(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 627 907**

(51) Int. CI.:

H04N 19/11	(2014.01)	***H04N 19/587***	(2014.01)
H04N 19/12	(2014.01)	***H04N 19/593***	(2014.01)
H04N 19/14	(2014.01)		
H04N 19/15	(2014.01)		
H04N 19/46	(2014.01)		
H04N 19/61	(2014.01)		
H04N 19/86	(2014.01)		
H04N 19/124	(2014.01)		
H04N 19/159	(2014.01)		
H04N 19/176	(2014.01)		

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **21.12.2011 PCT/US2011/066359**

(87) Fecha y número de publicación internacional: **28.06.2012 WO12088211**

(96) Fecha de presentación y número de la solicitud europea: **21.12.2011 E 11851099 (9)**

(97) Fecha y número de publicación de la concesión europea: **24.05.2017 EP 2656507**

(54) Título: **Codificación de intra-predicción mejorada usando representaciones planas**

(30) Prioridad:

21.12.2010 US 201061425670 P
04.03.2011 US 201161449528 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
01.08.2017

(73) Titular/es:

NTT DOCOMO, INC. (100.0%)
Sanno Park Tower, 11-1, Nagatacho 2-chome, Chiyoda-ku
Tokyo 100-6150, JP

(72) Inventor/es:

BOSSEN, FRANK, JAN y
KANUMURI, SANDEEP

(74) Agente/Representante:

FÚSTER OLAGUIBEL, Gustavo Nicolás

DESCRIPCIÓN

Codificación de intra-predicción mejorada usando representaciones planas

Solicitudes relacionadas

El presente documento de patente reivindica el beneficio de la fecha de presentación en 35 U.S.C. §119(e) de las solicitudes de patente estadounidenses provisionales n.^{os} 61/425 670, presentada el 21 de diciembre de 2010 y 61/449 528 presentada el 4 de marzo de 2011, cuyos contenidos al completo se incorporan al presente documento como referencia.

Antecedentes de la invención

1. Campo de la invención

La presente invención se refiere a una codificación de vídeo y en particular a predicción intra-trama mejorada con codificación de modo de predicción plana de baja complejidad.

2. Descripción de la técnica relacionada

El vídeo digital requiere una gran cantidad de datos para representar todas y cada una de las tramas de una secuencia de vídeo digital (por ejemplo, serie de tramas) de una manera sin comprimir. Para la mayoría de las aplicaciones no resulta viable transmitir vídeo digital sin comprimir a través de redes informáticas debido a limitaciones del ancho de banda. Además, el vídeo digital sin comprimir requiere una gran cantidad de espacio de almacenamiento. Normalmente, el vídeo digital se codifica de manera que se reduzcan los requisitos de almacenamiento y se reduzcan los requisitos de ancho de banda.

Una técnica para codificar vídeo digital es la predicción inter-trama, o inter-predicción. La inter-predicción aprovecha redundancias temporales entre diferentes tramas. Las tramas de vídeo temporalmente adyacentes incluyen normalmente bloques de píxeles, que permanecen sustancialmente iguales. Durante el procedimiento de codificación, un vector de movimiento interrelaciona el movimiento de un bloque de píxeles en una trama con un bloque de píxeles similares en otra trama. Por consiguiente, no se requiere que el sistema codifique el bloque de píxeles dos veces, sino que en vez de eso codifica el bloque de píxeles una vez y proporciona un vector de movimiento para predecir el otro bloque de píxeles.

Otra técnica para codificar vídeo digital es la predicción intra-trama o intra-predicción. La intra-predicción codifica una trama o una parte de la misma sin referencia a píxeles en otras tramas. La intra-predicción aprovecha redundancias espaciales entre bloques de píxeles dentro de una trama. Dado que bloques de píxeles espacialmente adyacentes tienen generalmente atributos similares, la eficacia del procedimiento de codificación se mejora haciendo referencia a la correlación espacial entre bloques adyacentes. Esta correlación puede aprovecharse mediante predicción de un bloque objetivo basándose en modos de predicción usados en bloques adyacentes.

Normalmente, un codificador comprende un predictor de píxeles, que comprende un inter-predictor, un intra-predictor y un selector de modo. El inter-predictor realiza la predicción para una imagen recibida, basándose en una trama de referencia de movimiento compensado. El intra-predictor realiza la predicción para la imagen recibida basándose en partes ya procesadas de la imagen o trama actual. El intra-predictor comprende además una pluralidad de modos de intra-predicción diferentes y realiza la predicción en los modos de predicción respectivos. Las salidas del inter-predictor y el intra-predictor se suministran al selector de modo.

El selector de modo determina qué método de codificación va a usarse, la codificación inter-predicción o la codificación intra-predicción, y, cuando va a usarse la codificación intra-predicción, determina qué modo de la codificación intra-predicción va a usarse de entre la pluralidad de modos de intra-predicción. En el proceso de determinación, el selector de modo usa funciones de coste para analizar qué método de codificación o qué modo proporciona el resultado más eficiente con respecto a costes de procesamiento y eficiencia de codificación.

Los modos de intra-predicción comprenden un modo DC y modos direccionales. El modo DC representa adecuadamente un bloque cuyos valores de píxel son constantes a través del bloque. Los modos direccionales son adecuados para representar un bloque que tiene un diseño a rayas en una dirección determinada. Existe otro diseño de imagen en el que la imagen se suaviza y sus valores de píxel cambian gradualmente en un bloque. El modo DC y los modos direccionales no son adecuados para predecir pequeños cambios graduales en el contenido de imagen y pueden crear artefactos de bloqueo molestos especialmente a tasas de bits de bajas a medias. Esto es debido a que cuando se codifican bloques con valores de píxel que cambian gradualmente, los coeficientes de AC de los bloques tienden a cuantificarse a cero, mientras que los coeficientes DC tienen valores no nulos.

Con el fin de hacer frente a este problema, los modos de intra-predicción en la norma H.264/AVC incluyen de

manera adicional un modo plano para representar un bloque con una imagen suavizada cuyos valores de píxel cambian gradualmente con un gradiente plano pequeño. En el modo plano de la norma H.264/AVC, se estima y señala un gradiente plano en un flujo de bits a un descodificador.

Sumario de la invención

La presente invención proporciona una codificación de modo plano de baja complejidad que puede mejorar la eficiencia de codificación de la codificación de intra-predicción. La presente invención se define mediante las reivindicaciones adjuntas.

En un aspecto de la presente invención, el codificador señala un residuo entre el bloque de predicción y un bloque objetivo en un flujo de bits a un descodificador.

En otro aspecto de la presente invención, un conjunto primario de núcleo de transformación H^N (i.j) se conmuta a un conjunto secundario de núcleo de transformación G^N (i.,j). El codificador transforma el residuo, usando el conjunto secundario de núcleo de transformación G^N (i.,j).

El conjunto secundario de núcleo de transformación G^N (i.,j) puede definirse por una de las siguientes ecuaciones:

a) $G^N(i,j) = k_i \times \text{sen}\left(\frac{(2i-1)j\pi}{2N+1}\right)$

(b) $G_F^N(i,j) = k_i \times \text{sen}\left(\frac{(2i-1)(2j-1)\pi}{4N}\right), \forall 1 \le i,j \le N$; y

(c) $G^N(i,j) = k_i \times \cos\left(\frac{(i-1)(2j-1)\pi}{2N}\right)$.

En otro aspecto de la presente invención, el conjunto secundario de núcleo de transformación G^N (i.,j) para un tamaño NxN se define mediante el conjunto primario de núcleo de transformación H^M (i.j) para un tamaño MxM, donde M > N. Específicamente, el conjunto secundario de núcleo de transformación G^N (i., j) puede definirse mediante

G^N (i, j) = k_i x H^{2N} (2i, N+ 1 - j), si se soportan los núcleos de transformación de tamaño 2Nx2N (H^{2N}), o

G^N (i,j) = H^N (i,j) de otro modo.

La presente invención también proporciona codificación de modo plano de baja complejidad usada para descodificar. En el modo plano, un descodificador calcula un primer valor de predicción y un segundo valor de predicción. El primer valor de predicción se calcula usando interpolación lineal entre un valor de píxeles de límite horizontal respectivos y un valor de uno de los píxeles de límite vertical. El segundo valor de predicción se calcula usando interpolación lineal entre un valor de píxeles de límite vertical respectivos y un valor de uno de los píxeles de límite horizontal. El descodificador realiza entonces el promedio del primer y segundo valor de predicción para obtener un valor de pixel de predicción respectivo en un bloque de predicción. El descodificador descodifica un residuo señalado desde el codificador que se generó en el modo plano en el codificador y añade el residuo descodificado al bloque de predicción para reconstruir datos de imagen.

Breve descripción de los dibujos

La figura 1 es un diagrama de bloques que muestra una arquitectura de hardware a modo de ejemplo en la que puede implementarse la presente invención.

La figura 2 es un diagrama de bloques que muestra una vista general de un codificador de vídeo al que puede aplicarse la presente invención.

La figura 3 es un diagrama de bloques que muestra una vista general de un descodificador de vídeo al que puede aplicarse la presente invención.

La figura 4 es un diagrama de bloques que muestra los módulos funcionales de un codificador según una realización de la presente invención.

La figura 5 es un diagrama de flujo que muestra un proceso de codificación realizado por el codificador de vídeo según una realización de la presente invención.

La figura 6 es un diagrama de bloques que muestra los módulos funcionales de un descodificador según una realización de la presente invención.

La figura 7 es un diagrama que muestra un proceso de descodificación realizado por el descodificador de vídeo según una realización de la presente invención.

La figura 8 es una representación esquemática de un bloque objetivo que contiene píxeles de 8x8 P(i, j) y píxeles de referencia usados para predecir los píxeles P(i, j).

La figura 9 es una representación esquemática que muestra el proceso de generación de píxeles de predicción según la codificación de modo plano propuesta en el documento JCT-VC A119.

La figura 10 es una representación esquemática que muestra el proceso de generación de píxeles de predicción según la codificación de modo plano de la presente invención.

La figura 11 es otra representación esquemática que muestra el proceso de generación de píxeles de predicción según la codificación de modo plano de la presente invención.

La figura 12 es un diagrama de flujo que muestra el proceso de conmutación entre un conjunto primario de núcleo de transformación y un conjunto secundario de núcleo de transformación.

