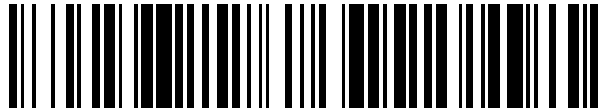


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 764 650**

51 Int. Cl.:

H04N 19/18 (2014.01)

H04N 19/48 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **26.02.2016 PCT/GB2016/050512**

87 Fecha y número de publicación internacional: **01.09.2016 WO16135510**

96 Fecha de presentación y número de la solicitud europea: **26.02.2016 E 16710483 (5)**

97 Fecha y número de publicación de la concesión europea: **20.11.2019 EP 3262835**

54 Título: **Compresión y descompresión de datos de imágenes mediante algoritmo de matriz de tamaño minimizado**

30 Prioridad:

27.02.2015 GB 201503433

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

04.06.2020

73 Titular/es:

**SHEFFIELD HALLAM UNIVERSITY (100.0%)
City Campus Howard Street Sheffield
South Yorkshire S1 1WB, GB**

72 Inventor/es:

**RODRIGUES, MARCOS y
SIDDEQ, MOHAMMED**

74 Agente/Representante:

PONS ARIÑO, Ángel

ES 2 764 650 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

Compresión y descompresión de datos de imágenes mediante algoritmo de matriz de tamaño minimizado

5 La presente invención se refiere a la compresión de datos y, en particular, a los procedimientos de procesamiento de datos y al aparato de procesamiento de datos para comprimir y/o descomprimir datos de imagen, incluidos datos de imágenes fijas y datos de imágenes de vídeo.

10 Se conocen varios procedimientos de compresión de datos y son particularmente útiles para archivos de datos grandes, como archivos de datos de imágenes. La compresión de datos de imagen se ha vuelto más importante en aplicaciones multimedia. Si bien el coste del almacenamiento de datos ha disminuido, la cantidad de datos de imagen y datos de vídeo generados y que requieren almacenamiento ha aumentado considerablemente, como es particularmente evidente para grandes repositorios de datos como YouTube y servicios de almacenamiento en la nube. Con el creciente crecimiento del tráfico de red y los requisitos de almacenamiento, los procedimientos más eficientes para comprimir datos de imágenes fijas y datos de imágenes de vídeo, al tiempo que conservan una alta calidad de imagen reconstruida, serían particularmente beneficiosos con una reducción en el tamaño de los datos comprimidos.

15 La transformada de coseno discreta (DCT) se ha utilizado ampliamente en la compresión de imágenes. La imagen se divide en segmentos y cada segmento está sujeto a la transformada de coseno discreta, que crea una serie de componentes de frecuencia que corresponden a distintos niveles de detalle de la imagen. Se pueden aplicar varias formas de codificación para almacenar solo los coeficientes DCT que se consideran importantes. Esta estrategia general se utiliza en el formato de archivo JPEG para imágenes fijas, y muchos procedimientos de compresión de vídeo y aplicaciones multimedia también se basan generalmente en él.

20 Sin embargo, la compresión JPEG usando DCT es con pérdidas y a menudo introduce artefactos de compresión en los datos de la imagen que son visibles en la imagen generada a partir de los datos de la imagen descomprimida. Además, cuando se aplica a archivos de imagen grandes, el archivo de imagen comprimido resultante puede ser bastante grande. Por ejemplo, el algoritmo JPEG puede comprimir una imagen de 9 Mbytes en un mínimo de 50 KB, y no menos que eso, y con una degradación notable en la imagen descomprimida.

30 Un paso más allá de JPEG es JPEG2000, que se basa en la transformación de ondas pequeñas discreta (DWT), que es una herramienta matemática para funciones de descomposición jerárquica. La compresión de imágenes usando las transformaciones de ondas pequeñas es un procedimiento poderoso que se puede preferir para comprimir imágenes a relaciones de compresión más altas con valores más altos de proporción máxima de señal a ruido (PSNR). Su superioridad frente a JPEG para lograr una alta relación de compresión, resistencia a errores y otras características ha llevado a la ISO JPEG2000. El códec JPEG2000 es más eficiente que su predecesor, JPEG, y supera muchos de los inconvenientes de JPEG. JPEG2000 también ofrece una mayor flexibilidad en comparación con muchos otros códecs como región de interés, alto rango dinámico de valores de intensidad, compresión multicomponente, con pérdida y sin pérdida, computación eficiente y control de la tasa de compresión.

40 La robustez de JPEG2000 deriva de su utilización de DWT para codificar datos de imagen. DWT muestra una alta eficacia en la compresión de imágenes debido a su soporte de representación de resolución múltiple en los dominios espacial y de frecuencia. Además, DWT también puede soportar la transmisión progresiva de imágenes y la codificación de la región de interés.

