fila. La Figura 24 muestra un fotograma (2400) con mosaicos horizontales de 2×3 y restos amplios de 1 símbolo para ser codificados en modos de omisión de fila y de omisión de columna.

20 Aunque se han utilizado mosaicos de 3×2 y de 2×3 en este ejemplo, en otras realizaciones, son utilizadas diferentes configuraciones de mosaicos y/o diferentes reglas de formación de mosaicos.

25 Los mosaicos de 6 elementos son codificados en primer lugar, seguidos por los mosaicos lineales codificados de omisión de columna y de omisión de fila. Si el tamaño de la agrupación es un múltiplo de 3×2 o de 2×3, los últimos mosaicos lineales no existen y el plano de bits está perfectamente compuesto de mosaicos. Los mosaicos rectangulares de elementos 6 son codificados utilizando una tabla VLC.

30 Cuando la inversión condicional es indicada a través del bit INVERT, el codificador invierte previamente el plano de bits antes de que el plano de bits sea colocado en mosaicos y codificado. En el lado del decodificador, la inversión condicional es implementada tomando la inversa de la salida final. (Cuando se ha utilizado el modo de diff-6, la inversión condicional no es realizada en esta operación).

35 La Figura 25 muestra una técnica (2500) para codificar información binaria en modo normal 6. El codificador realiza una comprobación inicial para determinar si la inversión del plano de bits es apropiada para mejorar la eficiencia de la codificación y, si lo es, realiza la inversión (2510). El codificador comprueba entonces si el número de filas en el plano de bits es un múltiplo de tres (2520). Si el número de filas no es un múltiplo de tres, el codificador agrupa los símbolos en el plano de bits en mosaicos horizontales de 2×3 (2530).

40 Si el número de filas es un múltiplo de tres, el codificador comprueba si el número de columnas en el plano de bits es un múltiplo de tres (2540). Si el número de columnas es un múltiplo de tres, el codificador agrupa los símbolos en el plano de bits en mosaicos horizontales de 2×3 (2530). Si el número de columnas no es un múltiplo de tres, el codificador agrupa los símbolos en mosaicos verticales de 3×2 (2550).

45 Después de agrupar los símbolos en mosaicos de 3×2 o de 2×3, el codificador codifica los grupos de seis símbolos en mosaico utilizando una técnica tal como una técnica de codificación de Huffman vectorial u otra técnica de codificación (2560). El codificador codifica cualesquiera símbolos restantes sin formar mosaicos utilizando las técnicas de codificación de omisión de fila y/o de omisión de columna descritas anteriormente (2570).

Un decodificador realiza una decodificación correspondiente para el modo de codificación normal 6.

50 En otras realizaciones, un codificador utiliza otras técnicas para codificar los símbolos en mosaico y no en mosaico.

4. Modos de Diff-2 y de Diff-6

55 Los modos de codificación diferenciales tales como el modo de diff-2 y de diff-6 codifican planos de bits generando en primer lugar un plano de bits de bits diferenciales (o residuales) para el plano de bits que ha de ser codificado, basándose en un predictor para el plano de bits que ha de ser codificado. El plano de bits residual es a continuación codificado utilizando, por ejemplo, el modo de codificación normal 2 o normal 6, sin inversión condicional.

60 En la segunda implementación, los modos diff-2 y diff-6 emplean codificación diferencial indicada por la operación diff. Si se utiliza cualquier modo diferencial, un plano de bits de bits diferenciales es generado en primer lugar examinando el predictor $\hat{b}(i, j)$ del plano de bits $b(i, j)$, que es definido como la operación causal:

$$\hat{b}(i, j) = \begin{cases} INVERT & i = j = 0, ob(i, j-1) \neq b(i-1, j) \\ b(0, j-1) & i = 0 \\ b(i-1, j) & \text{otro mod } o \end{cases} \quad (1)$$

5 En otras palabras, el predictor $\hat{b}(i, j)$ de un símbolo binario dado $b(i, j)$ será el símbolo binario justo a la izquierda $b(i-1, j)$ excepto en los siguientes casos especiales:

1. si $b(i, j)$ está en la esquina superior izquierda del plano de bits, o si el símbolo binario anterior $b(i, j-1)$ no es igual al símbolo binario a la izquierda $b(i-1, j)$, el predictor $\hat{b}(i, j)$ es igual al valor de INVERT; o
- 10 2. si 1) no se aplica y $b(i, j)$ está en la columna izquierda ($i=0$), el predictor $\hat{b}(i, j)$ será el símbolo binario anterior $b(i, j-1)$.

En el lado del codificador, la operación diff calcula el plano de bits residual r de acuerdo con:

$$15 \quad r(i, j) = b(i, j) \oplus \hat{b}(i, j) \quad (2)$$

Donde \oplus es la operación O exclusiva. El plano de bits residual es codificado utilizando los modos normal 2 y normal 6 sin inversión condicional.

20 En el lado del decodificador, el plano de bits residual es regenerado utilizando el modo normal apropiado. Subsiguientemente, los bits residuales son utilizados para regenerar el plano de bits original como la diferencia binaria 2D.

$$25 \quad b(i, j) = r(i, j) \oplus \hat{b}(i, j) \quad (3)$$

La Figura 26 muestra una técnica (2600) para codificar información binaria en un modo de codificación diferencial. El codificador calcula un predictor para un plano de bits (2610), por ejemplo, como se ha mostrado en la ecuación 1. El codificador calcula entonces un plano de bits residual, por ejemplo, realizando una operación XOR (O exclusiva) sobre el plano de bits y su predictor (2620). El codificador codifica entonces el plano de bits residual (por ejemplo en modo normal 2 o normal 6) (2630).

La Figura 27 muestra una técnica (2700) para decodificar información binaria codificada en un modo de codificación diferencial. El decodificador decodifica el plano de bits residual (2710) utilizando una técnica de decodificación apropiada, basada en el modo utilizado para codificar el plano de bits residual (por ejemplo modo normal 2 o normal 6). El decodificador calcula también el predictor para el plano de bits (2720), utilizando la misma técnica empleada en el codificador. El decodificador reconstruye entonces el plano de bits original, por ejemplo, realizando una operación XOR (O exclusiva) sobre el plano de bits residual decodificado y el plano de bits de predictor (2730).

5. Modo en Bruto

40 Todos los modos excepto el modo en bruto codifican un plano de bits al nivel de fotograma, que demanda una segunda pasada a través del fotograma durante la codificación. Sin embargo, para situaciones de latencia baja, la segunda pasada puede añadir un retraso inaceptable (por ejemplo debido a que la transmisión del encabezamiento de fotograma y la información de capa de macrobloques es retrasada hasta que el último macrobloque en el fotograma es alcanzado, debido al tiempo gastado codificando el plano de bits).

45 El modo en bruto utiliza el método tradicional de codificar el plano de bits en un bit por símbolo binario en la misma ubicación en la corriente de bits que el resto de la información del nivel de macrobloque. Aunque la codificación de símbolos a nivel de macrobloque no es un nuevo concepto en sí misma, la conmutación de la codificación de símbolos desde el nivel de fotograma a nivel de macrobloque proporciona una alternativa de latencia baja para la codificación a nivel de fotograma.

50 La Figura 28 muestra una técnica (2800) para codificar selectivamente información binaria para un macrobloque en un modo de codificación en bruto para aplicaciones de latencia baja. En primer lugar, el codificador comprueba si ha de utilizar el modo en bruto para codificar la información binaria (2810). Si es así, el codificador codifica un bit a nivel de macrobloque para un macrobloque (2820) y comprobación macrobloque es el último macrobloque en el fotograma (2830). Si el macrobloque no es el último macrobloque en el fotograma, el codificador continúa codificando un bit para el siguiente macrobloque al nivel de macrobloque (2820).

