

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 628 532**

51 Int. Cl.:

G06F 17/14 (2006.01)

H04N 19/126 (2014.01)

H04N 19/60 (2014.01)

H04N 19/61 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **27.08.2002 E 06020298 (3)**

97 Fecha y número de publicación de la concesión europea: **29.03.2017 EP 1750450**

54 Título: **Transformación y cuantificación de bloques mejoradas para codificación de imagen y vídeo**

30 Prioridad:

18.09.2001 US 955577

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

03.08.2017

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
ONE MICROSOFT WAY
REDMOND WA 98052, US**

72 Inventor/es:

MALVAR, HENRIQUE S.

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 628 532 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Transformación y cuantificación de bloques mejoradas para codificación de imagen y vídeo

Campo de la invención

5 La invención se refiere en general a la codificación y decodificación de señales de imagen y de vídeo, y más particularmente a una transformación de bloques mejorada y transformación inversa, junto con procedimientos de cuantificación y descuantificación, para codificar y decodificar señales de vídeo de imagen.

Antecedentes de la invención

10 La Transformada de Coseno Discreta (DCT) se usa comúnmente en codificación de transformación de bloques de imagen y vídeo (secuencias de imágenes), por ejemplo, JPEG y MPEG. Entre otras razones, esto es debido a que la DCT separa señales aleatorias (tales como las que corresponden a datos de imagen) en partes de frecuencia baja de importancia mayor y partes de frecuencia alta de importancia menor con respecto a la calidad visual de la imagen, facilitando por lo tanto la compresión.

15 A modo de ejemplo, H.26L es una tecnología de códec de imagen y/o vídeo que construye fotogramas usando bloques de píxeles de cuatro por cuatro. A diferencia de MPEG y JPEG, que usa bloques de ocho por ocho, H.26L obtiene compresión de calidad relativamente alta con bloques de píxeles de cuatro por cuatro usando información de predicción de otros bloques existentes en el mismo fotograma, (es decir, codificación intra-fotograma), además de estimación y compensación de movimiento entre fotogramas, (es decir, codificación inter-fotograma). En general, para conseguir predicción intra-fotograma, un codificador H.26L indica al correspondiente decodificador H.26L cuál otro bloque de píxeles anterior en el fotograma que se está creando se está usando como una base para la predicción, junto con identificar cuáles de los seis posibles predictores (fórmulas) usar al determinar los píxeles para el nuevo bloque (a partir de aquellos del bloque intra-fotograma anterior). Esto conduce a un error de predicción, que se proporciona también al decodificador para corregir el nuevo bloque. La información de error de predicción se codifica con la transformación de bloques (DCT) y se envía codificada al decodificador, para recalcular la información de error, que incluye mediante una transformación inversa, corregir el bloque predicho.

25 La codificación o decodificación de imagen y vídeo toma una cantidad significativa de potencia de procesamiento. Como es conocido, un codificador típicamente tiene mucha más potencia de procesamiento que muchos de los decodificadores que convierten los datos a imágenes, ya que los decodificadores se implementan típicamente en dispositivos de consumo. Por ejemplo, la decodificación de imagen y vídeo puede tener lugar en decodificadores de salón de televisión, asistentes digitales personales (PDA), ordenadores personales de tamaño de bolsillo y teléfonos celulares más avanzados.

30 Por lo tanto, cuando se consideran procedimientos de codificación y decodificación de imagen y vídeo, mantener la decodificación sencilla es importante, incluso si significa que la codificación tiene que realizarse computacionalmente de manera más compleja. Cualquier cosa que pueda simplificar la decodificación es deseable, con la codificación de que al hacer esto, los procedimientos de codificación y decodificación no impacten de manera adversa los niveles de compresión existentes, la calidad de imagen final y la cantidad de otros recursos necesarios en el decodificador en cualquier manera sustancial con relación a la tecnología existente.

35 T.D. Tran, "The BinDCT: Fast Multiplierless Approximation of the DCT", IEEE SIGNAL PROCESSING LETTERS, IEEE SIGNAL PROCESSING SOCIETY, Estados Unidos, (200006), vol. 7, n.º 6, ISSN 1070-9908, páginas 141 - 144, presenta una familia de transformaciones de bloque biortogonales rápidas denominadas binDCT que pueden implementarse usando únicamente operaciones de desplazamiento y adición. La transformación está basada en una estructura de malla amigable para VLSI que refuerza robustamente tanto las propiedades de fase lineal como de reconstrucción perfecta. Los coeficientes de malla están parametrizados como una serie de etapas de elevación diádica que proporcionan cálculo rápido, eficaz, en lugar de los coeficientes de transformación así como la capacidad para mapear números enteros a números enteros. Las nuevas transformaciones 8/spl veces/8 aproximan las conocidas DCT 8/spl veces/8 DCT de manera estrecha, consiguiendo un intervalo de ganancia de codificación de 8,77-8,82 dB, a pesar de requerir tan bajo como 14 desplazamientos y 31 adiciones por ocho muestras de entrada. La aplicación del binDCT tanto en codificación de imagen con pérdidas y sin pérdidas produce resultados muy competitivos en comparación con el rendimiento de la DCT de punto flotante original.

Sumario de la invención

50 La invención es lo que se reivindica en las reivindicaciones independientes.

Se especifican realizaciones preferidas en las reivindicaciones dependientes.

55 En resumen, la presente invención proporciona un procedimiento, sistema y transformación de bloques mejorados que simplifican significativamente la complejidad computacional para imágenes y vídeo tanto en el codificador como en el decodificador. Al mismo tiempo, la compresión, la calidad de imagen / vídeo y otros recursos únicamente se ven afectados de manera despreciable.

Más particularmente, se proporciona una transformación de bloques ortogonal y correspondiente transformación inversa con nuevas aproximaciones de números enteros a la Transformada de Coseno Discreta (DCT), junto con otros cambios informáticos que reducen significativamente la complejidad computacional tanto en el codificador como en el decodificador. De hecho, en una implementación, en el codificador y decodificador, el número de operaciones de transformación, por coeficiente, se ha reducido a cuatro adiciones y un desplazamiento en aritmética de 16 bits, (de cuatro adiciones y tres multiplicaciones en aritmética de 32 bits requeridas con la transformación especificada para H.26L). La presente invención transforma correctamente la información de píxel (por ejemplo, datos de corrección de errores) debido a la cuantificación (poner a escala y redondeo a un número entero) durante la codificación, y a la descuantificación durante la decodificación, mediante el uso de una de tres tablas seleccionadas basándose en cada posición del coeficiente, tener valores de parámetros que ya compensan factores de otras multiplicaciones de transformación, excepto para el realizado mediante la operación de desplazamiento durante los procedimientos de transformación y transformación inversa.

Además, durante la decodificación, en cada coeficiente cuantificado, el decodificador puede realizar una multiplicación de dieciséis bits para descuantificar ese coeficiente cuantificado en un coeficiente de transformación, en lugar de una multiplicación de treinta y dos bits (como se requiere en H.26L). En muchos dispositivos, con imágenes y vídeo típicos, los beneficios de procesamiento de la presente invención son por lo tanto significativos con relación a H.26L, especialmente en el decodificador, únicamente con una pérdida de calidad y/o compresión despreciable, si es que la hubiera.