Descripción detallada de los dibujos y las realizaciones preferidas en el presente documento

La figura 1 muestra una arquitectura de hardware a modo de ejemplo de un ordenador 100 en el que puede implementarse la presente invención. Obsérvese que la arquitectura de hardware mostrada en la figura 1 puede ser común tanto en un codificador de vídeo como en un descodificador de vídeo que implementan las realizaciones de la presente invención. El ordenador 100 incluye un procesador 101, memoria 102, dispositivo de almacenamiento 105 y uno o más dispositivos 106 de entrada y/o salida (I/O) (o periféricos) que están acoplados en comunicación a través de una interfaz local 107. La interfaz local 107 puede ser, por ejemplo, pero no se limita a, uno o más buses u otras conexiones por cable o inalámbricas, tal como se conoce en la técnica.

El procesador 101 es un dispositivo de hardware para ejecutar software, particularmente el almacenado en la memoria 102. El procesador 101 puede ser cualquier procesador fabricado a medida o comercialmente disponible, una unidad de procesamiento central (CPU), un procesador auxiliar entre varios procesadores asociados con el ordenador 100, un microprocesador basado en semiconductor (en forma de un microchip o conjunto de chips) o generalmente cualquier dispositivo para ejecutar instrucciones de software.

La memoria 102 comprende un medio legible por ordenador que puede incluir cualquiera de o una combinación de elementos de memoria volátil (por ejemplo, memoria de acceso aleatorio (RAM, tal como DRAM, SRAM, SDRAM, etc.)) y elementos de memoria no volátil (por ejemplo, ROM, disco duro, cinta, CD-ROM, etc.). Además, la memoria 102 puede incorporar medios de almacenamiento electrónicos, magnéticos, ópticos y/o de otros tipos. Un medio legible por ordenador puede ser cualquier medio que puede almacenar, comunicar, propagar o transportar el programa para su uso por o en conexión con el sistema, aparato o dispositivo de ejecución de instrucciones. Obsérvese que la memoria 102 puede tener una arquitectura distribuida, en la que diversos componentes están situados alejados unos de otros, pero a los que puede acceder el procesador 101.

El software 103 en la memoria 102 puede incluir uno o más programas separados, cada uno de los cuales contiene una lista ordenada de instrucciones ejecutable para implementar funciones lógicas del ordenador 100, tal como se describe a continuación. En el ejemplo de la figura 1, el software 103 en la memoria 102 define la funcionalidad de codificación de vídeo o descodificación de vídeo del ordenador 100 según la presente invención. Además, aunque no se requiere, es posible que la memoria 102 contenga un sistema operativo 104 (O/S). El sistema operativo 104 controla esencialmente la ejecución de programas informáticos y proporciona planificación, control de entrada-salida, gestión de archivos y datos, gestión de memoria y control de comunicación y servicios relacionados.

El dispositivo de almacenamiento 105 del ordenador 100 puede ser uno de muchos tipos diferentes de dispositivo de almacenamiento, incluyendo un dispositivo de almacenamiento estacionario o dispositivo de almacenamiento portátil. Como ejemplo, el dispositivo de almacenamiento 105 puede ser una cinta magnética, disco, memoria flash, memoria volátil o un dispositivo de almacenamiento diferente. Además, el dispositivo de almacenamiento 105 puede ser una tarjeta de memoria digital segura o cualquier otro dispositivo de almacenamiento 105 extraíble.

Los dispositivos de I/O 106 pueden incluir dispositivos de entrada, por ejemplo, pero sin limitarse a, una pantalla táctil, un teclado, ratón, escáner, micrófono u otro dispositivo de entrada. Además, los dispositivos de I/O 106 también pueden incluir dispositivos de salida, por ejemplo, pero sin limitarse a, una pantalla u otros dispositivos de salida. Los dispositivos de I/O 106 pueden incluir además dispositivos que se comunican a través tanto de entradas como de salidas, por ejemplo, pero sin limitarse a, un modulador/desmodulador (módem; para acceder a otro dispositivo, sistema o red), un transceptor de radiofrecuencia (RF), inalámbrico u otro, una interfaz telefónica, un puente, un enrutador u otros dispositivos que funcionan como entrada y como salida.

Tal como conocen bien los expertos habituales en la técnica, la compresión de vídeo se logra eliminando información redundante en una secuencia de vídeo. Existen muchas normas diferentes de codificación de vídeo, entre los ejemplos de las cuales incluyen MPEG-1, MPEG-2, MPEG-4, H.261, H.263 y H.264/AVC. Debe observarse que la presente invención no pretende limitarse en cuanto a la aplicación de cualquier norma de codificación de vídeo específica. Sin embargo, la siguiente descripción de la presente invención se proporciona usando el ejemplo de la norma H.264/AVC, que se incorpora al presente documento como referencia. H.264/AVC es la norma de codificación de vídeo más reciente y logra una mejora de rendimiento significativa con respecto a las normas de codificación anteriores tales como MPEG-1, MPEG-2, H.261 y H.263.

En H.264/AVC, cada trama o imagen de un vídeo puede descomponerse en varios segmentos. Los segmentos se dividen entonces en bloques de 16X16 píxeles denominados macrobloques, que después pueden dividirse adicionalmente en bloques de 8X16, 16x8, 8X8, 4X8, 8x4, hasta 4X4 píxeles. Hay cinco tipos de segmentos soportados por H.264/AVC. En los segmentos I, todos los macrobloques se codifican usando intra-predicción. En los segmentos P, los macrobloques pueden codificarse usando intra o inter-predicción. Los segmentos P solo permiten usar una señal de predicción compensada por movimiento (MCP) por macrobloque. En los segmentos B, pueden codificarse macrobloques usando intra o inter-predicción. Pueden usarse dos señales de MCP por predicción. Los segmentos SP permiten conmutar segmentos P entre diferentes flujos de vídeo de manera eficaz. Un segmento SI es una coincidencia exacta para un segmento SP para acceso aleatorio o recuperación de error, mientras que solo se usa intra-predicción.

La figura 2 muestra una vista general de un codificador de vídeo al que puede aplicarse la presente invención. Los bloques mostrados en la figura representan módulos funcionales realizados por el procesador 101 que ejecuta el software 103 en la memoria 102. Se alimenta una imagen de trama de vídeo 200 a un codificador de vídeo 201. El codificador de vídeo trata la imagen 200 en unidades de macrobloques 200A. Cada macrobloque contiene varios píxeles de imagen 200. En cada macrobloque se realiza una transformación en coeficientes de transformación seguida por una cuantificación en niveles de coeficientes de transformación. Además, se usa intra-predicción o inter-predicción, para no realizar las etapas de codificación directamente en los datos de píxel sino en las diferencias de los mismos con respecto a valores de píxel predichos, logrando así valores pequeños que se comprimen más fácilmente.

Para cada segmento, el codificador 201 genera varios elementos de sintaxis, que forman una versión codificada de los macrobloques del segmento respectivo. Todos los elementos de datos residuales en los elementos de sintaxis, que están relacionados con la codificación de coeficientes de transformación, tales como los niveles de coeficientes de transformación o un mapa de significación que indica niveles de coeficientes de transformación omitidos, se denominan elementos de sintaxis de datos residuales. Además de estos elementos de sintaxis de datos residuales, los elementos de sintaxis generados por el codificador 201 contienen elementos de sintaxis de información de control que contienen información de control sobre cómo se ha codificado cada macrobloque y cómo tiene que descodificarse, respectivamente. En otras palabras, los elementos de sintaxis pueden dividirse en dos categorías. La primera categoría, los elementos de sintaxis de información de control, contiene los elementos relacionados con un tipo de macrobloque, tipo de sub-macrobloque e información sobre modos de predicción de tipos tanto espacial como temporal, así como información de control basada en segmento y basada en macrobloque, por ejemplo. En la segunda categoría, todos los elementos de datos residuales, tales como un mapa de significación que indica las ubicaciones de todos los coeficientes significativos dentro de un bloque de coeficientes de transformación cuantificados y los valores de los coeficientes significativos, que se indican en unidades de niveles correspondientes a las etapas de cuantificación, se combinan y se convierten en elementos de sintaxis de datos residuales.

El codificador 201 comprende un codificador de entropía que codifica elementos de sintaxis y genera contraseñas aritméticas para cada segmento. Cuando se generan las contraseñas aritméticas para un segmento, el codificador de entropía aprovecha dependencias estadísticas entre los valores de datos de elementos de sintaxis en el flujo de bits de la señal de vídeo. El codificador 201 emite una señal de vídeo codificada para un segmento de imagen 200 a un descodificador de vídeo 301 mostrado en la figura 3.

La figura 3 muestra una vista general de un descodificador de vídeo al que puede aplicarse la presente invención. Asimismo, los bloques mostrados en la figura representan módulos funcionales realizados por el procesador 101 que ejecuta el software 103 en la memoria 102. El descodificador de vídeo 301 recibe la señal de vídeo codificada y en primer lugar realiza la descodificación de entropía de la señal de vuelta a los elementos de sintaxis. El descodificador 301 usa los elementos de sintaxis para reconstruir, macrobloque por macrobloque y después segmento por segmento, las muestras 300A de imagen de píxeles en la imagen 300.

La figura 4 muestra los módulos funcionales del codificador de vídeo 201. Estos módulos funcionales se realizan mediante el procesador 101 que ejecuta el software 103 en la memoria 102. Una imagen de vídeo de entrada es una trama o un campo de una imagen de vídeo natural (sin comprimir) definida por puntos de muestra que representan componentes de colores originales, tales como crominancia (“croma”) y luminancia (“luma”) (otras componentes son posibles, por ejemplo, tono, saturación y valor). La imagen de vídeo de entrada se divide en macrobloques 400 que representan cada uno un área de imagen cuadrada que consiste en 16X16 píxeles de la componente luma del color de la imagen. La imagen de vídeo de entrada también se reparte en macrobloques que representan cada uno 8X8

píxeles de cada una de las dos componentes de croma del color de la imagen. En el funcionamiento de codificador general, los macrobloques introducidos pueden predecirse de manera temporal o espacial usando inter o intra-predicción. Sin embargo, con el propósito de discusión, se supone que los macrobloques 400 son todos macrobloques de tipo segmento I y se someten únicamente a intra-predicción.