Si el codificador no utiliza el modo de codificación en bruto, el codificador codifica un plano de bits al nivel de fotograma

para los macrobloques en el fotograma (2840). Cuando la codificación de los macrobloques en el fotograma se ha hecho (2850), la codificación termina para el fotograma.

5 Aunque la técnica (2800) muestra modos de computación sobre una base de fotograma por fotograma, alternativamente el codificador conmuta sobre algunas otras bases.

10 Habiendo descrito e ilustrado los principios de nuestra invención con referencia a distintas realizaciones, se reconocerá que las distintas realizaciones pueden ser modificadas en disposición y detalle sin salir de tales principios. Debería comprenderse que los programas, procesos, o métodos descritos en este documento no están relacionados o limitados a ningún tipo particular de entorno de cálculo, a menos que se indique de otro modo. Distintos tipos de entornos de cálculo de propósito general o especializados pueden ser utilizados o realizar operaciones de acuerdo con las enseñanzas descritas en este documento. Elementos de realización mostrados en software pueden ser implementados en hardware y viceversa.

15 A la vista de las muchas posibles realizaciones a las que los principios de nuestro invento pueden ser aplicadas, reivindicamos como nuestro invento la totalidad de tales realizaciones ya que pueden quedar dentro del alcance de las siguientes reivindicaciones y de sus equivalencias.

REIVINDICACIONES

1. Un método implementado por ordenador de tratamiento de una o más imágenes de video, comprendiendo el método:
 - 5 seleccionar (1010, 1110) un modo de codificación a partir de un grupo de varios modos de codificación disponibles, en que el grupo de varios modos de codificación disponibles comprende un modo de codificación de predicción de fila y un modo de codificación de predicción de columna; y
 - 10 procesar (1020, 1120) un plano de bits de acuerdo con el modo de codificación seleccionado, en donde el plano de bits incluye información binaria para varios macrobloques de una imagen de video, en donde la información binaria representa características de los distintos macrobloques de la imagen de video, en donde cada uno de los varios macrobloques incluye varios píxeles, en donde la información binaria incluye un símbolo binario para cada uno de los distintos macrobloques de la imagen de video y **caracterizado por que:**
 - 15 en el modo de codificación de predicción de fila (1900), para una fila de macrobloques, un primer bit (1930) indica si el símbolo binario para cada uno de los macrobloques en la fila tiene un cierto valor (1920) y, si no lo tiene, el primer bit va seguido por un campo de bits (1940) de longitud igual al número de los macrobloques en la fila, indicando cada bit en el campo de bits el valor del símbolo binario para uno correspondiente de los macrobloques en la fila; y
 - 20 en el modos de codificación de predicción de columna (2000) para una columna de los macrobloques, un primer bit (2030) indica si el símbolo binario para cada uno de los macrobloques en la columna tiene un cierto valor (2020) y, si no lo tiene, el primer bit va seguido por un campo de bits (2040) de igual longitud al número de los macrobloques en la columna, indicando cada bit en el campo de bits el valor del símbolo binario para uno correspondiente de los macrobloques en la columna.
- 25 2. El método de la reivindicación 1, en donde el tratamiento es realizado a nivel de imagen.
3. El método de la reivindicación 1 en donde el grupo de varios modos de codificación disponibles comprende además uno o más modos de codificación vectorial de longitud variable.
- 30 4. El método de la reivindicación 1 en donde el grupo de varios modos de codificación disponibles comprende además uno o más modos de codificación diferencial.
5. El método de la reivindicación 1 en donde la información binaria significa si los macrobloques de la imagen de video son omitidos o no son omitidos, y en el que un macrobloque de la imagen de video es omitido si el movimiento para el macrobloque es igual al movimiento predicho del macrobloque y el macrobloque no tiene error residual.
- 35 6. El método de la reivindicación 1 en donde la información binaria en el plano de bits comprende información de macrobloques omitidos.
7. El método de la reivindicación 1 en donde la información binaria en el plano de bits comprende información de recuento del vector de movimiento.
- 40 8. El método de la reivindicación 1 en donde la información binaria en el plano de bits comprende indicadores de campo/fotograma.
- 45 9. Un medio legible por ordenador que almacena instrucciones ejecutables por ordenador para hacer que el sistema informático realice el método de una de las reivindicaciones 1 a 8 durante la descodificación del video, en donde el tratamiento comprende descodificación de entropía de acuerdo con el modo de codificación seleccionado.
- 50 10. Un medio legible por ordenador que almacena instrucciones ejecutables por ordenador para hacer que el sistema informático realice el método de una de las reivindicaciones 1 a 8 durante la codificación de video, en donde el tratamiento comprende codificación de entropía de acuerdo con el modo de codificación seleccionado.
11. Un descodificador de video adaptado para realizar el método de una de las reivindicaciones 1 a 8.
- 55 12. Un codificador de video adaptado para realizar el método de una de las reivindicaciones 1 a 8.