Otros beneficios y ventajas serán evidentes a partir de la siguiente descripción detallada cuando se toma en conjunto con los dibujos, en los que:

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques que representa un sistema informático ejemplar en el que puede incorporarse la presente invención;

la Figura 2 es un diagrama de bloques que representa componentes usados al codificar píxeles de datos de vídeo o de imagen en vídeo codificado, y decodificar el vídeo codificado en información de píxel, incluyendo transformación, cuantificación y descuantificación y transformación inversa de acuerdo con un aspecto de la presente invención;

la Figura 3 es una representación de una estructura en mariposa que representa la matriz de transformación mejorada de acuerdo con un aspecto de la presente invención; y

la Figura 4 es una representación de una estructura en mariposa que representa la matriz de transformación inversa mejorada de acuerdo con un aspecto de la presente invención.

Descripción detallada

ENTORNO DE OPERACIÓN EJEMPLAR

La Figura 1 ilustra un ejemplo de un entorno 120 informático adecuado en el que puede implementarse la invención, particularmente para decodificar datos de imagen y/o vídeo. El entorno 120 de operación es únicamente un ejemplo de un entorno informático adecuado y no se pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad de la invención. Otros sistemas, entornos y/o configuraciones informáticas bien conocidas que pueden ser adecuadas para uso con la invención incluyen, pero sin limitación, ordenadores personales, ordenadores de servidor, dispositivos de mano o portátiles, sistemas multiprocesador, sistemas basado en microprocesador, electrónica de consumo programable, PC de red, miniordenadores, ordenadores centrales, entornos informáticos distribuidos que incluyen cualquiera de los sistemas o dispositivos y similares. Por ejemplo, es probable que la imagen de codificación y/o los datos de imagen de vídeo a menudo se realicen en un ordenador con más potencia de procesamiento que los ordenadores personales portátiles modernos, pero no hay razón para que la codificación no pueda realizarse en el dispositivo ejemplar, o la decodificación en una máquina más potente.

La invención puede describirse en el contexto general de instrucciones ejecutables por ordenador, tales como módulos de programa, ejecutados mediante uno o más componentes u otros dispositivos. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos y así sucesivamente que realizan tareas particulares o implementan tipos de datos abstractos particulares. Típicamente la funcionalidad de los módulos de programa puede combinarse o distribuirse según se desee en diversas realizaciones. El dispositivo 120 informático típicamente incluye al menos alguna forma de medio legible por ordenador. Medio legible por ordenador puede ser cualquier medio disponible que pueda accederse por el dispositivo 120 informático. A modo de ejemplo, y no como limitación, el medio legible por ordenador puede comprender medio de almacenamiento informático y medio de comunicación. Medio de almacenamiento informático incluye, medio extraíble y no extraíble volátil y no volátil implementado en cualquier procedimiento o tecnología para almacenamiento de información tal como instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos. Medio de almacenamiento informático incluye, pero sin limitación, RAM, ROM, EEPROM, memoria flash u otra tecnología de memoria, CD-ROM, discos versátiles digitales (DVD) u otro almacenamiento óptico, casetes magnéticos, cinta magnética, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda usarse para almacenar la información deseada y que pueda accederse por el dispositivo 120

informático. Los medios de comunicación típicamente incorporan instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tal como una onda portadora u otros mecanismos de transporte e incluyen cualquier medio de entrega de información. La expresión “señal de datos modulada” significa una señal que tiene una o más de sus características establecidas o cambiadas de tal manera para codificar información en la señal. A modo de ejemplo, y no como limitación, medio de comunicación incluye medio cableado tal como una red cableada o conexión cableada directa, y medio inalámbrico tal como medio acústico, RF, infrarrojos y otro. Las combinaciones de cualquiera de lo anterior deberían incluirse también en el alcance de medio legible por ordenador.

La Figura 1 muestra componentes funcionales de un dispositivo informático portátil de este tipo, que incluye un procesador 122, una memoria 124, una pantalla 126, y un teclado 128 (que pueden ser un teclado físico o virtual). La memoria 124 incluye en general tanto memoria volátil (por ejemplo, RAM) como memoria no volátil (por ejemplo, ROM, tarjetas PCMCIA, y así sucesivamente). Un sistema 130 operativo está residente en la memoria 124 y se ejecuta en el procesador 122, tal como el sistema operativo Windows® CE de Microsoft® Corporation, u otro sistema operativo.

Uno o más programas 132 de aplicación se cargan en memoria 124 y se ejecutan en el sistema 130 operativo. Ejemplos de aplicaciones incluyen programas de correo electrónico, programas de agenda, programas de PIM (gestión de información personal), programas de procesamiento de textos, programas de hojas de cálculo, programas de explorador de internet y así sucesivamente. El ordenador 120 personal portátil puede incluir también un gestor 134 de notificación cargado en la memoria 124, que se ejecuta en el procesador 122. El gestor 134 de notificación maneja solicitudes de notificación, por ejemplo, desde los programas 132 de aplicación.

El ordenador 120 personal portátil tiene una fuente 136 de alimentación, que se implementa como una o más baterías. La fuente 136 de alimentación puede incluir adicionalmente una fuente de alimentación externa que reemplaza o recarga las baterías integradas, tal como un adaptador de CA o una base de acoplamiento alimentada.

El ordenador 120 personal portátil ejemplar representado en la Figura 1 se muestra con tres tipos de mecanismos de notificación externos: uno o más diodos 140 de emisión de luz (LED) y un generador 144 de audio. Estos dispositivos pueden estar directamente acoplados a la fuente 136 de alimentación de modo que cuando se activan, pueden permanecer encendidos durante una duración dictada por un mecanismo de notificación incluso aunque el procesador 122 del ordenador personal portátil y otros componentes deban desconectarse para conservar potencia de batería. El LED 140 permanece encendido preferentemente o de manera indefinida hasta que el usuario actúa. Obsérvese que las versiones modernas del generador 144 de audio usan demasiada potencia para las baterías de los ordenadores personales portátiles de hoy en día y por eso están configurados para apagarse cuando el resto del sistema lo hace o a alguna duración finita después de la activación.

CODIFICACIÓN Y DECODIFICACIÓN

Por medio de los antecedentes, la Transformada de Coseno Discreta (DCT) mapea un vector **x** de longitud N en un nuevo vector **X** de coeficientes de transformación mediante una transformación lineal **X=H x**, donde el elemento en la k-ésima fila y la n-ésima columna de **H** se define por

$$H_{kn} = H(k, n) = c_k \sqrt{\frac{2}{N}} \cos\left((n + 1/2) \frac{k\pi}{N} \right)$$

para k = 0, 1, ..., N-1, y n = 0, 1, ..., N-1, con $c_0 = \sqrt{2}$ y $c_k = 1$ para k > 0. La matriz de DCT es ortogonal, por lo que su inversa equivale a su traspuesta, es decir $\mathbf{x} = \mathbf{H}^{-1} \mathbf{X} = \mathbf{H}^T \mathbf{X}$.

Una desventaja de la DCT es que las entradas H(k,n) son números irracionales, y por esto los datos de entrada de números enteros x(n) se mapearán a coeficientes de transformación irracionales X(k). Como resultado, con ordenadores digitales, cuando las transformaciones directa e inversa se calculan en cascada, los datos de salida no son exactamente iguales a los datos de entrada. En otras palabras, si se calculan como $\mathbf{X} = \mathbf{H} \mathbf{x}$, y $\mathbf{u} = \text{redondear}(\mathbf{H}^T \mathbf{X})$, entonces no se cumple que u(n) = x(n) para toda n. Sin embargo, introduciendo factores de escala apropiados α , γ , por ejemplo, $\mathbf{X} = \gamma \mathbf{H} \mathbf{x}$ y $\mathbf{u} = \text{redondear}(\alpha \mathbf{H}^T \mathbf{X})$, entonces u(n) = G x(n), donde G es un número entero, para casi toda n, cuando α y γ se eligen para que sean lo suficientemente grandes, pero esto no garantiza un resultado exacto.