La intra-predicción se logra en un módulo de intra-predicción 401, cuyo funcionamiento se comentará con detalle a continuación. El módulo de intra-predicción 401 genera un bloque de predicción 402 a partir de píxeles de límite horizontal y vertical de bloques adyacentes, que se han codificado, reconstruido y almacenado anteriormente en una memoria de trama 403. Un residuo 404 del bloque de predicción 402, que es la diferencia entre un bloque objetivo 400 y el bloque de predicción 402, se transforma por un módulo 405 y después se cuantifica usando un cuantificador 406. El módulo de transformación 405 transforma el residuo 404 en un bloque de coeficientes de transformación 407. Entonces se someten los coeficientes de transformación 407 cuantificados a codificación de entropía en un módulo de codificación de entropía 408 y se transmiten (junto con otra información relacionada con la intra-predicción) como una señal de vídeo codificada 409.

El codificador de vídeo 201 contiene funcionalidad de descodificación para realizar la intra-predicción en bloques objetivo. La funcionalidad de descodificación comprende un cuantificador 410 y un módulo de transformación inversa 411 que realiza la cuantificación inversa y la transformación inversa en los coeficientes de transformación 406 cuantificados para producir el residuo de predicción 410 descodificado, que se añade al bloque de predicción 402. La suma del residuo de predicción 410 descodificado y el bloque de predicción 402 es un bloque 413 reconstruido, que se almacena en la memoria de trama 403 y se leerá de la misma y será usado por el módulo de intra-predicción 401 para generar un bloque de predicción 402 para descodificar un siguiente bloque objetivo 400. Un filtro de desbloqueo puede colocarse de manera opcional o bien en la entrada o bien en la salida de la memoria de trama 403 para eliminar los artefactos de bloqueo de las imágenes reconstruidas.

La figura 5 es un diagrama de flujo que muestra procedimientos realizados por el codificador de vídeo 401. Según la norma H.264/AVC, la intra-predicción implica predecir cada píxel del bloque objetivo 400 en una pluralidad de modos de predicción, usando interpolaciones de píxeles de límite (“píxeles de referencia”) de bloques adyacentes anteriormente codificados y reconstruidos. Los modos de predicción se identifican mediante números enteros positivos 0, 1, 2..., cada uno asociado con una instrucción o un algoritmo diferente para predecir píxeles específicos en el bloque objetivo 400. El módulo de intra-predicción 401 ejecuta una intra-predicción en los modos de predicción respectivos y genera diferentes bloques de predicción. En un algoritmo de búsqueda completa (“FS”), cada uno de los bloques de predicción generados se compara con el bloque objetivo 400 para encontrar el modo de predicción óptimo, lo cual minimiza el residuo de predicción 404 o produce un residuo de predicción 404 menor entre los modos de predicción. La identificación del modo de predicción óptimo se comprime y se envía al descodificador 301 con otros elementos de sintaxis de información de control.

Cada modo de predicción puede describirse mediante una dirección general de predicción tal como se describe verbalmente (es decir, horizontal hacia arriba, vertical y diagonal hacia abajo y a la izquierda). Una dirección de predicción puede describirse gráficamente mediante una dirección angular. El ángulo correspondiente a un modo de predicción tiene una relación general con respecto a la dirección desde la ubicación promedio ponderada de los píxeles de referencia usados para predecir un píxel objetivo en la ubicación de píxel objetivo. En el modo de predicción DC, el bloque de predicción 402 se genera de tal manera que cada píxel en el bloque de predicción 402 se establece uniformemente al valor medio de los píxeles de referencia.

Volviendo a la figura 5, el módulo de intra-predicción 401 emite el bloque de predicción 402, que se resta del bloque objetivo 400 para obtener el residuo 404 (etapa 503). El módulo de transformación 405 transforma el residuo 404 en un bloque de coeficientes de transformación (etapa 504). El cuantificador 406 cuantifica los coeficientes de transformación en coeficientes de transformación cuantificados. El modo de codificación de entropía 408 realiza la codificación de entropía de los coeficientes de transformación cuantificados (etapa 506), que se envían junto con la identificación comprimida del modo de predicción óptimo. El cuantificador inverso 410 cuantifica de manera inversa los coeficientes de transformación cuantificados (etapa 507). El módulo de transformación inversa 411 realiza la transformación inversa para obtener el residuo de predicción descodificado 412 (etapa 508), que se añade con el bloque de predicción 402 para convertirse en el bloque 413 reconstruido (etapa 509).

La figura 6 muestra los módulos funcionales del descodificador de vídeo 301. Estos módulos funcionales se realizan mediante el procesador 101 ejecutando el software 103 en la memoria 102. La señal de vídeo codificada del codificador 201 es recibida en primer lugar por un descodificador de entropía 600 y se realiza la descodificación de entropía de vuelta a los coeficientes de transformación cuantificados 601. Los coeficientes de transformación cuantificados 601 se cuantifican de manera inversa mediante un cuantificador inverso 602 y se transforman de manera inversa mediante un módulo de transformación inversa 603 para generar un residuo de predicción 604. A un módulo de intra-predicción 605 se le notifica el modo de predicción seleccionado por el codificador 201. Según el modo de predicción seleccionado, el módulo de intra-predicción 605 realiza un proceso de intra-predicción similar al que se realizó en la etapa 503 de la figura 5 para generar un bloque de predicción 606, usando píxeles de límite de bloques adyacentes previamente reconstruidos y almacenados en una memoria de trama 607. El bloque de predicción 606 se añade al residuo de predicción 604 para reconstruir un bloque 608 de señal de vídeo

descodificada. El bloque 608 reconstruido se almacena en la memoria de trama 607 para su uso en la predicción de un próximo bloque.

La figura 7 es un diagrama de flujo que muestra los procesos realizados por el codificador de vídeo 201. El descodificador de vídeo 301 descodifica la identificación del modo de predicción óptimo señalado desde el codificador de vídeo 201 (etapa 701). Usando el modo de predicción descodificado, el módulo de intra-predicción 605 genera el bloque de predicción 606, usando píxeles de límite de bloques adyacentes previamente reconstruidos y almacenados en una memoria de trama 607 (etapa 702). El descodificador aritmético 600 descodifica la señal de vídeo codificada del codificador 201 de vuelta a los coeficientes de transformación cuantificados 601 (etapa 703). El cuantificador inverso 602 cuantifica de manera inversa los coeficientes de transformación cuantificados a los coeficientes de transformación (etapa 704). El módulo de transformación inversa 603 transforma de manera inversa los coeficientes de transformación en el residuo de predicción 604 (etapa 705), que se añade con el bloque de predicción 606 para reconstruir el bloque 608 de señal de vídeo descodificada (etapa 706).

El proceso de codificación realizado por el codificador de vídeo 201 puede explicarse adicionalmente con referencia a la figura 8. La figura 8 es una representación esquemática de un bloque objetivo que contiene píxeles de 8x8 P(i,j) y píxeles de referencia usados para predecir los píxeles P(i,j). En la figura 8, los píxeles de referencia consisten en 17 píxeles horizontales y 17 píxeles verticales, en la que el píxel de la parte superior izquierda es común tanto para límites horizontales como verticales. Por tanto, 32 píxeles diferentes están disponibles para generar píxeles de predicción para el bloque objetivo. Obsérvese que aunque la figura 8 muestra un bloque de 8x8 que va a predecirse, la siguiente explicación se generaliza para convertirse en aplicable para diversos números de píxeles en configuraciones diferentes. Por ejemplo, un bloque que va a predecirse puede comprender una matriz 4x4 de píxeles. Un bloque de predicción también puede comprender una matriz 8 x 8 de píxeles, una matriz 16 x 16 de píxeles, o matrices de píxeles más grandes. Otras configuraciones píxel, incluyendo tanto matrices cuadradas como rectangulares, también pueden constituir un bloque de predicción.

Suponiendo que un bloque de píxeles ({ P(i,j) : 1 ≤ i,j ≤ N}) experimente la codificación intra-predicción que usa píxeles de referencia horizontales y verticales $(\{P(i,0):0\le i\le 2N\}\cup\{P(0,j):0\le j\le 2N\})$. Donde P_o (i,j) indica los valores de píxel originales del bloque objetivo, PP (i,j) indica los valores de píxel predichos, P_R(i,j) indica los valores de residuo, P_Q(i,j) indica los valores de residuo comprimidos y PC (i,j) indica los valores comprimidos para los píxeles P(i,j), las siguientes ecuaciones definen su relación:

$$P_R(i,j)=P_O(i,j)-P_P(i,j),\forall 1\le i,j\le N$$

$$P_T(1:N,1:N)=Q_F(H_F^N*P_R(1:N,1:N)*(H_F^N)^T)$$

$$P_Q(1:N,1:N)=H_I^N*Q_I(P_T(1:N,1:N))*(H_I^N)^T$$

$$P_C(i,j)=P_Q(i,j)+P_P(i,j),\forall 1\le i,j\le N$$

H_F^N, es una matriz N x N que representa el núcleo de transformación hacia delante. H_I^N es una matriz N x N que representa el núcleo de transformación inversa. P_T (1 : N, 1: N) representa las señales de residuo cuantificadas y transformadas en un flujo de bits. Q_F ( ) representa la operación cuantificación y Q_I ( ) representa la operación cuantificación inversa.