Figura 1, técnica anterior

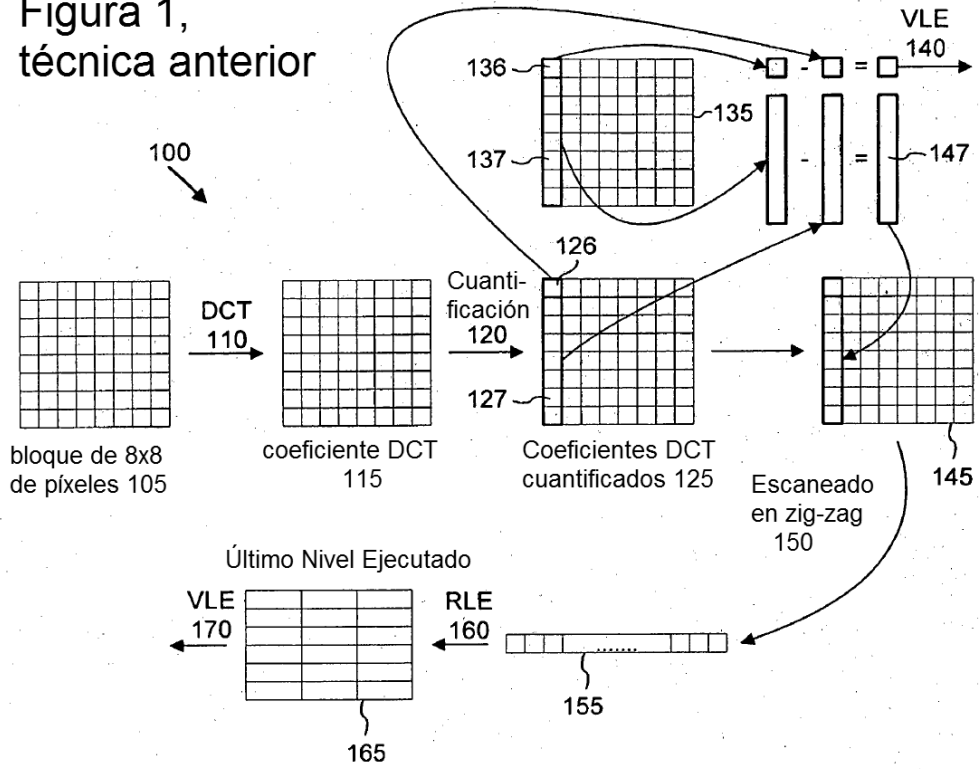


Figura 2, técnica anterior

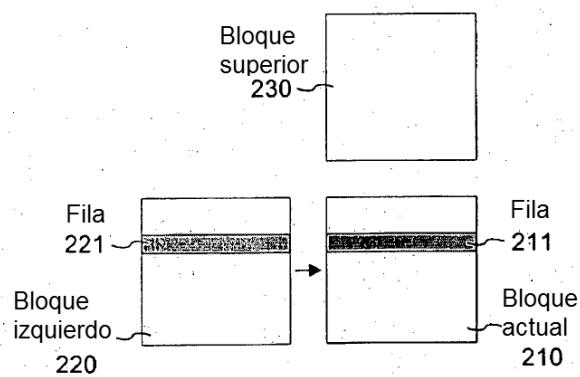


Figura 3, técnica anterior

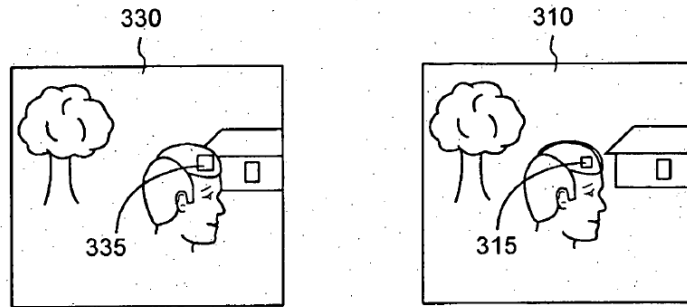
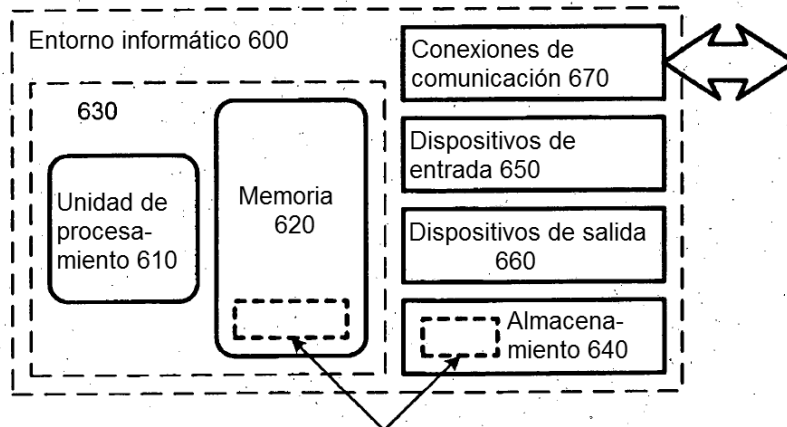


Figura 6



Software 680 que implementa codificación o descodificación de macrobloque omitido