En un codificador de vídeo de movimiento compensado, por ejemplo, los datos de fotogramas decodificados pasados se usan como información de referencia para información de predicción que se usará para generar el fotograma actual. Por lo tanto, como parte de la codificación, el codificador genera sus propios fotogramas decodificados, por los que el codificador necesita calcular transformadas inversas. Si se usa fórmula $\mathbf{u} = \text{redondear}(\alpha \mathbf{H}^T \mathbf{X})$, entonces se usan diferentes formatos de punto flotante y estrategias de redondeo en diferentes procesadores que conducen a diferentes resultados. Como resultado, habrá una desviación entre los datos decodificados en el codificador frente a los decodificados por los decodificadores (que tienen diferentes procesadores), en la cual la imagen empeora cada vez más por cada nuevo fotograma, puesto que el codificador

está basando en predicción / información de movimiento en fotogramas de bloques que cada vez son menos y menos como los fotogramas de bloques que el decodificador está produciendo.

Una solución al problema de desviación de datos aproxima la matriz \mathbf{H} por una matriz que contiene únicamente números enteros, en la cual se eliminan errores de redondeo. Si las filas de \mathbf{H} son ortogonales y tienen la misma norma (suma de los cuadrados), entonces se deduce que \mathbf{u} puede calcularse exactamente en aritmética de números enteros para todos los enteros \mathbf{x} . En otras palabras, cuando la transformación directa se calcula por $\mathbf{X} = \mathbf{H} \mathbf{x}$ y la transformación inversa por $\mathbf{u} = \mathbf{H}^T \mathbf{X}$, entonces resulta $\mathbf{u} = \mathbf{G} \mathbf{x}$, donde \mathbf{G} es un número entero igual a la norma cuadrada de cualquiera de las filas en \mathbf{H} .

Una manera para generar aproximaciones de números enteros a la DCT es usando la fórmula general:

$$Q(k, n) = \text{redondear}(\alpha H(k, n))$$

donde α es un parámetro de escala.

En la norma de compresión de vídeo H.26L, la imagen está compuesta de bloques de cuatro píxeles por cuatro píxeles, en la cual $N = 4$ en la fórmula de DCT en H.26L. Esto da como resultado que la matriz de DCT equivalga a:

$$\mathbf{H} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ c & s & -s & -c \\ 1 & -1 & -1 & 1 \\ s & -c & c & -s \end{bmatrix}$$

donde $c \triangleq \sqrt{2} \cos(\pi/8)$ y $s \triangleq \sqrt{2} \sin(\pi/8)$.

La matriz de transformación en la versión actual de H.26L se obtiene estableciendo $\alpha = 26$, en la cual los valores calculan:

$$\mathbf{Q}_0 = \text{redondear}(26\mathbf{H}) = \begin{bmatrix} 13 & 13 & 13 & 13 \\ 17 & 7 & -7 & -17 \\ 13 & -13 & -13 & 13 \\ 7 & -17 & 17 & -7 \end{bmatrix}$$

Con un valor de parámetro de escala de 26, las filas y columnas de \mathbf{Q}_0 son ortogonales entre sí (es decir, el producto interno de cualesquiera dos columnas es cero), y todas las filas y columnas tienen una norma igual a 676. De hecho, para valores en los que $\alpha < 50$, únicamente $\alpha = 2$ o $\alpha = 26$ proporcionan matrices ortogonales con filas de igual norma. Sin embargo, la solución para $\alpha = 2$ no conduce a buena compresión, y los valores mayores para α no son atractivos debido al aumento en la complejidad computacional (por ejemplo, longitud de palabra) requeridos para calcular los resultados de la transformación directa $\mathbf{X} = \mathbf{Q}_0 \mathbf{x}$. Por lo tanto, se ha elegido hasta ahora $\alpha = 26$ en H.26L.

La transformación inversa se define por $\mathbf{x}' = \mathbf{Q}_0 \mathbf{X}$, de modo que puede calcularse también con aritmética de números enteros. A partir de la definición anterior, $\mathbf{x}' = 676 \mathbf{x}$, es decir los datos reconstruidos \mathbf{x}' son iguales a los datos originales \mathbf{x} amplificados por una ganancia de números enteros de 676 (que es la norma de cualquiera de las filas en \mathbf{Q}_0).

De acuerdo con un aspecto de la presente invención, se proporciona una matriz de transformación de bloques que usa aproximaciones de números enteros que son ortogonales, aunque tienen diferentes valores que reducen significativamente la complejidad computacional cuando se codifica y decodifica. Más particularmente, usando $\alpha = 2,5$, se genera la siguiente matriz:

$$\mathbf{Q}_D \triangleq \text{redondear}(2,5\mathbf{H}) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

Obsérvese que las filas de Q_D son ortogonales entre sí, y, aunque sus normas son diferentes (las filas cero y dos tienen normas igual a cuatro, mientras que las filas uno y tres tienen normas iguales a diez), como se describe a continuación esto se maneja en cuantificación y descuantificación de manera que no aumenta significativamente la complejidad computacional.

5 Volviendo a la Figura 2 de los dibujos, se muestra un codificador 200 de transformación de bloques y decodificador 222 de transformación de bloques, conectable mediante algún tipo de medio y/o dispositivos 220 intermedios. Como se entiende, un codificador 200 no es necesario que haga su salida directamente disponible al decodificador 222, y por lo tanto la línea etiquetada 220 en la Figura 2 puede representar virtualmente cualquier tipo de medio o dispositivos, tales como un cable de red, medio de transmisión inalámbrica, línea de teléfono, dispositivo de almacenamiento, encaminador y/o virtualmente cualquier combinación de los mismos.

10 En general, el codificador 200 opera transformando cada bloque de $N \times N$ datos 202 de píxeles de entrada (por ejemplo, representando información de corrección de error) mediante unas transformaciones 204 bidimensionales que pueden separarse. Más particularmente, en primer lugar las filas de datos de píxel en el bloque se transforman mediante un procedimiento 206 de transformación de fila, y a continuación los resultados de la transformación de fila (no representadas específicamente en la Figura 2) se transforman mediante un procedimiento 208 de transformación de columna en los coeficientes 210 de transformación resultantes. En otras palabras, la DCT bidimensional esencialmente realiza una DCT unidimensional en cada fila de un bloque de píxeles, seguido por una DCT unidimensional en cada columna de los bloques de píxeles que se produjeron por las DCT unidimensionales en las filas. Las transformaciones de fila y columna pueden realizarse en orden inverso, con el mismo resultado obtenido.