Los valores de píxel predichos P_P (i,j) se determinan mediante un modo de intra-predicción realizado con los píxeles de referencia $\{P(i,0):0\le i\le 2N\}\cup\{P(0,j):0\le j\le 2N\}$. H.264/ AVC soporta predicción Intra_4x4, predicción Intra_8x8 y predicción Intra_16x16. La predicción Intra_4x4 se realiza en nueve modos de predicción, incluyendo un modo de predicción vertical, un modo de predicción horizontal, un modo de predicción DC y seis modos de predicción angular. La predicción Intra_8x8 se realiza en los nueve modos de predicción tal como se realiza en predicción Intra_4x4. La predicción Intra_16x 16 se realiza en cuatro modos de predicción, incluyendo un modo de predicción vertical, un modo de predicción horizontal, un modo de predicción DC y un modo de predicción plana. Por ejemplo, los valores de píxel predichos P_P (i.j) obtenidos en el modo de predicción DC, el modo de predicción vertical y el modo de predicción horizontal están definidos tal como sigue:

Modo de predicción DC:

$$P_P(i,j)=\frac{\sum_{k=1}^{N}P_C(k,0)+P_C(0,k)}{2N},\forall 1\le i,j\le N$$

Modo de predicción vertical:

$$P_P(i,j) = P_C(0,j), \forall 1 \le i,j \le N$$

Modo de predicción horizontal:

$$P_P(i,j) = P_C(i,0), \forall 1 \le i,j \le N$$

Recientemente, la propuesta n.º JCT-VC A119 se presentó al Joint Collaborative Team on Video Coding (JCT-VC), la cual se incorpora en el presente documento como referencia. La propuesta n.º JCT-VC A119 propone una operación de modo plano de baja complejidad que usa una combinación de operaciones de interpolación lineal y bilineal para predecir valores de píxel que cambian gradualmente con un gradiente plano pequeño. El proceso de modo plano propuesto se muestra esquemáticamente en la figura 9. El proceso comienza con la identificación del valor P_P (N,N) del píxel de la parte inferior derecha en un bloque que va a predecirse. Entonces, se realizan interpolaciones lineales entre el valor P_P (N,N) y el valor de píxel de referencia P_C (N,0) para obtener valores de píxel predichos P_P (N,j) de la fila inferior en el bloque. Del mismo modo, se realizan interpolaciones lineales entre el valor P_P (N,N) y el valor de píxel de referencia P_C (0,N) para obtener valores de píxel predichos P_P (i,N) de la columna más a la derecha en el bloque. Tras esto, se realizan interpolaciones bilineales de entre los valores de píxel predichos P_P (N,j) y P_P (i,N) y los valores de píxel de referencia P_C (i,0) y P_C (0,j) para obtener el resto de los valores de píxel P_P (i,j) en el bloque. El proceso de modo plano propuesto puede expresarse mediante las siguientes ecuaciones:

Columna derecha:

$$P_P(i,N) = \frac{(N-i) \times P_C(0,N) + i \times P_P(N,N)}{N}, \forall 1 \le i \le (N-1)$$

Fila inferior:

$$P_P(N,j) = \frac{(N-j) \times P_C(N,0) + j \times P_P(N,N)}{N}, \forall 1 \le j \le (N-1)$$

Resto de los píxeles:

$$P_P(i,j) = \frac{(N-i) \times P_C(0,j) + i \times P_P(N,j) + (N-j) \times P_C(i,0) + j \times P_P(i,N)}{2N}, \forall 1 \le i,j \le (N-1)$$

Existen dos problemas a resolver que pueden encontrarse en el proceso de modo plano propuesto en el documento JCT-VC A119. En el proceso propuesto, el valor P_P (N,N) del píxel de la parte inferior derecha está señalado en un flujo de bits al descodificador y usado para descodificar el bloque objetivo en el descodificador. En otras palabras, el descodificador necesita el valor del píxel de la parte inferior derecha para realizar la predicción en el modo plano propuesto. Además, en el proceso propuesto, el residuo no se obtiene en el modo plano y, por tanto, no se señala al descodificador. La omisión de la señalización residual puede contribuir a la reducción de datos de vídeo codificados que van a transmitirse, pero limita la aplicación del modo plano a codificación de vídeo de baja tasa de bits.

El modo plano según la presente invención está diseñado para resolver los problemas anteriormente mencionados asociados al proceso de modo plano propuesto en el documento JCT-VC A119. Según una realización de la presente invención, el valor P_P (N,N) del píxel de la parte inferior derecha se obtiene a partir de los píxeles de referencia. Por tanto, no es necesario señalar el valor de píxel P_P (N,N) del píxel de la parte inferior derecha al descodificador. En otra realización de la presente invención, el bloque de predicción formado en el modo plano se usa para obtener un residuo, que se transforma y cuantifica para señalarse al descodificador. La aplicación de transformación de coseno discreta convencional (DCT) y la cuantificación con un parámetro de cuantificación medio o grueso tiende a dar coeficientes de AC nulos y coeficientes de DC no nulos a partir de residuos obtenidos en el modo plano. Para evitar esto, una realización de la presente invención usa un núcleo de transformación secundario, en lugar del núcleo de transformación primario, para transformar un residuo obtenido en el modo plano. Además, otra realización realiza cuantificación adaptativa en el modo plano en la que el parámetro de cuantificación cambia de manera adaptativa según la actividad espacial en el bloque objetivo.

En una realización de la presente invención, el valor P_P (N,N) del píxel de la parte inferior derecha se calcula a partir de los píxeles de referencia. El valor P_P (N,N) se calcula según uno de los tres métodos siguientes:

Método 1:

$$P_P(N,N) = ((P_C(N,0) + P_C(0,N)) >> 1),$$

en la que el operador “>>” representa a operación de desplazamiento hacia la derecha con o sin redondeo.

Método 2:

$$P_P(N,N) = w_h \times P_C(N,0) + w_v \times P_C(0,N),$$

en la que w_h y w_v son pesos determinados, usando P_C (0,1 : N) y P_C (1 : N,0). Por ejemplo, w_h y w_v se calculan de la siguiente manera:

$$w_h = \frac{\text{var}(P_C(1:N,0))}{\text{var}(P_C(1:N,0) + \text{var}(P_C(0,1:N))}$$

$$w_v = \frac{\text{var}(P_C(0,1:N))}{\text{var}(P_C(1:N,0) + \text{var}(P_C(0,1:N))}$$

en la que el operador “var( )” representa una operación para calcular una varianza.

Método 3:

$$P_P(N,N) = ((P_C^f(N,0) + P_C^f(0,N)) >> 1),$$

en la que $P_C^f(0,N) = f(P_C(0,0), P_C(0,1), ..., P_C(0,2N))$ y

$P_C^f(N,0) = f(P_C(0,0), P_C(1,0), ..., P_C(2N,0))$. $y = f(x_0, x_1, ..., x_{2N})$ representa una operación aritmética. En una realización de la presente invención, la operación aritmética se define como $y = f(x_0, x_1, ..., x_{2N}) = \frac{x_{N-1} + 2x_N + x_{N+1}}{4}$. En otra realización de la presente invención, la operación aritmética se define simplemente como $y = f(x_0, x_1, ..., x_{2N}) = x_{2N}$. Obsérvese que en la presente invención, el valor P_P (N,N) del píxel de la parte inferior derecha no se señala al descodificador. En cambio, el descodificador calcula el valor P_P (N,N) según el método adoptado por el codificador, que puede estar predeterminado, o la identificación del cual puede señalarse al descodificador.

La figura 10 es una vista esquemática que muestra el proceso de predicción de valores de píxel realizado en el modo plano según la realización de la presente invención, en el que está implementado el método 1 anterior. El proceso comienza con calcular el valor P_P (N,N) del píxel de la parte inferior derecha en un bloque usando el método 1. Después de que se calcula el valor P_P (N,N), se realizan interpolaciones lineales entre el valor P_P (N,N) y el valor de píxel de referencia P_C (N,0) para obtener valores de píxel predichos P_P (N,j) de la fila inferior en el bloque. Del mismo modo, se realizan interpolaciones lineales entre el valor P_P (N,N) y el valor de píxel de referencia P_C (0,N) para obtener valores de píxel predichos P_P (i,N) de la columna más a la derecha en el bloque. Tras esto, se realizan interpolaciones bilineales de entre los valores de píxel predichos P_P (N,j) y P_P (i,N) y los valores de píxel de referencia P_C (i,0) y P_C (0,j) para obtener el resto de los valores de píxel P_P (i,j) en el bloque. Tal como se muestra mediante las siguientes ecuaciones y la figura 11, el método 1 puede simplificar la operación de predicción de los valores de píxel P_P (i,j) en un bloque objetivo:

$$P_P(i,j) = ((P_P^h(i,j) + P_P^v(i,j)) >> 1), \ \forall 1 \le i,j \le N,$$

en los que $P_P^h(i,j) = \frac{(N-j) \times P_C(i,0) + j \times P_C(0,N)}{N}$ y

$$P_P^v(i,j) = \frac{(N-i) \times P_C(0,j) + i \times P_C(N,0)}{N}$$ si se necesita exactitud relativa.

Las ecuaciones anteriores requieren divisiones por el valor N para calcular los valores de píxel P_P (i,j) en el bloque. Las operaciones de división pueden evitarse usando una operación aritmética entera tal como sigue:

$$P_P(i,j) = ((P_P^h(i,j) + P_P^v(i,j)) >> (1 + \log_2 N), \ \forall 1 \le i,j \le N,$$

en los que $P_P^h(i,j) = (N-j) \times P_C(i,0) + j \times P_C(0,N)$ y

$$P_P^v(i,j) = (N-i) \times P_C(0,j) + i \times P_C(N,0)$$

Si la exactitud entera es suficiente, los valores de píxel P_P (i,j) pueden expresarse mediante

$$P_P(i,j) = ((P_P^h(i,j) + P_P^v(i,j)) >> 1), \ \forall 1 \le i,j \le N$$

En los que $P_P^h(i,j) = ((N-j) \times P_C(i,0) + j \times P_C(0,N)) >> (\log_2 N)$ y

$$P_P^v(i,j) = ((N-i) \times P_C(0,j) + i \times P_C(N,0)) >> (\log_2 N)$$

El método 1 puede modificarse tal como sigue:

$$P_P(i,j) = ((P_P^h(i,j) + P_P^v(i,j) >> 1), \ \forall 1 \le i,j \le N$$

$$P_P^h(i,j) = \frac{(N-j) \times P_C(i,0) + j \times P_C^f(0,N)}{N}$$

$$P_P^v(i,j) = \frac{(N-i) \times P_C(0,j) + i \times P_C^f(N,0)}{N}$$

$$P_C^f(0,N) = f(P_C(0,0), P_C(0,1), ..., P_C(0,2N))$$
$$P_C^f(N,0) = f(P_C(0,0), P_C(1,0), ..., P_C(2N,0)),$$

en el que $y = f(x_0, x_1, ..., x_{2N})$ representa una operación aritmética. En una realización de la presente invención, la operación aritmética se define como $y = f(x_0, x_1, ..., x_{2N}) = \frac{x_{N-1} + 2x_N + x_{N+1}}{4}$. En otra realización de la presente invención, la operación aritmética se define simplemente como $y = f(x_0, x_1, ..., x_{2N}) = x_{2N}$.