Figura 4, técnica anterior

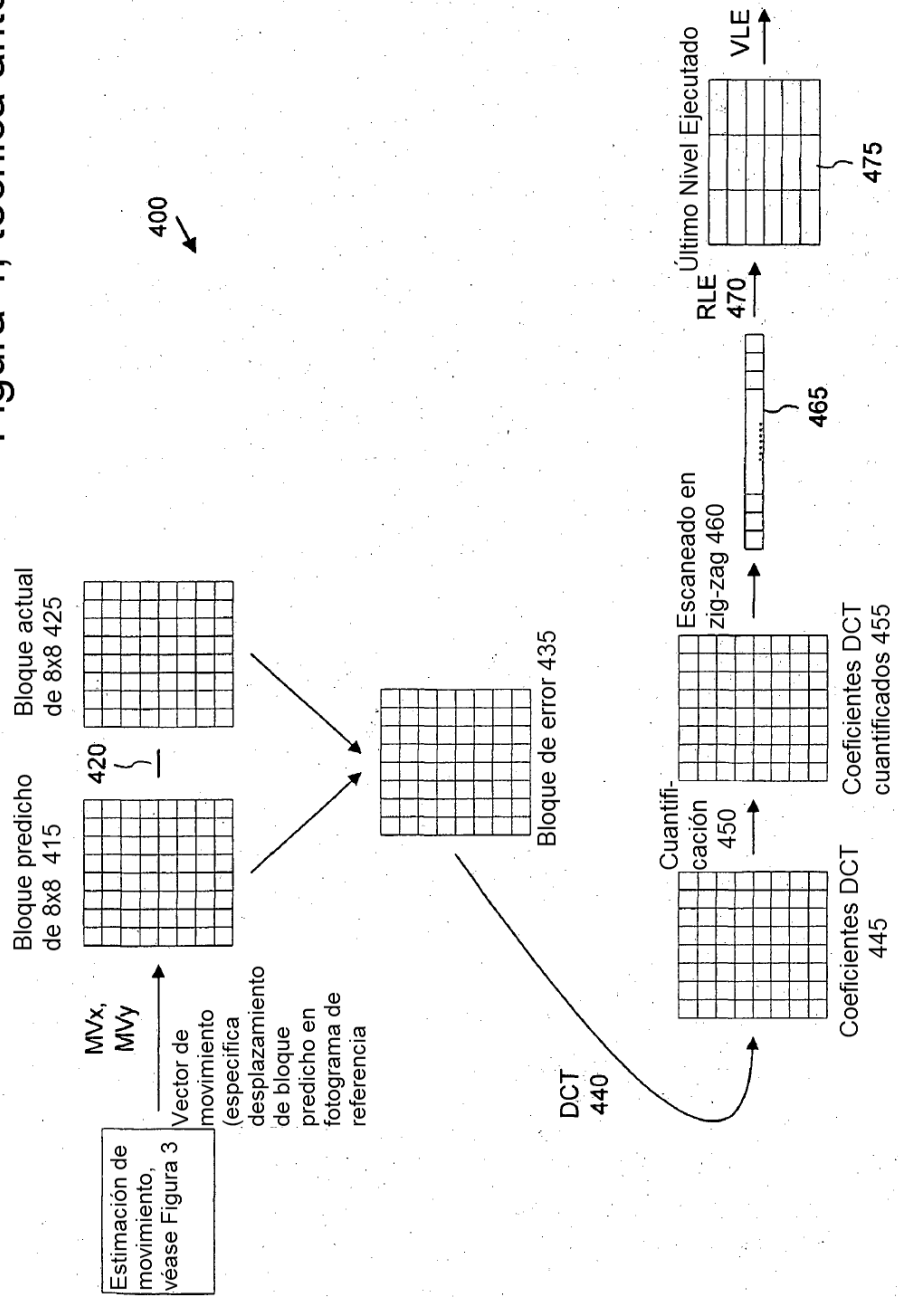


Figura 5, técnica anterior

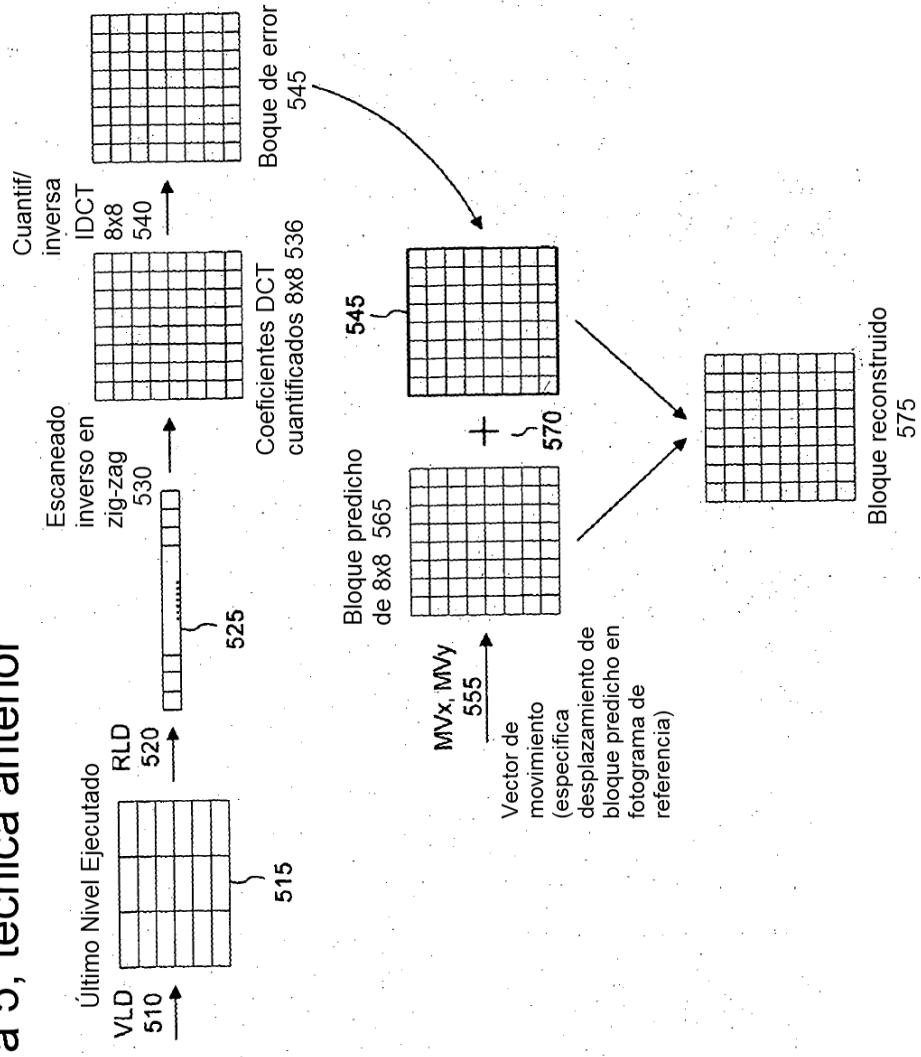


Figura 7

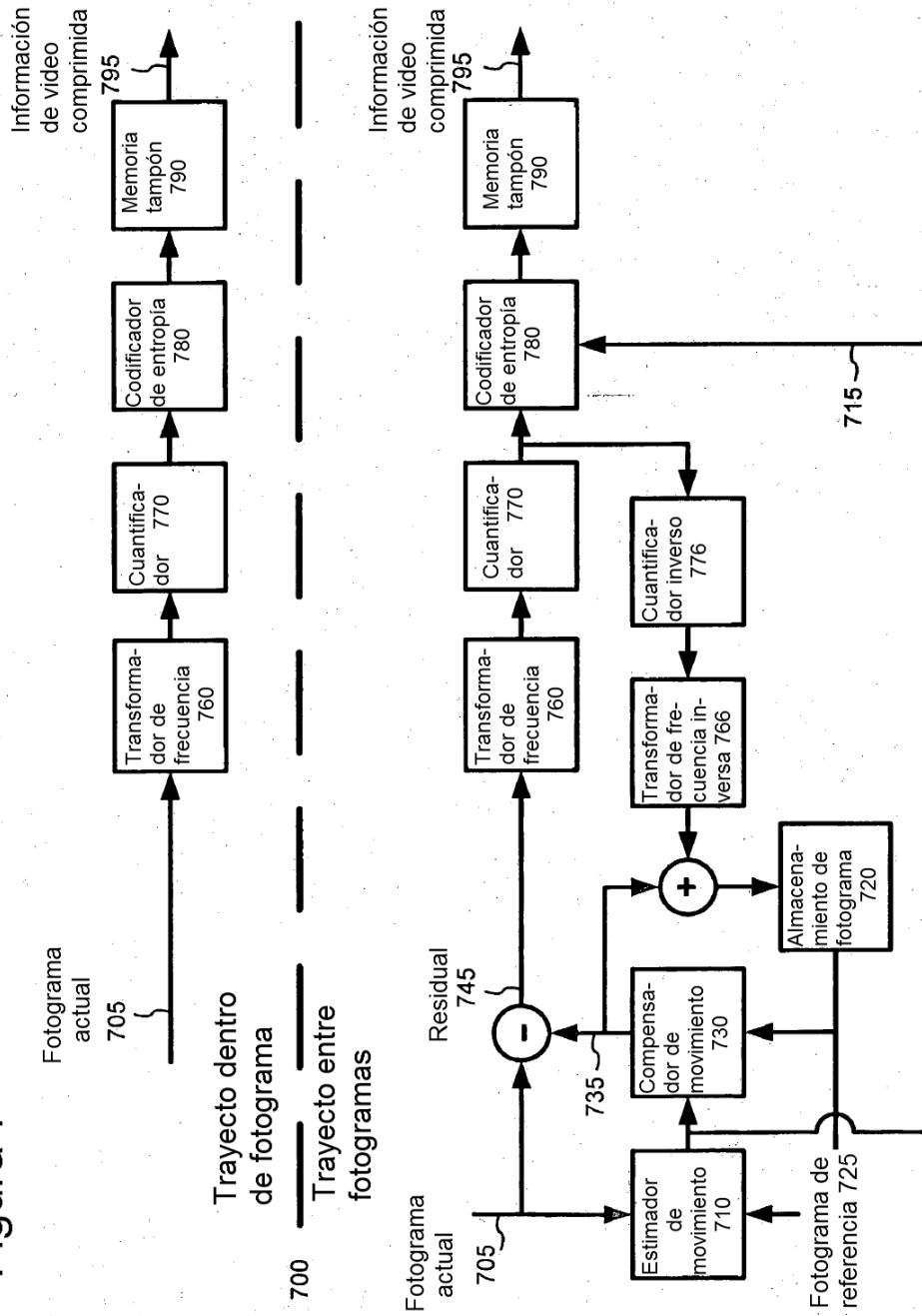


Figura 8

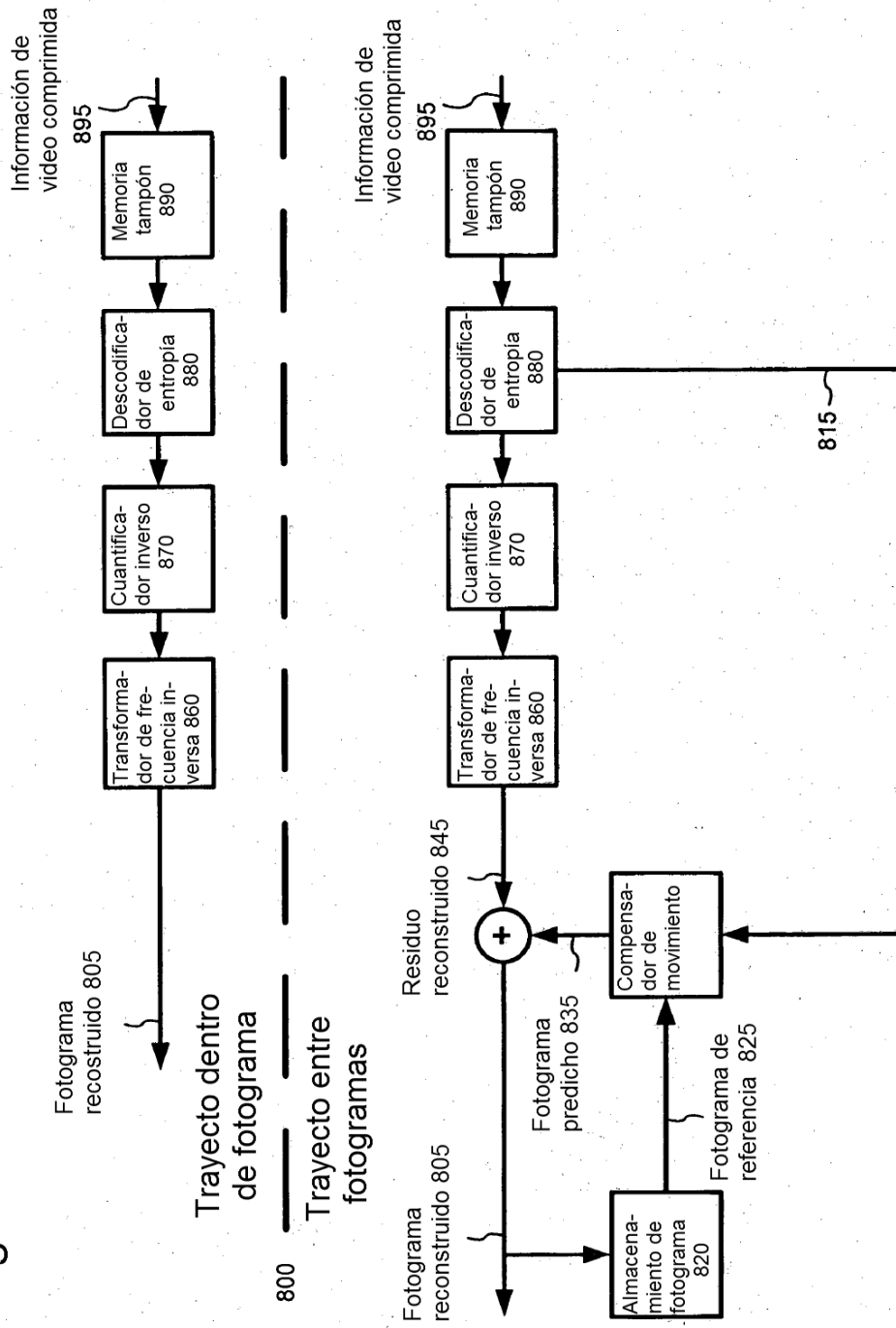


Figura 9

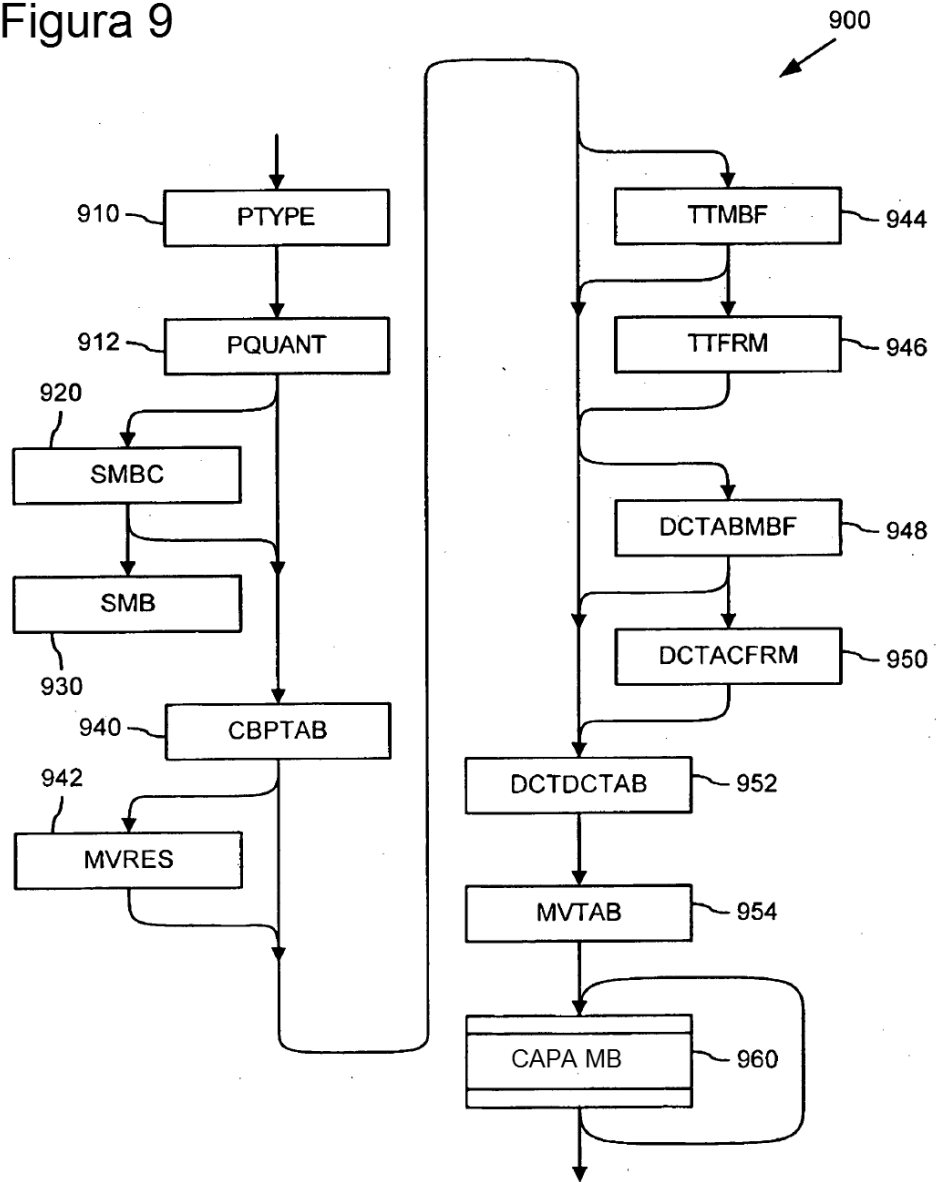


Figura 10

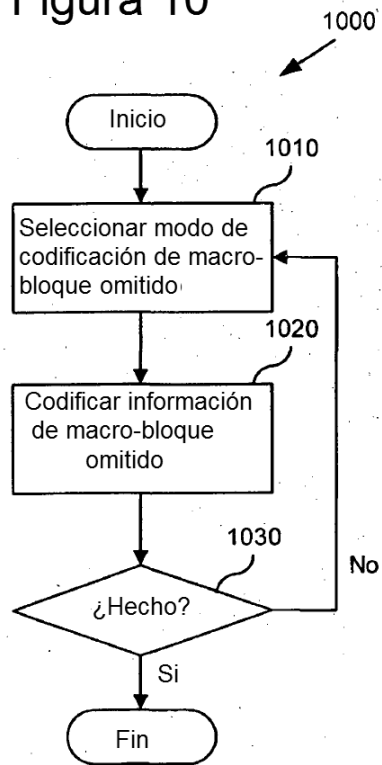


Figura 11

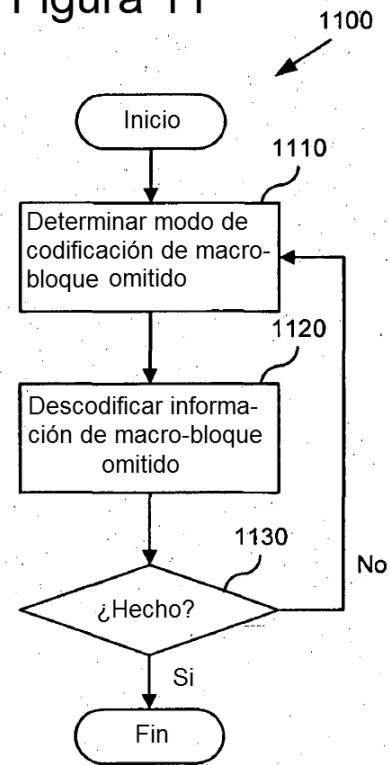


Figura 12

N	S	N	N	S	N
S	S	S	S	S	S
S	S	S	S	S	S
N	S	N	N	S	N

Fotograma ejemplar

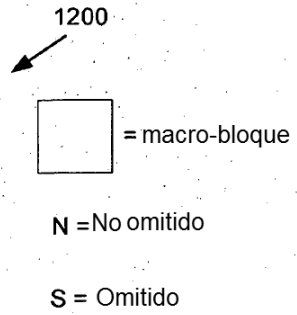


Figura 13

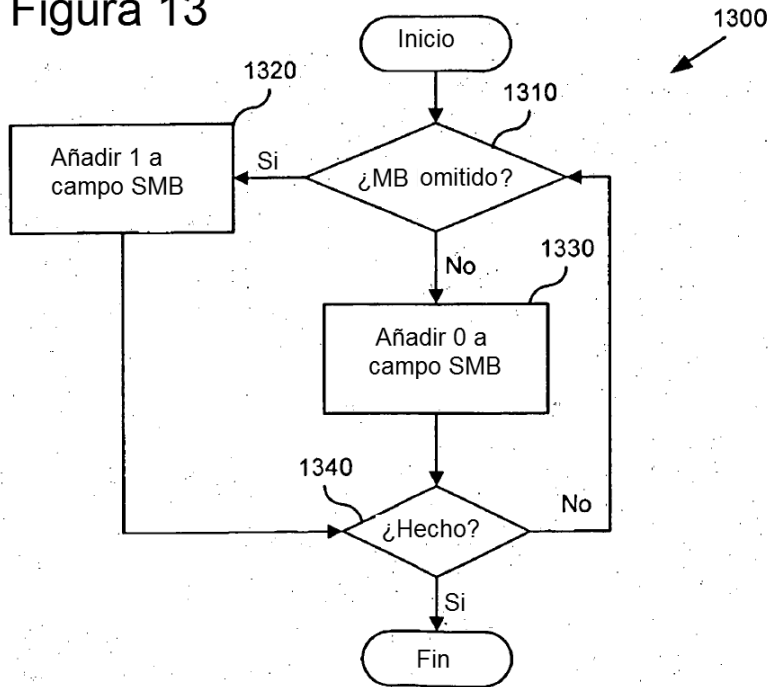


Figura 14

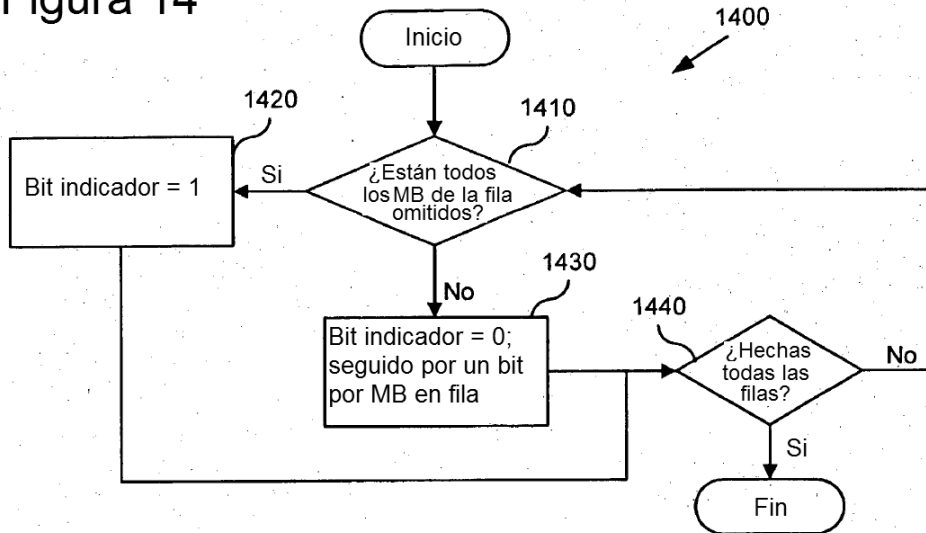