20 En H.26L, la matriz de transformación especificada da como resultado las siguientes fórmulas para calcular los coeficientes de transformación:

$$A = 13a + 13b + 13c + 13d$$

$$B = 17a + 7b - 7c - 17d$$

$$C = 13a - 13b - 13c + 13d$$

$$D = 7a - 17b + 17c - 7d$$

25 donde [abcd] representa en primer lugar una fila de valores de píxeles, y a continuación, después de la transformación de fila, representa una columna de aquellos valores de datos transformados en fila, en un bloque de cuatro por cuatro. Sin embargo, en la práctica, estas fórmulas pueden simplificarse, y la matriz de DCT tiene una estructura recursiva que reduce el número de operaciones de multiplicación y adición requeridas. Sin embargo, la matriz anterior requiere al menos cuatro adiciones y tres multiplicaciones para calcular los coeficientes de transformación para cada píxel. Además, puesto que cada píxel puede ser un valor de nueve bits con signo, cuando se multiplica por los factores de fila y columna (la ganancia equivale a la norma de 676) el cálculo requiere aritmética de 32 bits, en la que un único cálculo lleva tanto tiempo como los cálculos de 16 bits. Aunque estas consideraciones no son normalmente tan significativas como durante la codificación, con estos valores de matriz especificados de H.26L, las operaciones adicionales y aritmética de 32-bits también tienen lugar durante la decodificación, en el que el gasto es significativo.

35 En contraste a la matriz de H.26L especificada, usando la misma representación de fórmula general, como se ha descrito anteriormente, la matriz de transformación de bloques de la presente invención es:

$$Q_D \triangleq \text{redondear}(2,5H) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

que proporciona las siguientes fórmulas para calcular los coeficientes de transformación:

$$A = a + b + c + d$$

$$B = 2a + b - c - 2d$$

$$C = a - b - c + d$$

$$D = a - 2b + 2c - d$$

5 Parte de la razón de que esta fórmula / matriz funciona de manera más óptima es que en lugar de realizar multiplicaciones individuales en la etapa de transformación, los factores de multiplicación se manejan esencialmente en la fase de cuantificación, simplemente cambiando los valores usados en la puesta a escala. Como resultado, la única multiplicación con estas fórmulas es por una potencia de dos, que en los procesadores se consigue mediante una única operación de desplazamiento a la izquierda, no una multiplicación real. Con u, v, y y z usadas como variables auxiliares, y en la que “<< 1” significa desplazamiento a la izquierda un bit (equivalente a multiplicar por dos pero computacionalmente más rápido, las fórmulas anteriores simplifican las siguientes fórmulas:

$$u = a + d;$$

$$v = b + c;$$

$$y = b - c;$$

$$z = a - d;$$

$$A = u + v;$$

$$C = u - v;$$

$$B = y + (z \ll 1);$$

10
$$D = \bar{z} - (y \ll 1);$$

Mediante la estructura en mariposa representada en la Figura 3, (en la que una línea entre dos puntos indica una suma, y los números en las líneas (los -1, 2 y -2 indican un factor de multiplicación) que hallan estos coeficientes de transformación realmente requieren únicamente cuatro adiciones y un desplazamiento para cada píxel dado, proporcionando una mejora de rendimiento muy sustancial con relación a los factores especificados en H.26L. Además, debido a los valores de coeficientes inferiores, todas las operaciones de transformación pueden realizarse en aritmética de 16 bits (con datos de píxeles de 9 bits). Más particularmente, con la matriz de transformación Q_D anteriormente definida, después de una transformación bidimensional 2-D, la amplificación de señal máxima es 36, en la cual los coeficientes de salida después de la transformación bidimensional abarcarán 6 bits más que la entrada. Por lo tanto, para entrada de 9 bits, los coeficientes de salida tienen un rango dinámico de 15 bits, que significa que la transformación directa bidimensional puede calcularse con aritmética de 16 bits. Además, como se describe a continuación, el aumento de rendimiento resultante de muchas menos operaciones, cada una de las cuales requiere únicamente aritmética de 16 bits, se repite esencialmente durante las operaciones de decodificación con la transformación inversa, proporcionando un aumento significativo en rendimiento de decodificación.

Una vez que se calculan los coeficientes 210 de transformación, el bloque de los coeficientes 210 se cuantifica poniendo a escala los valores y redondeándolos a sus números enteros más cercanos. Esto se representa en la Figura 2 por el procedimiento 212 de cuantificación, que selecciona de entre los valores de cuantificación a partir de los parámetros 214 de cuantificación para poner a escala los coeficientes 210 de transformación en coeficientes 216 cuantificados. Como se ha descrito anteriormente, debido a que no existe una norma, sino tres en la matriz mejorada, tres tablas de cuantificación Q0, Q1 y Q2 contienen estos valores, dependiendo de dónde se posicione en el bloque el coeficiente transformado cuantificar.

La fórmula de cuantificación para poner a escala y redondeo es como sigue, que debería calcularse con precisión de 32 bits:

$$L = [K \times A(QP, r) + fX] \gg 20$$

5 donde L es el coeficiente cuantificado, K es el coeficiente transformado, A(QP,r) es el factor de escala indexado por el parámetro de cuantificación QP y r, en el que r identifica qué tabla (Q0, Q1 o Q2 usar), fX está en el intervalo $[0, 0,5] \times 2^{20}$ (fX tiene el mismo signo que K), y $\gg 20$ significa desplazamiento a la derecha veinte lugares (divido por 1.048.576), para poner a escala el número hacia abajo. El procedimiento de cuantificación por lo tanto introduce errores. Obsérvese que aunque se usa aritmética de 32 bits para cuantificación, esto se requiere únicamente cuando se realiza codificación, que no es tan significativo como cuando se realiza decodificación (en el que la descuantificación necesita únicamente precisión de 16 bits, como se describe a continuación).

10 El índice r por lo tanto selecciona cuáles de las tres tablas de cuantificación Q0, Q1 o Q2 usar para ajustar las tres normas diferentes, basándose en la posición del coeficiente K en el bloque:

r=0 (usar Q0) si el coeficiente proviene de una de las posiciones {(0,0), (0,1), (1,0), (1,1)};
 r=1 (usar Q1) si el coeficiente proviene de una de las posiciones
 15 {(0,2), (0,3), (1,2), (1,3), (2,0), (2,1), (3,0), (3,1)};
 r=2 (usar Q2) si el coeficiente proviene de una de las posiciones {(2, 2), (2,3), (3,2), (3,3)}.

Después del desplazamiento a la derecha en 20 bits, cada resultado cuantificado L se ajusta a un número entero de 16 bits.

20 En una implementación ejemplar, el factor de escala A(QP, r) depende del parámetro de cuantificación QP y grupo de posición de coeficiente r de acuerdo con las siguientes tablas:

A(QP=0..31, r=0) = {104858, 93418, 83226, 74146, 66056, 58849, 52429, 46709, 41613, 37073, 33028, 29425, 26214, 23354, 20806, 18536, 16514, 14712, 13107, 11677, 10403, 9268, 8257, 7356, 6554, 5839, 5202, 4634, 4129, 3678, 3277, 2919};

25 A(QP=0..31, r=1) = {66318, 59082, 52636, 46894, 41778, 37220, 33159, 29541, 26318, 23447, 20889, 18610, 16579, 14771, 13159, 11723, 10444, 9305, 8290, 7385, 6580, 5862, 5222, 4652, 4145, 3693, 3290, 2931, 2611, 2326, 2072, 1846};

A(QP=0..31, r=2) = {41943, 37367, 33290, 29658, 26422, 23540, 20972, 18684, 16645, 14829, 13211, 11770, 10486, 9342, 8323, 7415, 6606, 5885, 5243, 4671, 4161, 3707, 3303, 2942, 2621, 2335, 2081, 1854, 1651, 1471, 1311, 1168}.