El método 1 puede modificarse tal como sigue:

$$P_P(i,j) = ((P_P^h(i,j) + P_P^v(i,j) >> 1), \ \forall 1 \le i,j \le N$$

$$P_P^h(i,j) = \frac{(N-j) \times P_C(i,0) + j \times P_C^f(i,N)}{N}$$

$$P_P^v(i,j) = \frac{(N-i) \times P_C(0,j) + i \times P_C^f(N,j)}{N}$$

$$P_C^f(i,N) = g(i, P_C(0,0), P_C(0,1), ..., P_C(0,2N))$$
$$P_C^f(N,j) = g(j, P_C(0,0), P_C(1,0), ..., P_C(2N,0)),$$

en el que $y = g(i, x_0, x_1, ..., x_{2N})$ representa una función que puede definirse mediante una de las cuatro ecuaciones siguientes:

Ecuación 1:

$$y = g(i, x_0, x_1, ..., x_{2N}) = x_{2N}$$

Ecuación 2:

$$y = g(i, x_0, x_1, ..., x_{2N}) = x_{(N+i)}$$

Ecuación 3:

$$y = g(i, x_0, x_1, ..., x_{2N}) = \frac{(N-i) \times x_N + i \times x_{2N}}{N}$$

Ecuación 4:

$y = g(i, x_0, x_1, ..., x_{2N}) = x_{(N+i)}^f$; en la que $x_{(N+i)}^f$ es un valor filtrado de $x_{(N+i)}$ cuando se aplica un filtro sobre la

matriz $[x_0, x_1, ...x_{2N}]$. En una realización de la presente invención, el filtro puede ser un filtro de tres pasos $\frac{[1,2,1]}{4}$.

En las realizaciones anteriores, se asume que los píxeles de referencia y horizontales $\{P(i,0): 0 \le i \le 2N\} \cup \{P(0,j): 0 \le j \le 2N\}$ están todos disponibles para la predicción. Los píxeles de referencia pueden no estar disponibles si el bloque objetivo está ubicado en un límite de segmento o trama. Si los píxeles de referencia verticales $\{P(i,0): 0 \le i \le 2N\}$ no están disponibles para la predicción, pero los píxeles de referencia horizontales $\{P(0,j): 0 \le j \le 2N\}$ están disponibles, la asignación $P_C(i,0) = P_C(0,1), \forall 1 \le i \le 2N$ se realiza para generar los píxeles de referencia verticales para la predicción. Si los píxeles de referencia horizontales $\{P(0,j): 0 \le i \le 2N\}$ no están disponibles para la predicción pero los píxeles de referencia verticales $\{P(i,j): 0 \le j \le 2N\}$ están disponibles, la asignación de $P_C(0,j) = P_C(1,0), \forall 1 \le i \le 2N$ se realiza para generar los píxeles de referencia horizontales para la predicción. Si ni los píxeles de referencia verticales ni los píxeles de referencia horizontales están disponibles para la predicción, la asignación de $P_C(i,0) = P_C(0,j) = (1 << (N_b - 1)), \forall 1 \le i, j \le 2N$ se realiza para generar tanto píxeles de referencia verticales como horizontales. En la ecuación, N_b representa la profundidad de bits usada para representar los valores de píxel.

En una realización de la presente invención, tal como los bloques de predicción generados en los otros modos de predicción, un bloque de predicción generado en el modo plano se usa para obtener un residuo $P_R(1:N,1:N)$ que se transforma mediante el módulo de transformación 405 y se cuantifica mediante el cuantificador 406. El residuo transformado y cuantificado $P_T(1:N,1:N)$ se señaliza en un flujo de bits al descodificador. Además, el residuo transformado y cuantificado $P_T(1:N,1:N)$ se transforma y se cuantifica de manera inversa mediante el módulo de transformación inversa 410 y el cuantificador inverso 411 para convertirse en un residuo comprimido $P_Q(1:N,1:N)$, que se almacena en la memoria de trama 403 para su uso en la predicción de bloques objetivo posteriores.

El residuo transformado y cuantificado completo $P_T(1:N,1:N)$ puede señalarse en un flujo de bits al descodificador. De manera alternativa, solo una parte del residuo $P_T(1:K,1:K)$ puede señalarse en un flujo de bits al descodificador. K es más pequeño que N (K < N) y se establece a un valor predeterminado, por ejemplo, 1. El valor de K puede señalarse en un flujo de bits al descodificador. Si el descodificador recibe solo una parte del residuo $P_T(1:K,1:K)$ descodifica la parte del residuo y establece 0 para la parte restante del residuo. Aunque solo una parte del residuo se señaliza al descodificador, el residuo completo $P_T(1:N,1:N)$ se transforma y cuantifica de manera inversa para obtener un residuo comprimido $P_Q(1:N,1:N)$ con el fin de predecir bloques objetivo posteriores.

Además, en otra realización de la presente invención, el parámetro de cuantificación se cambia de manera adaptativa para cuantificar un residuo generado en el modo plano. El modo plano se aplica a un bloque con una imagen suavizada cuyos valores de píxel cambian gradualmente con un gradiente plano pequeño. Un residuo a partir de un bloque suavizado de este tipo tiende a cuantificarse a cero con un parámetro de cuantificación medio o grueso. Para garantizar que la cuantificación produce coeficientes no nulos, en la realización de la presente invención, el parámetro de cuantificación se conmuta a un parámetro de cuantificación más fino cuando se cuantifica un residuo generado en el modo plano. El parámetro de cuantificación (QP_{plano}) usado para cuantificar un residuo generado en el modo plano puede definirse con un parámetro de cuantificación de base (QP_{base}). Q_{base} puede establecerse a un valor predeterminado que representa un parámetro de cuantificación más fino. Si QP_{baseP} no es conocido para el descodificador, puede señalarse en un flujo de bits al descodificador, o más específicamente señalarse en el encabezado de segmento o en el parámetro de imagen establecido, tal como se define en el documento H.264/AVC.

En una realización de la presente invención, QP_{plano} se establece simplemente a QP_{baseP} ($QP_{plano} = QP_{baseP}$). QP_{plano} puede definirse con una suma de QP_{baseP} y QP_N ($QP_{plano} = QP_{baseP} + QP_N$), donde QP_N se determina, usando una tabla de consulta que indica valores de QP_N en relación con valores de N. QP_{plano} puede definirse de manera alternativa como $QP_{plano} = QP_{baseP} + QP_{diff}$ (N). QP_{diff} (N) es una función del valor N y se señaliza en un flujo de bits al descodificador, o más específicamente se señala en el encabezado de segmento o en el parámetro de imagen establecido, tal como se define en el documento H.264/AVC. El descodificador determina QP_{diff} (N) a partir del flujo de bits para cada uno de los valores N soportados en su esquema de códec de vídeo.

En otra realización de la presente invención, añadiendo un parámetro de cuantificación diferencial (QP_{delta}), QP_{baseP} se modifica como $QP_{baseP} = QP_{baseP} + QP_{delta}$. QP_{delta} es un parámetro de cuantificación determinado a partir de una actividad espacial en un bloque o grupo de bloques para ajustar QP_{baseP} de manera adaptativa a la actividad espacial. QP_{delta} está señalado en un flujo de bits al descodificador. Como QP_{delta} se determina a partir de una actividad espacial en un bloque, puede volverse nulo dependiendo del contenido de imagen en el bloque y no afecta

a QP_{baseP} para el modo de predicción plana.

Además, en otra realización de la presente invención, QP_{plano} se determina con un parámetro de cuantificación normal QP_{normal}, que se usa para cuantificar residuos generados en modos de predicción distintos del modo plano. En una realización de este tipo, QP_{plano} se determina según una de las siguientes cinco maneras:

1 $QP_{plano.} = QP_{normal}$

2. $QP_{plano} = QP_{normal} + QP_N$, donde QP_N se determina a partir de una tabla de consulta que indica valores de QP_N en relación a valores de N.

3. $QP_{plano} = QP_{normal} + QP_{diff}$ (N), donde QP_{diff} (N) es una función del valor N y se señaliza en un flujo de bits al descodificador.

4. $QP_{plano} = QP_{normal} + QP_{delta}$, donde QP_{delta} es un parámetro de cuantificación determinado a partir de una actividad espacial en un bloque o grupo de bloques para ajustar de manera adaptativa QP_{normal} y está señalado en un flujo de bits al descodificador.

5. $QP_{Plano} = QP_{normal} + QP_N + QP_{delta}$

En otra realización de la presente invención, el módulo de transformación 405 y el módulo de transformación inversa 410 usan un conjunto secundario de núcleos de transformación hacia delante e inversa $(G_F^H \text{ and } G_I^H)$ para la transformación hacia delante e inversa de un residuo generado en el modo plano, en lugar de usar el conjunto primario de núcleos de transformación hacia delante e inversa $(H_F^H \text{ and } H_I^H)$. El conjunto primario de núcleos de transformación se usa para transformar residuos generados en modos de predicción distintos del modo plano y adecuados para bloques en los que existe energía de alta frecuencia. Por otra parte, los bloques que van a someterse al modo de predicción plano tienen actividades espaciales bajas en los mismos y necesitan núcleos de transformación adaptados para bloques con imágenes suavizadas. En esta realización, el módulo de transformación 405 y el módulo de transformación inversa 410 se conmutan entre el conjunto primario de núcleos de transformación y el conjunto secundario de núcleos de transformación, tal como se muestra en la figura 12, y usan el conjunto primario de núcleo de transformación al transformar residuos generados en modos de predicción distintos del modo plano, mientras que usan el conjunto secundario de núcleo de transformación al transformar residuos generados en el modo de predicción plana. Sin embargo, obsérvese que el conjunto secundario de núcleo de transformación no se limita a transformar residuos generados en el modo de predicción plano y puede usarse para transformar residuos generados en modos de predicción distintos del modo plano.

El conjunto secundario de núcleo de transformación hacia delante (G_F^N) puede ser una aproximación de punto fijo obtenida a partir de una de las siguientes opciones:

Opción 1 (tipo-7 DST):

$$G_F^N(i,j) = k_i \times \text{sen}(\frac{(2i-1)j\pi}{2N+1}), \forall 1 \le i,j \le N$$

Opción 2 (tipo-4 DST):

$$G_F^N(i,j) = k_i \times \text{sen}(\frac{(2i-1)(2j-1)\pi}{4N}), \forall 1 \le i,j \le N$$

Opción 3 (tipo-2 DCT, conocida normalmente como DCT):

$$G_F^N(i,j) = k_i \times \cos(\frac{(i-1)(2j-1)\pi}{2N}), \forall 1 \le i,j \le N$$

Opción 4:

$G_F^N(i,j) = k_i \times H_F^{2N}(2i, N+1-j), \forall 1 \le i,j \le N$ si los núcleos de transformación de tamaño 2Nx2N (H_F^{2N}) son soportados por el códec de vídeo. De lo contrario,

$G_F^N(i,j) = H_F^N(i,j), \forall 1 \le i,j \le N$. Por tanto, en la opción 4, si los tamaños de transformación más pequeño y más grande soportados en un código de vídeo son 4x4 y 32x32, el conjunto secundario de núcleo de

transformación para un tamaño 4x4 se obtiene a partir del conjunto primario de núcleo de transformación para un tamaño 8x8. Del mismo modo, el conjunto secundario de núcleo de transformación para un tamaño 8x8 se obtiene a partir del conjunto primario de núcleo de transformación para un tamaño 16x 16, y el conjunto secundario de núcleo de transformación para un tamaño 16x 16 se obtiene a partir del conjunto primario de núcleo de transformación para un tamaño 32x32. Sin embargo, debido a la limitación de tamaño en la que el tamaño más grande soportado es 32x32, el conjunto secundario de núcleo de transformación para un tamaño 32x32 se obtiene a partir del conjunto primario de núcleo de transformación para un tamaño 32x32.