Figura 15

```

    for (mbrow = 0; mbrow < NumMBRows; mbrow++)
    {
        if (get_bits(1) == 1)
            ## Todos los macro-bloques en esta fila de macrobloques son omitidos
        si no
        {
            ## Al menos un macro-bloque en esta fila no es omitido
            ## Obtener estatus codificado de cada macro-bloque en la fila comenzando de izquierda a derecha
            for (mb = 0; mb < NumMBsPerRow; mb++)
                k = get_bits(1); ## k igual estatus de macro-bloque
                ## k == 0 si no está borrado, k == 1 si es omitido
        }
    }
  
```

Figura 16

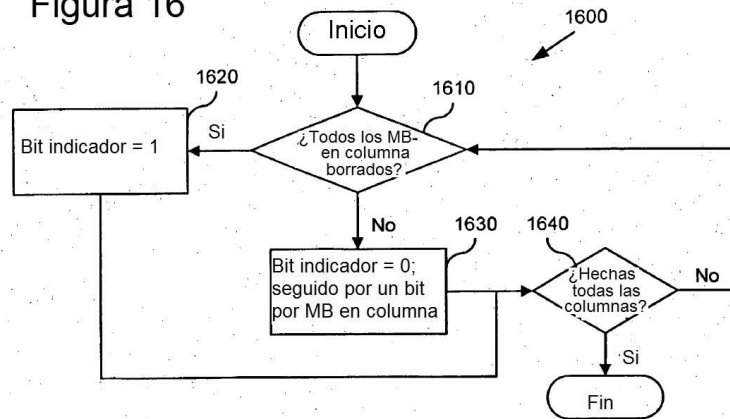


Figura 17

```

1700
for (mbcol = 0; mbcol < NumMBColumns; mbcol++)
{
    if (get_bits(1) == 1)
        ## Todos los macro-bloques en esta columna de macrobloques son omitidos
    si no
    {
        ## Al menos un macro-bloque en esta columna no es omitido
        ## Obtener estatus codificado de cada macro-bloque en columna comenzando desde
        parte superior a inferior
        for (mb = 0; mb < NumMBsPerColumn; mb++)
            k = get_bits(1); ## k igual a estatus de macro-bloque
            ## k == 0 si no está borrado, k == 1 si es omitido
    }
}
    
```