30 Los valores particulares anteriores se diseñaron para satisfacer la especificación de diseño en H.26L de que los tamaños de etapa de cuantificación deberían doblarse por cada incremento de seis en el índice de tabla. Para otras aplicaciones de codificación de vídeo o de imagen fija, pueden diseñarse otras tablas de cuantificación, en vista de los incrementos de fidelidad deseados. Obsérvese que para almacenar las tablas de cuantificación en el codificador, es necesario algún espacio de tabla adicional para almacenar tres tablas en lugar de una, pero esta cantidad de espacio adicional es únicamente 64 bytes, que es despreciable.

35 Volviendo a la Figura 2, que sigue al procedimiento de cuantificación, los coeficientes 216 cuantificados se alimentan a un codificador 218 por entropía que, en general, reduce adicionalmente el número de bits necesarios para codificar el bloque. Los codificadores por entropía (y decodificadores) son conocidos, y por lo tanto no se describen en el presente documento, ya que la presente invención opera antes y esencialmente de manera independiente de la codificación por entropía, y después y esencialmente de manera independiente de la decodificación por entropía.

40 Volviendo a una consideración de decodificación en el decodificador 222 de transformación de bloques, en algún momento, independientemente de cómo se entreguen, los bits de salida codificados por entropía se alimentan como bits de entrada a un decodificador 224 por entropía. En general, tales decodificadores por entropía son conocidos, y por lo tanto entre otras posibles operaciones, es suficiente establecer que el codificador 224 por entropía reproduzca los coeficientes 226 cuantificados para un bloque dado. Obsérvese que típicamente la codificación y decodificación no tienen pérdidas, es decir, los coeficientes 216 cuantificados alimentados en el codificador 218 por entropía serán idénticos a los coeficientes 226 cuantificados producidos por el decodificador 224 por entropía.

45 En general, el decodificador 222 de transformación de bloques refleja la operación del codificador 200 de transformación de bloques, aunque como se describe a continuación y de acuerdo con la presente invención, se proporciona una matriz de transformación inversa modificada que posibilita que se use aritmética de 16 bits a través de todo el procedimiento de decodificación, simplificando enormemente en general la complejidad computacional del decodificador.

50 Una vez que los coeficientes 226 cuantificados se recuperan del decodificador 224 por entropía, cada coeficiente cuantificado L se convierte a un valor reconstruido (descuantificado) K' mediante un procedimiento 228 de

descuantificación implementando la fórmula:

$$K' = L \times B(QP, r)$$

donde el factor de escala B depende del índice QP usado cuando se codifica, y r determina los parámetros de descuantificación 230 seleccionando un parámetro de una de las tablas D0, D1, o D2 (para ajustar las tres normas).
 5 Obsérvese que r puede deducirse a partir de la posición del coeficiente que se está descuantificando en el bloque, como se ha descrito anteriormente con respecto a cuantificación, aunque es factible enviar el valor r desde el codificador, que posiblemente reduciría la complejidad computacional en el decodificador pero requeriría que se enviaran al menos dos más bits por coeficiente para identificar r.

De esta manera, el factor de escala B(QP, r) depende del índice de parámetro de cuantificación para QP usado
 10 cuando se realiza codificación, y el grupo de posición de coeficiente r. En una implementación ejemplar, esa dependencia se especifica de acuerdo con las siguientes tablas:

B(QP=0..31, r=0) = {80, 90, 101, 113, 127, 143, 160, 180, 202, 226, 254, 285, 320, 359, 403, 453, 508, 570, 640, 718, 806, 905, 1016, 1140, 1280, 1437, 1613, 1810, 2032, 2281, 2560, 2874};

15 B(QP=0..31, r=1) = {101, 114, 127, 143, 161, 180, 202, 227, 255, 286, 321, 361, 405, 454, 510, 572, 643, 721, 810, 909, 1020, 1145, 1285, 1443, 1619, 1817, 2040, 2290, 2570, 2885, 3239, 3635};

B(QP=0..31, r=2) = {128, 144, 161, 181, 203, 228, 256, 287, 323, 362, 406, 456, 512, 575, 645, 724, 813, 912, 1024, 1149, 1290, 1448, 1625, 1825, 2048, 2299, 2580, 2896, 3252, 3650, 4095, 4596}.

Como con las tablas de codificación, los valores particulares anteriores se diseñaron para satisfacer la especificación
 20 de diseño en H.26L de que los tamaños de la etapa de cuantificación deberían doblarse por cada incremento de seis en el índice de tabla. Para otras aplicaciones de codificación de vídeo o de imagen fija, pueden diseñarse otras tablas de cuantificación, en vista de los incrementos de fidelidad deseados.

Como puede apreciarse, la fórmula de descuantificación pone a escala los coeficientes de vuelta a valores mayores,
 25 aunque cada uno de estos valores de escala es suficientemente bajo para asegurar que únicamente es necesaria aritmética de 16 bits en el decodificador. Debería observarse que puesto que cada entrada en una tabla está relacionada matemáticamente (basándose en la norma) para corresponder a entradas de tabla de QP indexados en las otras dos tablas, es posible de manera alternativa tener únicamente una tabla de valores QP, con un ajuste matemático apropiado basándose en el valor r, en lugar de tener tres tablas. Sin embargo, las consultas en tablas pequeñas son relativamente ineficaces, y el número de bytes requeridos para almacenar tres tablas de treinta y dos
 30 entradas en dos bytes por entrada en lugar de una tabla de treinta y dos entradas a cuatro bits por entrada (requerida con la transformación de H.26L especificada) son sesenta y cuatro bytes adicionales, que es insignificante en dispositivos informáticos modernos, considerando especialmente que los coeficientes de transformación generados por descuantificación se adaptan en dieciséis palabras de bits en lugar de palabras dobles de 32 bits, reduciendo de esta manera la cantidad total de memoria necesaria.

También como se representa en la Figura 2, una vez que se han descuantificado los coeficientes 226 cuantificados
 35 mediante la fórmula y tablas anteriores, está presente una matriz de cuatro por cuatro de coeficientes 232 de transformación reconstruidos. A partir de estos coeficientes 232 de transformación, los píxeles reconstruidos se generan alimentándolos a un procedimiento 234 de transformación inverso bidimensional que comprende los procedimientos 236 y 238 de transformación inversa de columna y fila, respectivamente.

De acuerdo con la presente invención, en lugar de usar la transformación inversa $Q_i = Q_D^T$, como se haría
 40 normalmente, para permitir el cálculo de descuantificación de 16 bits y transformación inversa, la presente invención define la siguiente matriz de transformación inversa para uso:

$$Q_i \triangleq \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}$$

Obsérvese que las columnas de Q_i son ortogonales entre sí, pero sus normas son diferentes. Sin embargo, esto se
 45 manejó mediante el valor "r" usado al seleccionar el parámetro de descuantificación, como se ha descrito anteriormente. Además, obsérvese que al generar Q_i desde Q_D , se realizó transposición en Q_D , con las columnas uno y tres multiplicadas por una mitad. De nuevo, sin embargo, los valores QP ya presentes en las tablas D0, D1 y D2 se han ajustado para esto con antelación, eliminando la necesidad de compensar cualquier complejidad computacional adicional. Además, la única "multiplicación" que será necesaria hacer cuando se calculan las transformaciones inversas con una matriz de este tipo es por una mitad, que realmente se realiza por una operación
 50 de desplazamiento a la derecha altamente eficaz. Obsérvese que la pequeña cantidad de ruido introducido por tal

división imprecisa no tiene esencialmente impacto en el rendimiento de tasa-distorsión.