El factor de escala k_i puede definirse para satisfacer $\sum_{j=1}^{N}(G_F^N(i,j))^2 = 1, \forall 1 \le i \le N$. El factor de escala k_i puede usarse para ajustar el parámetro de cuantificación tal como se usa en el documento H.264/AVC. El conjunto secundario de núcleo de transformación inversa G_I^N puede obtenerse, usando el núcleo de transformación hacia delante G_F^N a partir de $G_I^N * G_F^N = I^N$, donde I^N representa la matriz de identificación de tamaño NxN.

Si el conjunto primario de núcleo de transformación satisface la propiedad $H_F^{2N}(i,j) = (-1)^{j+1} \times H_F^{2N}(i,2N+1-j), \forall 1 \le i,j \le 2N$ es preferible el conjunto secundario de núcleo de transformación definido en la opción 4. La opción 4 es ventajosa porque el conjunto secundario de núcleo de transformación no necesita almacenarse de manera independiente al conjunto primario de núcleo de transformación debido a que el conjunto secundario puede obtenerse a partir del conjunto primario. Si el conjunto primario de núcleo de transformación para un tamaño 2Nx2N (H_F^{2N}) es una aproximación de tipo-2 DCT, se satisface la propiedad anterior, y el conjunto secundario de núcleo de transformación para un tamaño NxN (G_F^N) puede ser una aproximación de tipo-4 DST. Si el conjunto primario de núcleo de transformación no satisface la propiedad anterior, es preferible el conjunto secundario de núcleo de transformación definido en la opción 1.

El modo de predicción plano puede seleccionarse de una de dos maneras. En la primera manera, un bloque de predicción generado en el modo de predicción plano se evalúa la eficiencia de codificación, junto con los bloques de predicción generados en los otros modos de predicción. Si el bloque de predicción generado en el modo plano muestra la mejor eficiencia de codificación de entre los bloques de predicción, se selecciona el modo plano. Alternativamente, se evalúa solo la eficiencia de codificación del modo plano. El modo de predicción plano es preferible para un área en la que una imagen se suaviza y su gradiente plano es pequeño. Por consiguiente, el contenido de un bloque objetivo se analiza para ver la cantidad de energía de alta frecuencia en el bloque y de las discontinuidades de imagen a lo largo de los bordes del bloque. Si la cantidad de energía de alta frecuencia sobrepasa un umbral, y no se encuentran discontinuidades importantes a lo largo de los bordes del bloque, se selecciona el modo plano. De lo contrario, se evalúan bloques de predicción generados en los otros modos de predicción para seleccionar un modo. En ambos casos, se señala una selección del modo de predicción plano en un flujo de bits al descodificador.

Aunque sin duda se le ocurrirán muchas alteraciones y modificaciones de la presente invención a un experto habitual en la técnica tras haber leído la descripción anterior, debe entenderse que no se pretende de ninguna manera que ninguna realización particular mostrada y descrita a modo de ilustración se considere limitativa. Por tanto, no se pretende que referencias a detalles de diversas realizaciones limiten el alcance de las reivindicaciones, que en sí mismas solo mencionan aquellas características que se consideran esenciales para la invención.

REIVINDICACIONES

1. Método de codificación de vídeo para predecir valores de píxel en un bloque objetivo en un modo plano, comprendiendo el método etapas ejecutables por ordenador ejecutadas mediante un procesador de un descodificador de vídeo para implementar:

calcular primeros valores de predicción para píxeles de predicción respectivos en un bloque de predicción para el bloque objetivo usando interpolación lineal entre valores de píxel de límite horizontal respectivos, en la misma posición horizontal que los píxeles de predicción respectivos, en la parte exterior superior del bloque objetivo y un valor de uno de los píxeles de límite vertical en la parte exterior izquierda del bloque objetivo;

calcular segundos valores de predicción para los píxeles de predicción respectivos usando interpolación lineal entre valores de los píxeles de límite vertical respectivos, en la misma posición vertical que los píxeles de predicción respectivos, y un valor de uno de los píxeles de límite horizontal en la parte exterior superior del bloque objetivo; y

promediar los valores de predicción primero y segundo para el mismo pixel de predicción para obtener valores de píxel de predicción respectivos en el bloque de predicción.

2. Método según la reivindicación 1, que comprende además:

descodificar un residuo señalado desde un codificador que se generó en el modo plano en un codificador; y

añadir el residuo descodificado al bloque de predicción para reconstruir datos de imagen.

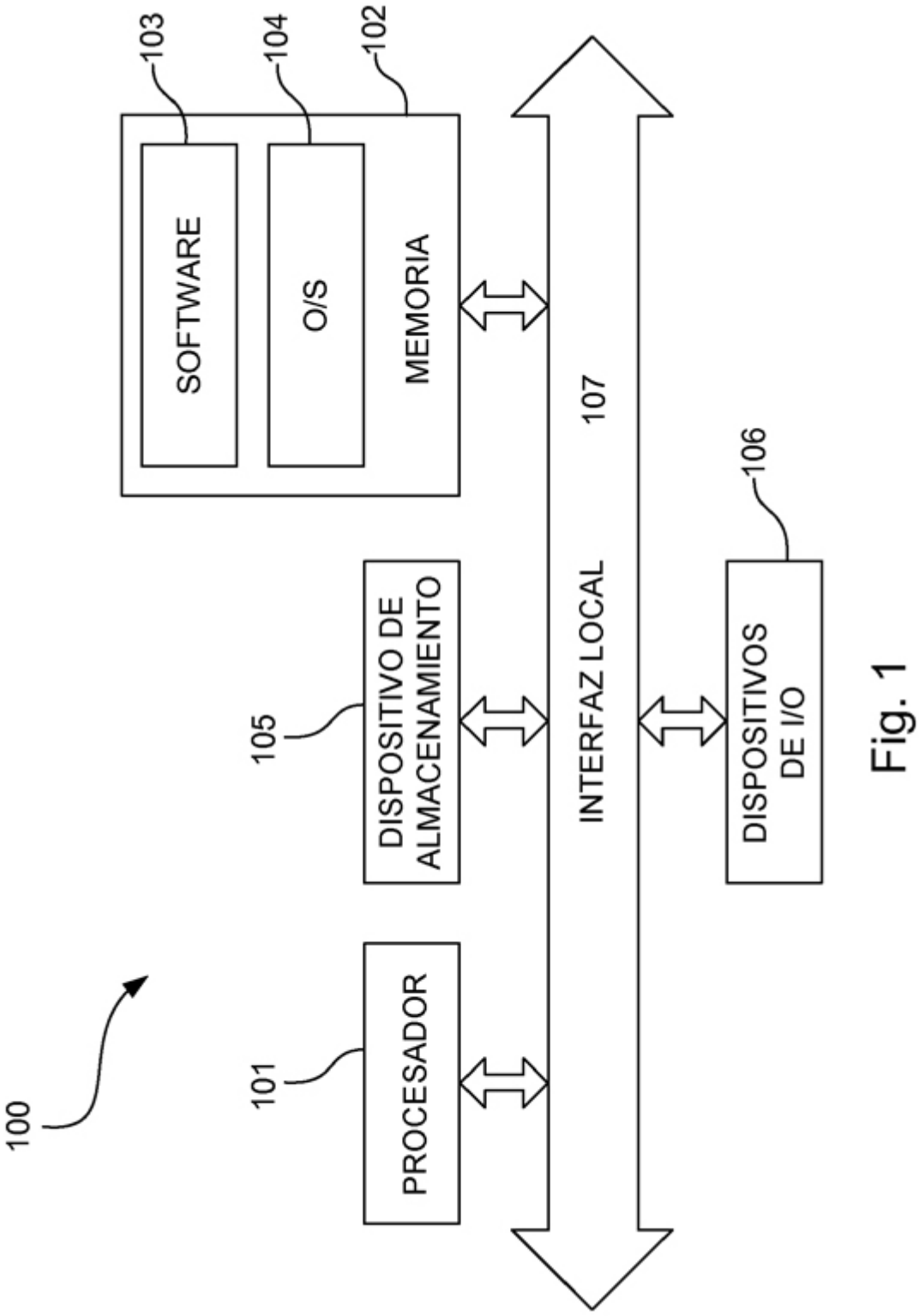

100
101
PROCESADOR
105
DISPOSITIVO DE ALMACENAMIENTO
INTERFAZ LOCAL
107
106
DISPOSITIVOS DE I/O
103
SOFTWARE
104
O/S
102
MEMORIA