Figura 18

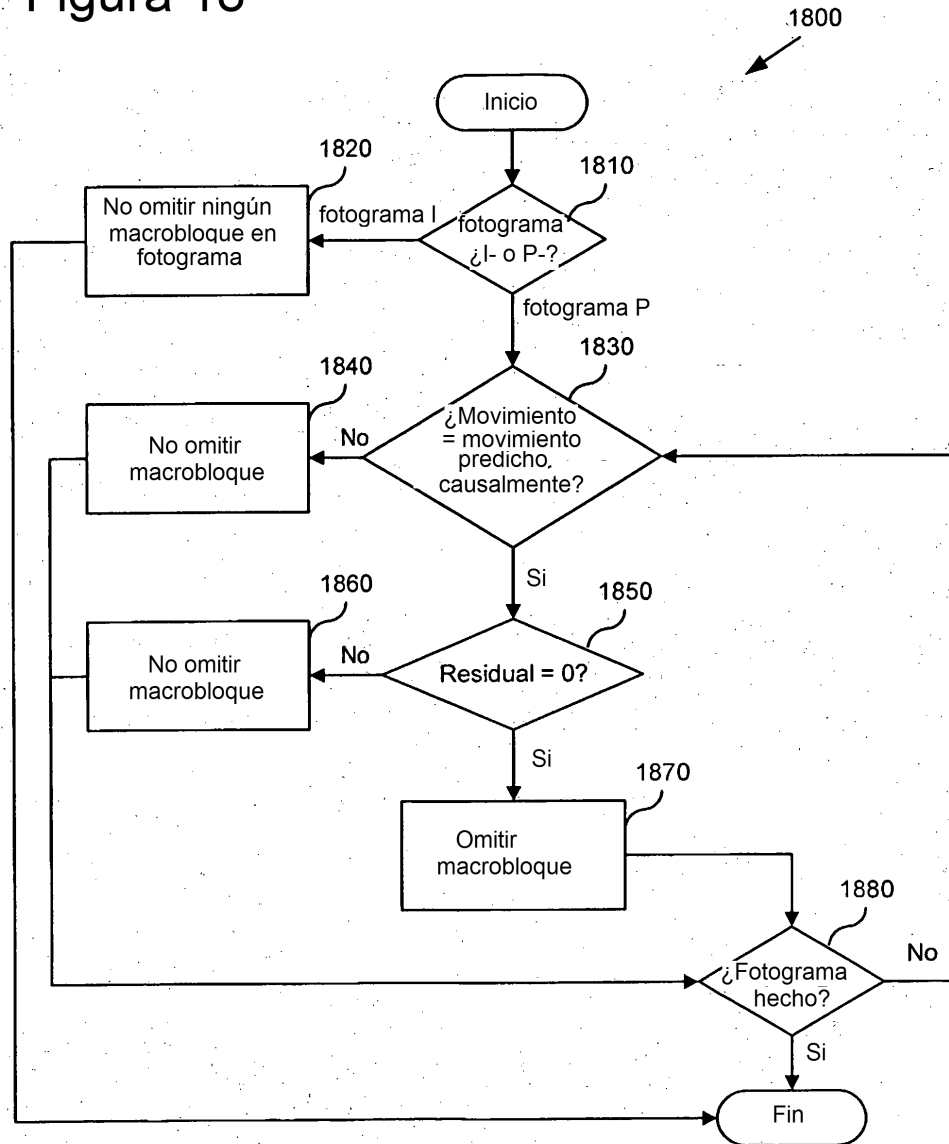


Figura 19

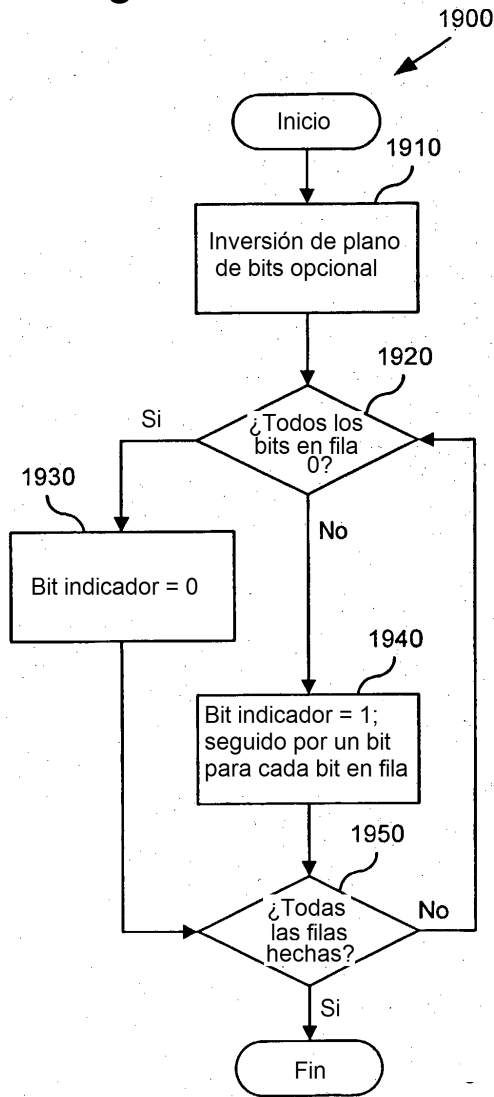


Figura 20

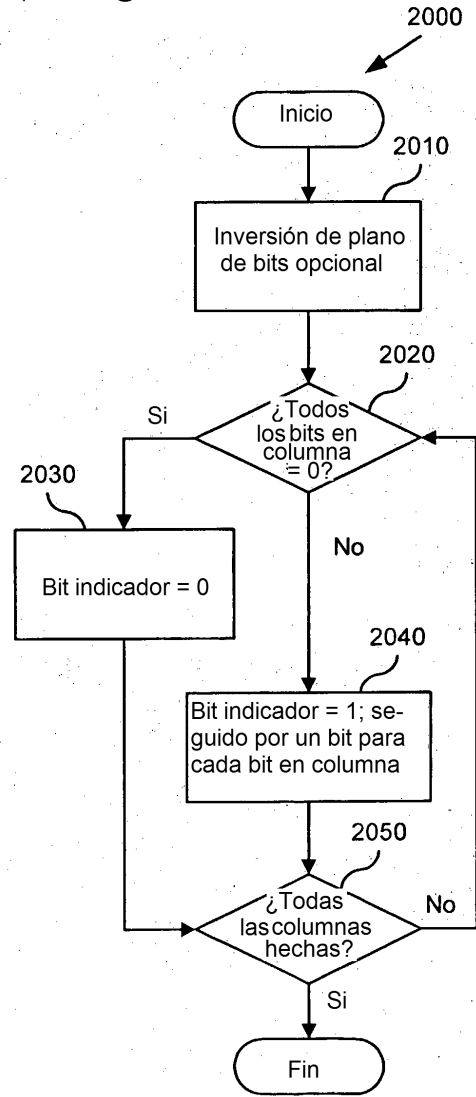


Figura 21

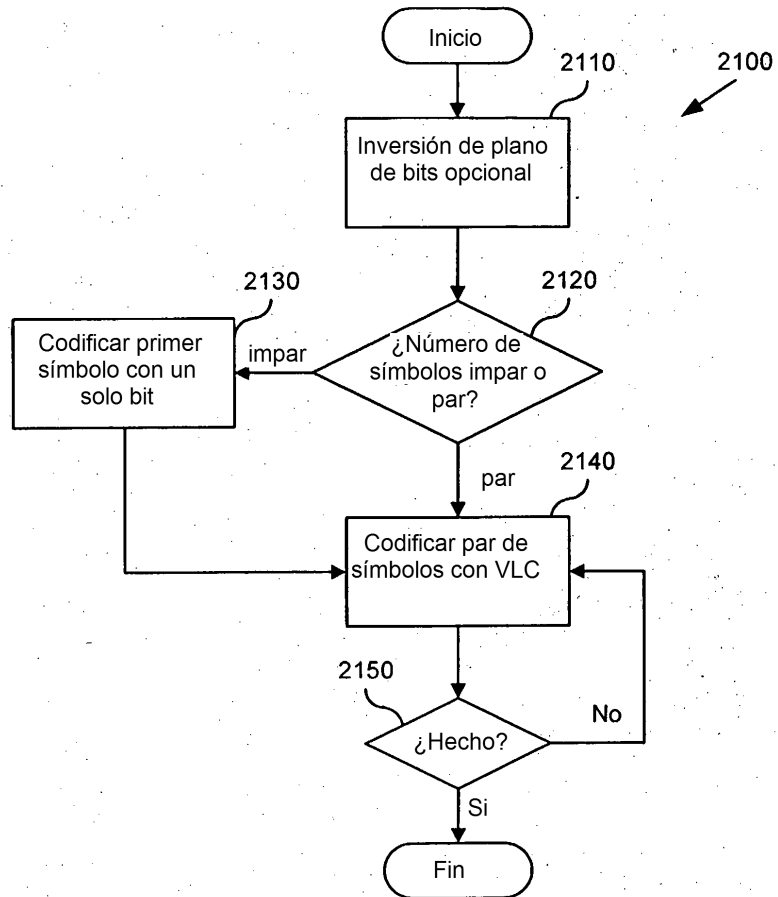


Figura 22

2200

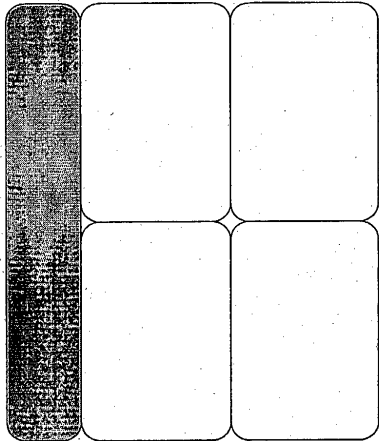


Figura 23