Con Q_1 definida como anteriormente, su ganancia máxima equivale a cuatro. Por lo tanto, el procedimiento de transformación inversa bidimensional únicamente expande el intervalo dinámico en cuatro bits, permitiendo el cálculo en aritmética de 16 bits.

- 5 El procedimiento 236 de transformación inversa de columna y el procedimiento 238 de transformación inversa de fila (que opera en el resultado del procedimiento de transformación inversa de columna) representado en la Figura 2 generan los datos 240 de píxel de salida. Para este fin, usando la matriz de transformación inversa anterior y denominado [A B C D] una fila o columna particular para que se transforme a la inversa, los correspondientes valores transformados inversos [a' b' c' d'] se calculan por:

$$u = A + C;$$

$$v = A - C;$$

$$y = (B \gg 1) - D;$$

$$z = (D \gg 1) + B;$$

$$a' = u + z;$$

10

$$b' = v + y;$$

$$c' = v - y;$$

$$d' = u - z;$$

donde u, v, y y z son variables auxiliares y $\gg 1$ significa desplazar un bit a la derecha (equivalente a multiplicar por una mitad). Similar a la codificación, mediante la estructura en mariposa representada en la Figura 4, estas ecuaciones realmente reducen hasta cuatro operaciones de adición y una operación de desplazamiento por coeficiente, todas en aritmética de 16 bits. Finalmente, el procedimiento de transformación inversa 234 (Figura 2) pone a escala los valores de píxeles reconstruidos por un factor de 2^{-7} (realizado mediante una operación de desplazamiento a la derecha de 7 bits).

15

Los valores de píxeles reconstruidos después del desplazamiento de 7 bits tienen un intervalo de 9 bits, por lo que la transformación inversa puede calcularse en aritmética de 16 bits. Las tablas de descuantificación D0, D1 y D2 establecidas anteriormente son de manera que las ecuaciones de descuantificación no se desbordarán cuando se calculan en aritmética de 16 bits. Las siguientes tablas resumen algunas de las ventajas computacionales cuando se usan la transformación y la transformación inversa de la presente invención con relación a aquellas especificadas en H.26L:

20

Transformación	Operaciones por píxel, transformación directa	Cuantificación	Descuantificación	Operaciones por píxel, transformación inversa
Especificada en H.26L	4 adiciones, 3 multiplicaciones en aritmética de 32 bits	1 multiplicación en aritmética de 32 bits	1 multiplicación en aritmética de 32 bits	4 adiciones, 3 multiplicaciones en aritmética de 32 bits
Presente invención	4 adiciones, 1 desplazamiento, en aritmética de 16 bits	1 multiplicación en aritmética de 32 bits	1 multiplicación en aritmética de 16 bits	4 adiciones, 1 desplazamiento, en aritmética de 16 bits

25

Como puede apreciarse fácilmente, tales ahorros son significativos, particularmente en procesadores de 16 bits. Por ejemplo, en la práctica un procesador de PDA de 16 bits típico puede ver una mejora en la decodificación de aproximadamente el doble de la velocidad con la presente invención.

Además, la transformación / transformación inversa se ha probado con respecto a su impacto en calidad y compresión. Los resultados para ganancia de codificación (normalmente definidos como un aumento en la relación de señal a ruido) muestran que el uso de las transformaciones de la presente invención con datos de prueba dan como resultado una pérdida de únicamente alrededor de 0,01 dB con relación a las transformaciones de H.26L, que es despreciable, y de hecho es probable que sea mucho menor que con la información de señal de vídeo real, tal como errores de predicción de píxel. Además, el rendimiento global se probó con relación a las transformaciones de H.26L especificadas a través de cada uno de los treinta y dos parámetros de cuantificación, dando como resultado únicamente más o menos medio por ciento de intervalo de diferencias, con una media muy cercana a cero, indicando de esta manera que la transformación de la presente invención rindió así como la transformación especificada en H.26L.

Como puede observarse a partir de la anterior descripción detallada, se proporcionan unas matrices de transformación y transformación inversa mejoradas para codificación y decodificación de imagen o vídeo, respectivamente, que reducen significativamente la complejidad computacional con respecto a otras transformaciones conocidas sin impactar de manera adversa la compresión o calidad. Las multiplicaciones de transformación se eliminan aunque se obtienen resultados corregidos puesto que los valores de parámetros de cuantificación y descuantificación compensan aquellos factores de multiplicación, excepto para uno realizado por una operación de desplazamiento durante los procedimientos de transformación y transformación inversa. Debido a que los valores usados, las operaciones de transformación durante las operaciones de codificación, y descuantificación y transformación durante la decodificación pueden realizarse en aritmética de 16 bits (para datos de píxeles representados por nueve bits o menos).

Aunque la invención es susceptible de diversas modificaciones y construcciones alternativas, se muestran ciertas realizaciones ilustradas de la misma en los dibujos y se han descrito anteriormente en detalle. Debería entenderse, sin embargo, que no hay intención de limitar la invención a las formas específicas desveladas, sino por el contrario, la intención es cubrir todas las modificaciones, construcciones alternativas y equivalentes que caen en el alcance de la invención.

REIVINDICACIONES

1. Un procedimiento de operación de un codificador (200) de transformación de bloque para codificar información de píxeles de vídeo o de imagen, comprendiendo el procedimiento:

5 aplicar, mediante dicho codificador de transformación de bloque, un procedimiento (204) de transformación a la información (202) para construir coeficientes (210) de transformación a partir de la misma, aplicando el procedimiento de transformación cálculos de transformación que tienen valores de manera que cada operación de multiplicación en los cálculos de transformación se realiza mediante una operación de desplazamiento, en el que el procedimiento de transformación es ortogonal, en el que los valores de los cálculos de transformación están basados en valores representados en una matriz de transformación con varias filas que tienen diferentes normas, y en el que el procedimiento de transformación es una transformación bidimensional con transformaciones (208, 206) de columna y fila, teniendo la transformación bidimensional diferentes normas; y
 10 cuantificar (212), mediante dicho codificador de transformación de bloque, los coeficientes de transformación en coeficientes (216) de transformación cuantificados, en el que los coeficientes de transformación están organizados de manera lógica en un bloque, en el que la cuantificación pone a escala los coeficientes de transformación mediante factores de escala seleccionados a partir de varios grupos de factores de escala, en el que la selección está basada en la posición de coeficiente de transformación relativa en el bloque y un índice de parámetro de cuantificación, y en el que la cuantificación usa la selección de entre los varios grupos de factores de escala para al menos en parte ajustar las diferentes normas de la transformación bidimensional.

20 2. El procedimiento de la reivindicación 1, en el que la información comprende información de corrección de error para el bloque, y en el que el bloque es un bloque de 4x4.

3. El procedimiento de la reivindicación 1, en el que cada operación de multiplicación en los cálculos de transformación se realiza mediante una operación de desplazamiento a la izquierda, y en el que los cálculos de transformación consisten en dieciséis operaciones de desplazamiento a la izquierda, varias operaciones de adición, y varias operaciones de resta por bloque de 4x4 de la información.

25 4. El procedimiento de la reivindicación 1, en el que los cálculos de transformación se realizan ejecutando operaciones de multiplicación como operaciones de desplazamiento a la izquierda y pueden incluir dieciséis operaciones de desplazamiento a la izquierda, varias operaciones de adición, y varias operaciones de resta por bloque de 4x4 de la información.