Fig. 1

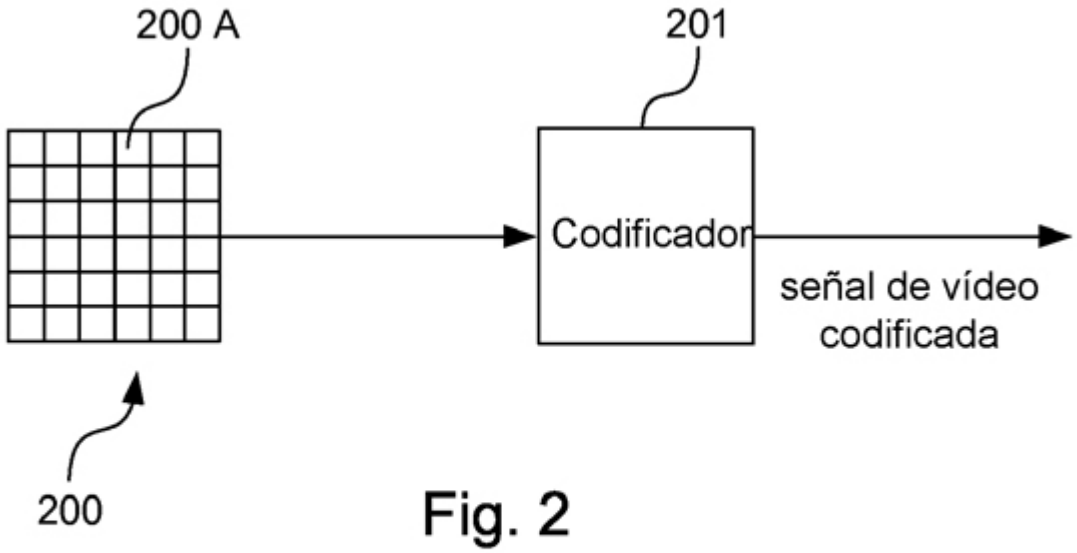

200 A
201
Codificador
señal de vídeo codificada
200

Fig. 2

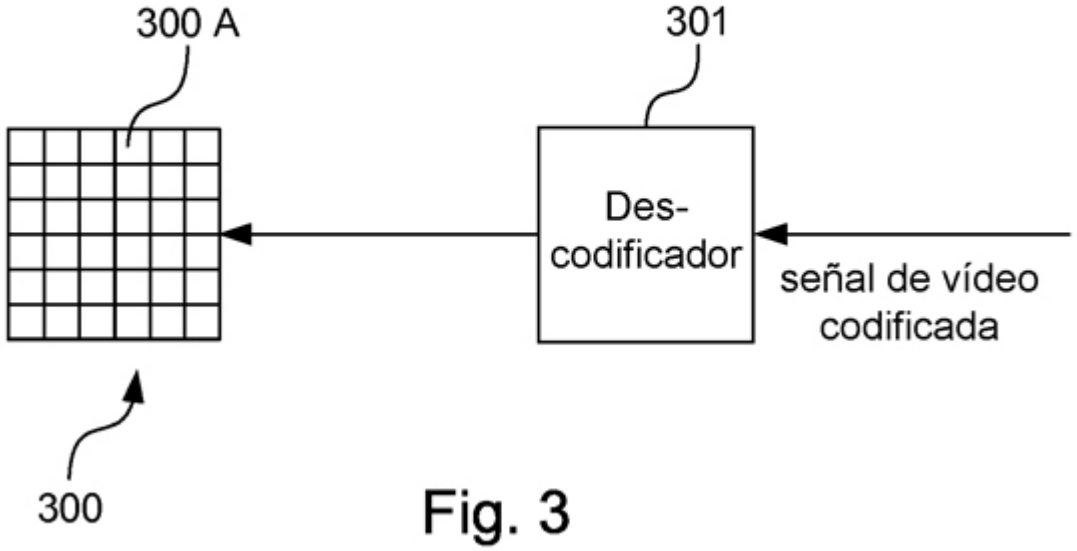

300 A
301
Des-codificador
señal de vídeo codificada
300

Fig. 3

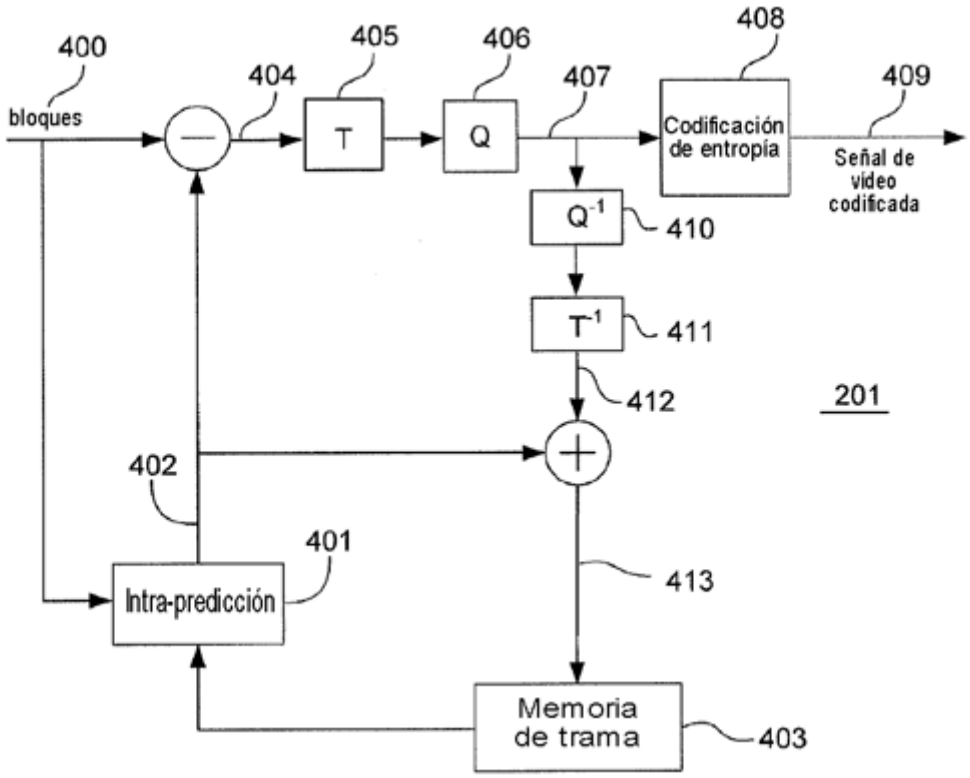

400
bloques
404
405
T
406
Q
407
408
Codificación de entropía
409
Señal de vídeo codificada
Q-1
410
T-1
411
412
201
402
401
Intra-predicción
413
Memoria de trama
403

Fig. 4

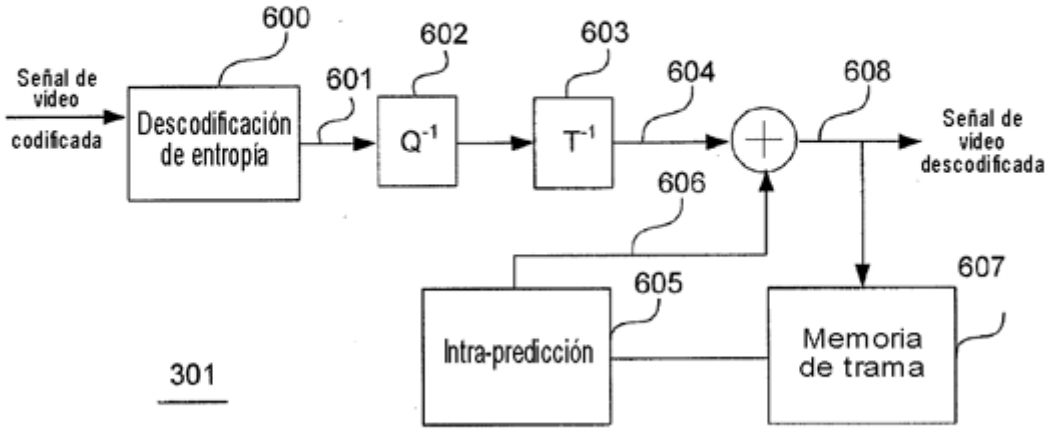
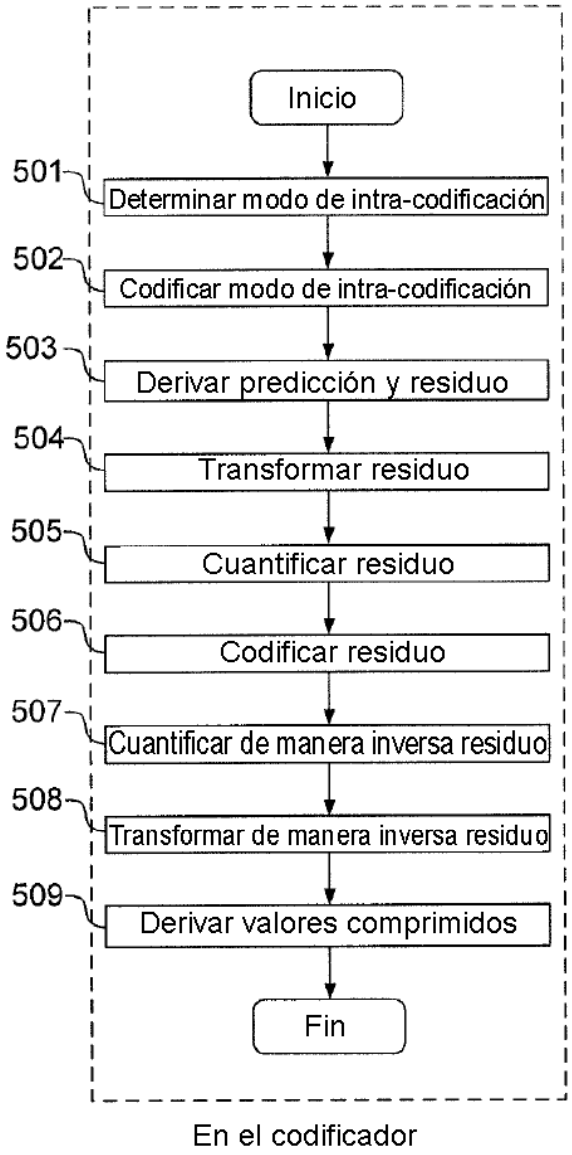

600
Señal de vídeo codificada
Descodificación de entropía
601
602
Q-1
603
T-1
604
608
Señal de vídeo descodificada
606
605
607
Intra-predicción
Memoria de trama
301