2300

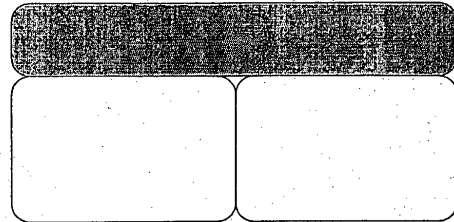


Figura 24

2400

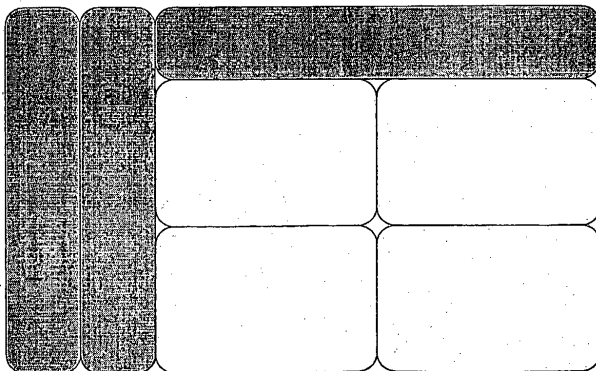


Figura 25

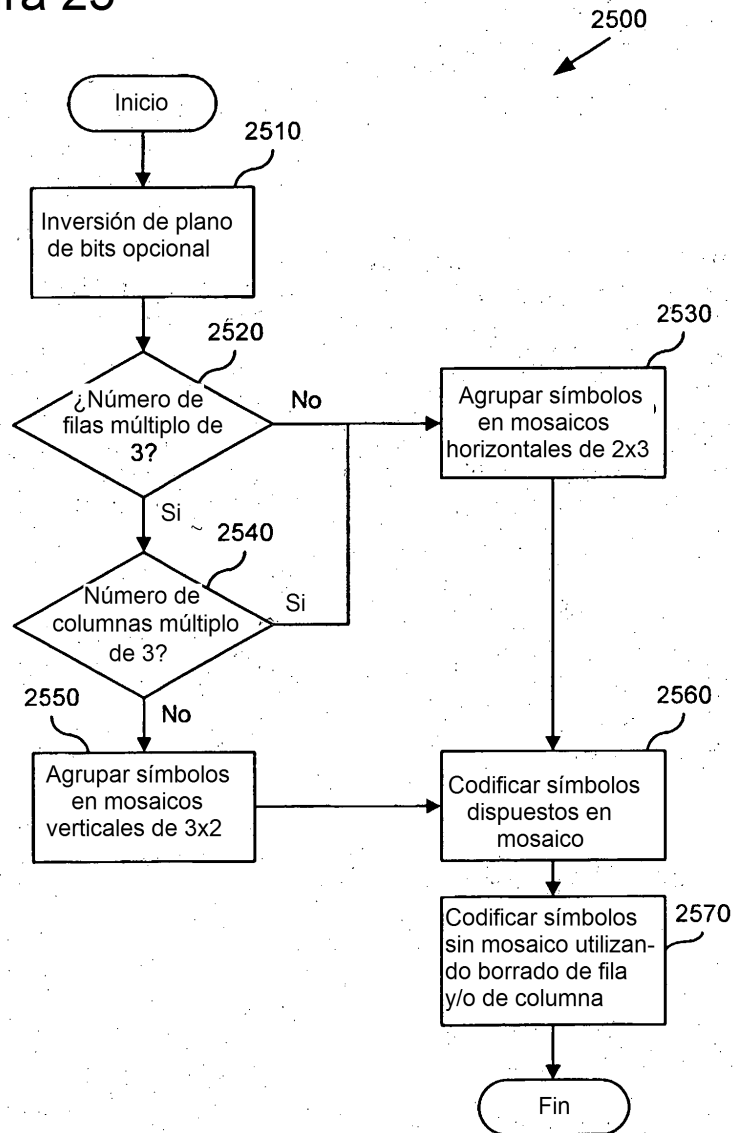


Figura 26

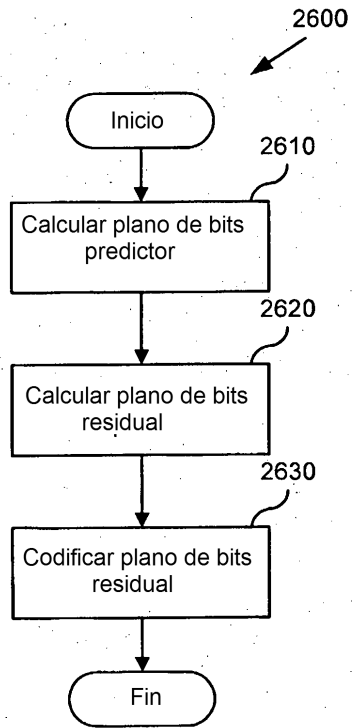


Figura 27

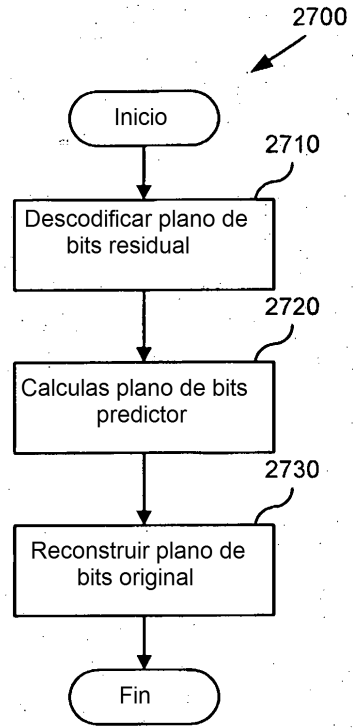


Figura 28