30 5. El procedimiento de la reivindicación 1, en el que para cada uno de los coeficientes de transformación, la cuantificación incluye:

basándose en la posición del coeficiente de transformación en el bloque, seleccionar un grupo de factores de escala de entre los varios grupos de factores de escala, en el que un índice de grupo identifica el grupo seleccionado;
 35 dependiendo del índice de parámetro de cuantificación, seleccionar un factor de escala en el grupo seleccionado, en el que el índice de parámetro de cuantificación y el índice de grupo juntos indican el factor de escala seleccionado en una operación de búsqueda; y
 poner a escala el coeficiente de transformación mediante el factor de escala seleccionado.

6. El procedimiento de la reivindicación 1, en el que la transformación bidimensional usa cálculos basándose en valores representados en la matriz:

40
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}.$$

7. El procedimiento de la reivindicación 1, en el que para cada uno de los varios grupos de factores de escala, poner a escala con factores de escala del grupo aproximadamente se dobla para cada incrementos de seis en el valor de parámetro de cuantificación.

8. El procedimiento de la reivindicación 1, que comprende adicionalmente:

45 descuantificar (228) los coeficientes de transformación cuantificados en coeficientes de transformación descuantificados; y
 aplicar un procedimiento (234) de transformación inversa a los coeficientes de transformación cuantificados para generar información reconstruida a partir de los mismos, aplicando el procedimiento de transformación inversa cálculos de transformación inversa que tienen valores de manera que cada operación de multiplicación en los
 50 cálculos de transformación inversa se realiza mediante una operación de desplazamiento.

9. El procedimiento de la reivindicación 1, en el que el bloque de coeficientes de transformación es un bloque de 4x4, en el que los varios grupos de factores de escala consisten en tres grupos de factores de escala, en el que la transformación bidimensional tiene tres normas diferentes, y en el que cada uno de los tres grupos de factores de escala está adaptado para ajustar una norma diferente de las tres diferentes normas de la transformación bidimensional.

10. Un sistema para proporcionar información de vídeo o de imagen, comprendiendo el sistema:

un codificador de transformación de bloque que incluye:

1) un medio de transformación que aplica una transformación a la información para construir coeficientes de transformación a partir de la misma, aplicando el medio de transformación cálculos de transformación que tienen valores de manera que cada operación de multiplicación en los cálculos de transformación puede realizarse mediante una operación de desplazamiento, en el que la transformación es ortogonal, en el que los valores de los cálculos de transformación están basados en valores representados en una matriz de transformación con varias filas que tienen diferentes normas, y en el que la transformación es una transformación bidimensional con transformaciones de columna y fila, teniendo la transformación bidimensional diferentes normas; y

2) un medio de cuantificación que cuantifica los coeficientes de transformación en coeficientes de transformación de números enteros cuantificados, en el que los coeficientes de transformación están organizados de manera lógica en un bloque, en el que la cuantificación pone a escala los coeficientes de transformación mediante factores de escala seleccionados a partir de varios grupos de factores de escala, en el que la selección está basada en la posición de coeficiente de transformación relativa en el bloque y un índice de parámetro de cuantificación, y en el que la cuantificación usa la selección de entre los varios grupos de factores de escala para al menos en parte ajustar las diferentes normas de la transformación bidimensional.

11. El sistema de la reivindicación 10, en el que el sistema es un dispositivo informático portátil, comprendiendo adicionalmente el sistema:

un procesador para ejecutar el codificador de transformación de bloque;
 memoria;
 una batería;
 un teclado; y
 una pantalla.

12. El sistema de la reivindicación 10, en el que cada operación de multiplicación en los cálculos de transformación se realiza mediante una operación de desplazamiento a la izquierda de 1 bit, y en el que los cálculos de transformación consisten en dieciséis operaciones de desplazamiento a la izquierda de 1 bit, varias operaciones de adición, y varias operaciones de resta por bloque de 4x4 de la información.

13. El sistema de la reivindicación 10, en el que un índice de grupo identifica un grupo seleccionado de factores de escala, en el que el índice de parámetro de cuantificación y el índice de grupo juntos indican un factor de escala seleccionado en una operación de búsqueda.

14. El sistema de la reivindicación 10, en el que las transformaciones de columna y fila aplican cálculos de transformación de columna y fila basándose en valores representados en la matriz:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}.$$

15. El sistema de la reivindicación 10, que comprende adicionalmente:

un decodificador de transformación de bloque que incluye:

1) un medio de descuantificación que descuantifica los coeficientes de transformación de números enteros cuantificados en coeficientes de transformación cuantificados; y

2) un medio de transformación inversa que genera información reconstruida a partir de los coeficientes de transformación descuantificados, aplicando el medio de transformación inversa cálculos de transformación inversa que tienen valores, de manera que cada operación de multiplicación en los cálculos de transformación inversa se realiza mediante una operación de desplazamiento.