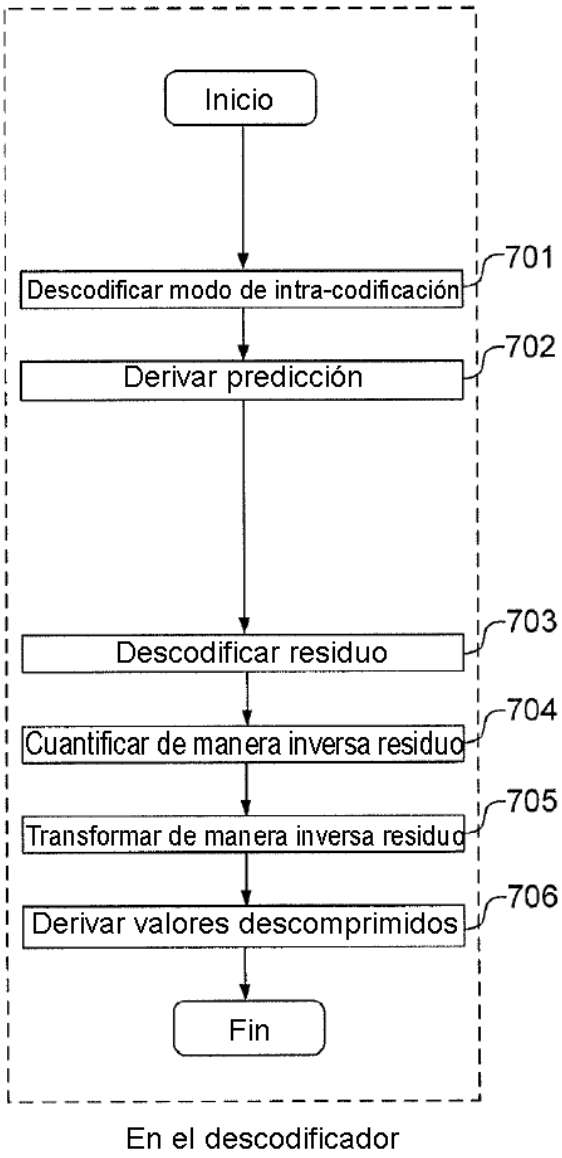
Fig. 6

DIAGRAMAS



En el codificador

FIG. 5



En el descodificador

FIG. 7

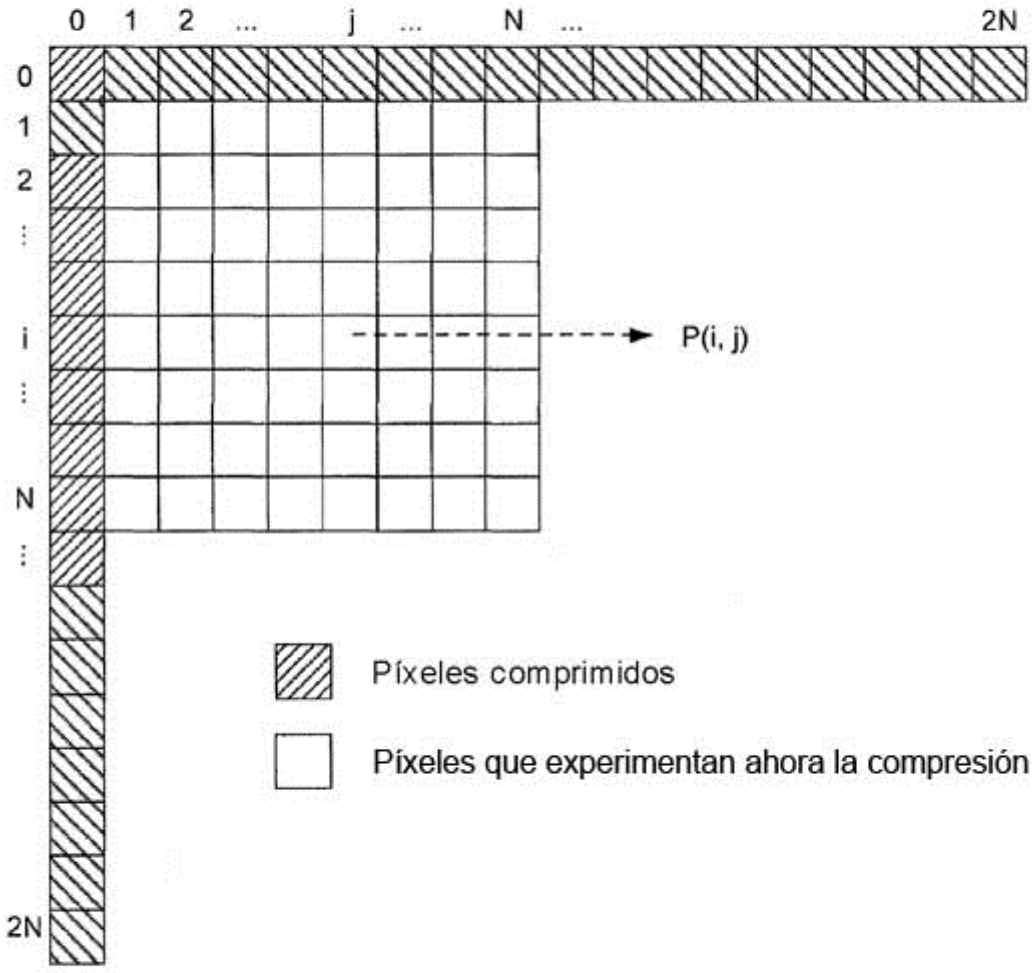

0 1 2 ... j ... N ... 2N
0
1
2
i
N
2N
P(i, j)
Píxeles comprimidos
Píxeles que experimentan ahora la compresión

FIG. 8

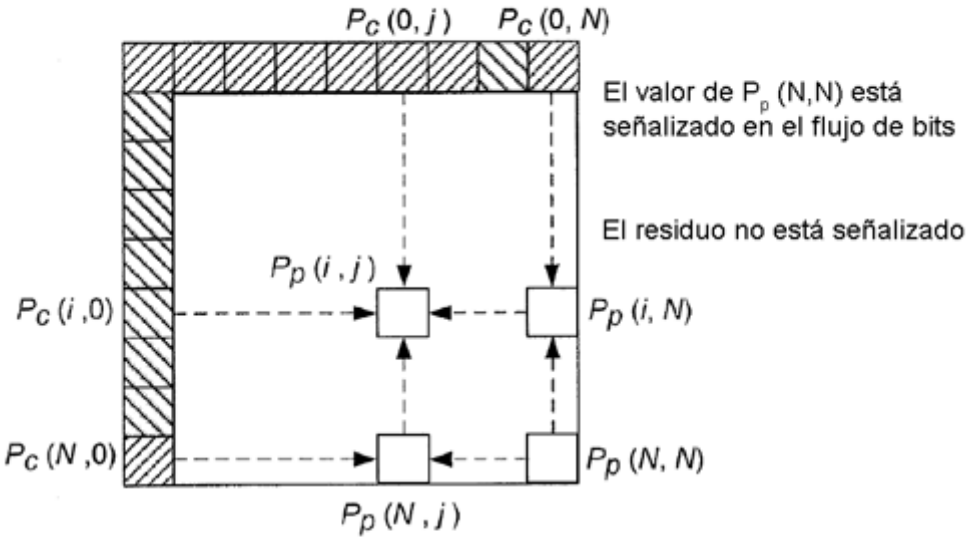

P_C (0, j)
P_C (0, N)
El valor de P_p (N,N) está señalizado en el flujo de bits
El residuo no está señalizado
P_p (i , j)
P_C (i ,0)
P_p (i, N)
P_C (N ,0)
P_p (N, N)
P_p (N , j)

FIG. 9

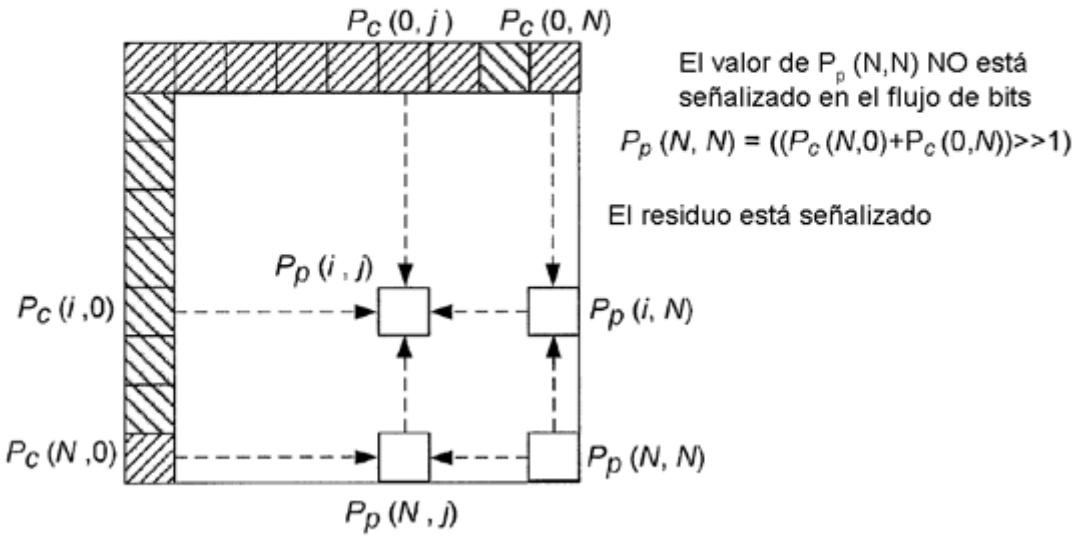

P_C (0, j)
P_C (0, N)
El valor de P_p (N,N) NO está señalizado en el flujo de bits
P_p (N, N) = ((P_c (N,0)+P_c (0,N))>>1)
El residuo está señalizado
P_p (i , j)
P_C (i ,0)
P_p (i, N)
P_C (N ,0)
P_p (N, N)
P_p (N , j)

FIG. 10

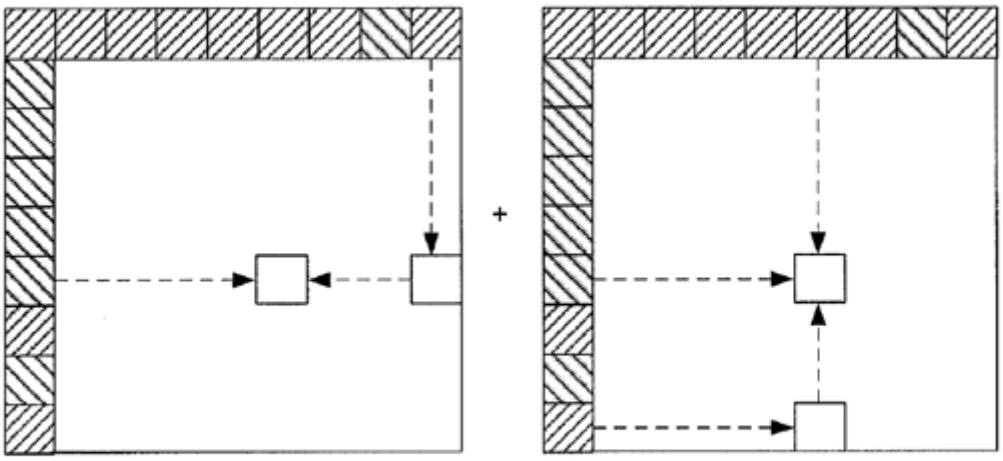

+

FIG. 11


Inicio
¿Modo de intra-codificación =plano?
No
Sí
Usar conjunto de transformación primario
Fin
Usar conjunto de transformación secundario

FIG. 12