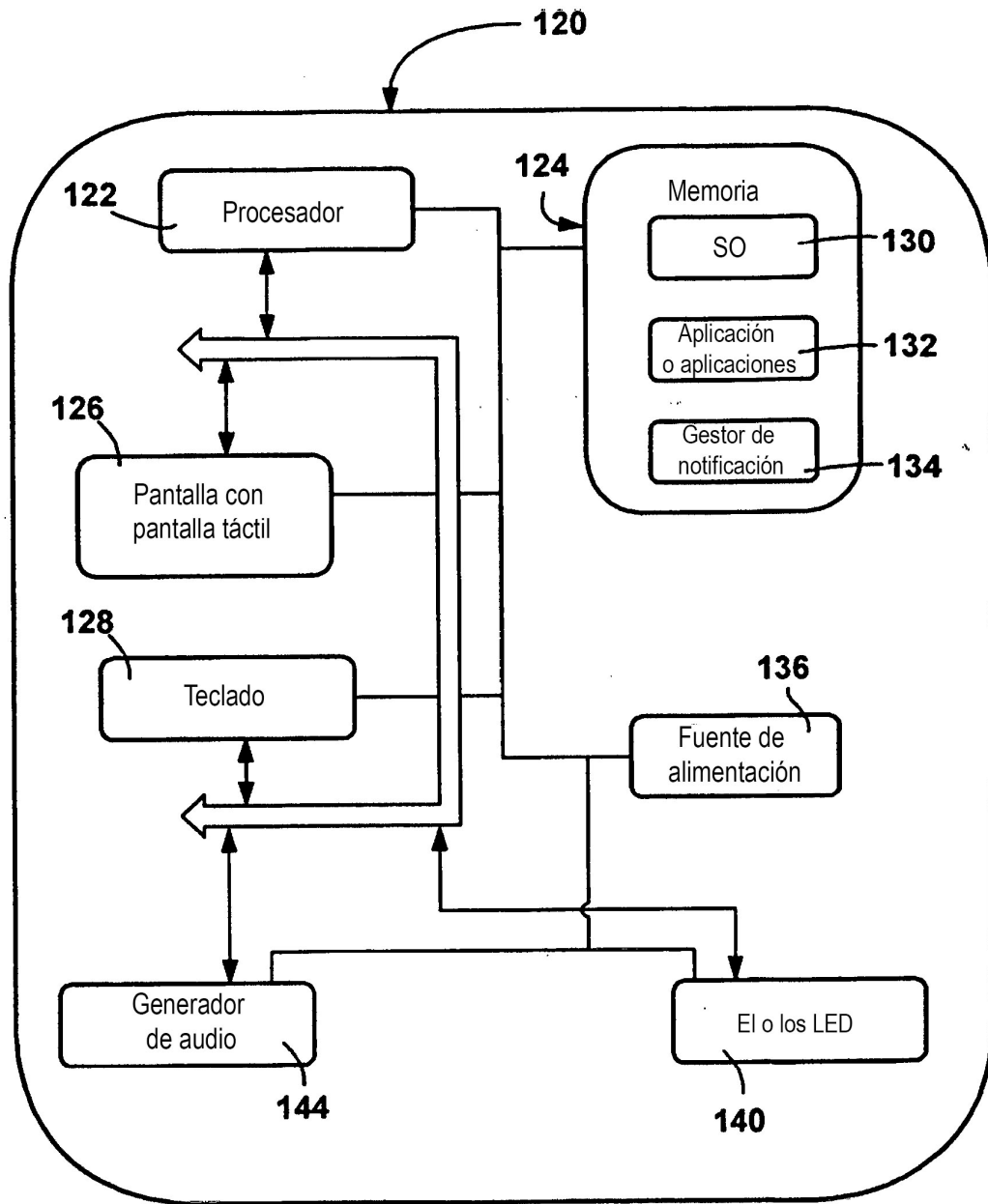


FIG. 1

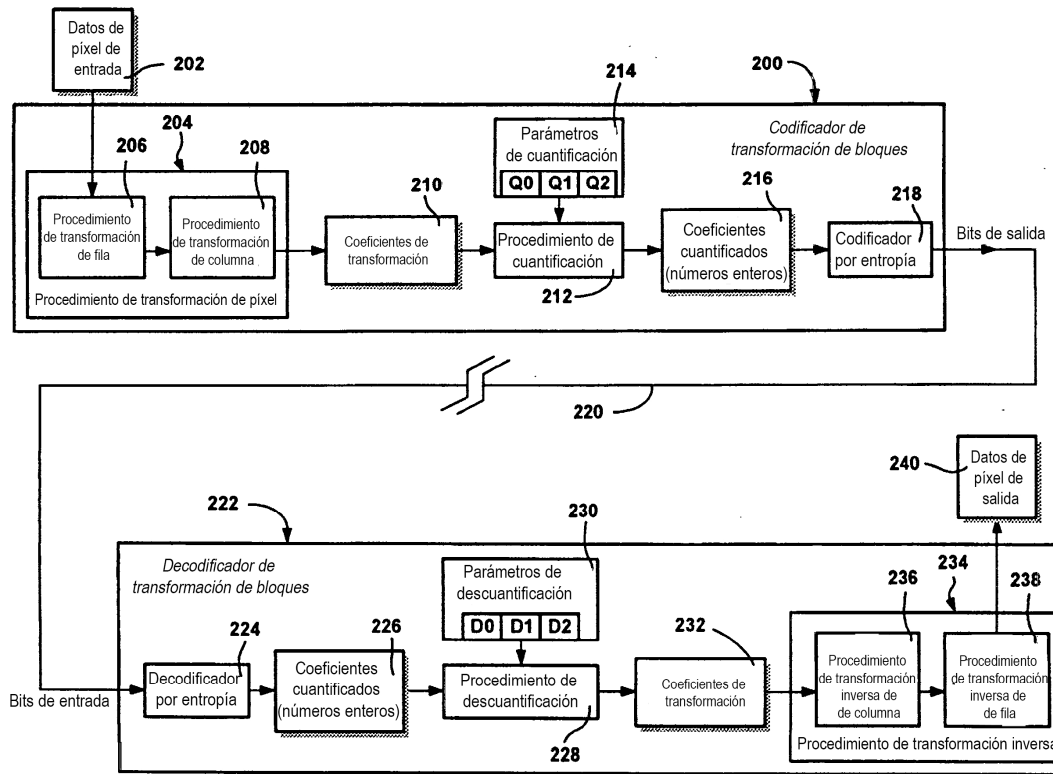


FIG. 2

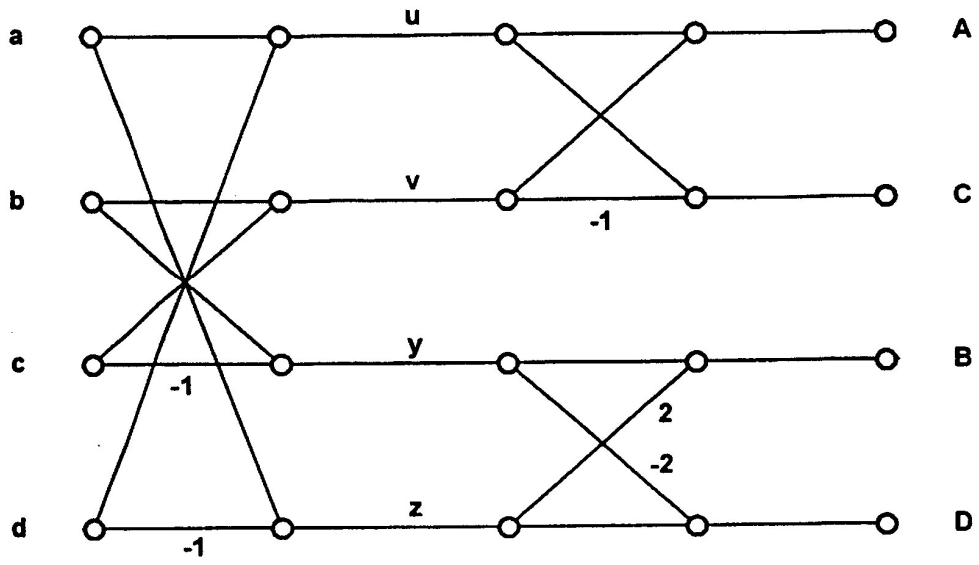


FIG. 3

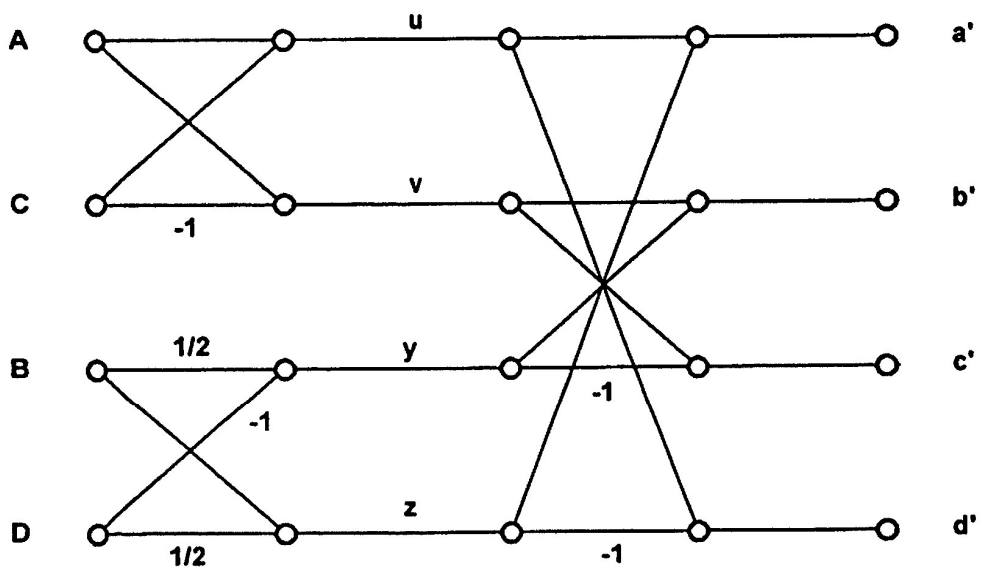


FIG. 4