45 Aunque por sí solo JPEG2000 es una mejor técnica de compresión de datos de imagen que JPEG, la compresión de vídeo a menudo se realiza usando JPEG ya que JPEG2000 tiende a dar lugar a desenfoque de imagen cuando se aplica al vídeo y, por lo tanto, tiende a evitarse. JPEG no da lugar a imágenes borrosas en el vídeo y, por lo tanto, a menudo se prefiere.

50 Además, en algunas aplicaciones puede ser deseable poder comprimir datos de imágenes en 3D. Si bien la geometría y la conectividad de una malla 3D pueden abordarse mediante una serie de técnicas, como la interpolación polinómica de alto grado o las ecuaciones diferenciales parciales, el tema de la compresión eficiente de imágenes 2D tanto para la reconstrucción 3D como para el mapeo de texturas para aplicaciones 3D de luz estructurada, no se ha abordado.

55 Además, en muchas aplicaciones, puede ser deseable transmitir modelos 3D a través de una red de comunicaciones, como Internet, por ejemplo, para compartir modelos CAD/-CAM, actualizar contenido para aplicaciones de entretenimiento o apoyar el diseño colaborativo, análisis y visualización de conjuntos de datos de ingeniería, médicos y científicos. Sin embargo, el ancho de banda impone límites firmes en la cantidad de transmisión de datos y, junto

con los costes de almacenamiento, limitan la complejidad de los modelos 3D que se pueden transmitir a través de Internet y otros entornos de red.

5 MOHAMMED MUSTAFA SIDDEQ, «Using Two Levels DWT with Limited Sequential Search Algorithm for Image
Compression», JOURNAL OF SIGNAL AND INFORMATION PROCESSING, (2012-01-01), vol. 03, no. 01, ISSN 2159-
4465, pages 51 - 62, XP055268961, describe un algoritmo de compresión de imagen basado en dos niveles de
transformación de ondas pequeñas discreta (DWT). La subbanda de baja frecuencia se minimiza utilizando un
algoritmo DCT y un algoritmo Minimize-Matrix-Size que convierte los coeficientes AC en una matriz de valores reales
y los coeficientes DC se almacenan en una columna DC. La columna DC se transforma usando una DCT
10 unidimensional en una matriz T, y luego la matriz T se comprime usando RLE y codificación aritmética. Las subbandas
de alta frecuencia se comprimen mediante una técnica de «eliminar ceros y almacenar datos» (EZSD) que elimina
cada submatriz de 8x8 que contiene ceros y almacena datos distintos de cero en una matriz.

15 Mohammed Mustafa Siddeq, «Applied two Stages Minimize-Matrix-Size Algorithm with DCT on DWT for Image
Compression», UK Journal of Information and Computing Science, (2012-01-01), vol. 7, no. 1, pages 37 - 53, describe
un algoritmo para la compresión de imágenes basado en un algoritmo Minimize-Matrix-Size (MMS) para codificación
y un algoritmo de búsqueda secuencial limitada (LSS) para decodificación. Se utiliza una transformación de ondas
pequeñas discreta (DWT) para descomponer la imagen en una subbanda de baja frecuencia y tres subbandas de alta
frecuencia. Cada valor de nxn de la subbanda de baja frecuencia (LL) se transforma mediante un DCT bidimensional
20 y los coeficientes DC se almacenan en una matriz DC y los coeficientes AC se almacenan en una matriz AC. El
algoritmo MMS se aplica luego a la matriz de AC para convertir cada grupo de coeficientes de AC en un único valor
de coma flotante. La matriz DC es transformada nuevamente por DWT y se aplica el algoritmo MMS. El algoritmo LLS
se usa durante la decodificación para estimar los valores originales de la matriz DC y la matriz AC usando un
procedimiento iterativo que depende de la probabilidad de los datos de la matriz AC y la matriz DC.

25 Por lo tanto, existe la necesidad de técnicas de compresión de datos de imagen que no introduzcan artefactos de
compresión importante y/o produzcan archivos de datos de imagen comprimidos de pequeño tamaño. Sería
especialmente beneficioso si la misma técnica fuera aplicable tanto a los datos de imágenes fijas como de vídeo y
también fuera adecuada para los datos de imágenes en 3D.

30 La presente invención proporciona un procedimiento para la compresión de imágenes basado en DCT, pero usando
un proceso de cuantificación que crea dos matrices de datos. Una de estas matrices se contrae utilizando un algoritmo
de minimización del tamaño de la matriz. Este procedimiento logra una relación de compresión de imagen eficiente y
una precisión superior de reconstrucción de imágenes 2D y 3D, en comparación con JPEG estándar y JPEG2000, a
35 la vez que es adecuado tanto para compresión de imágenes fijas como de vídeo.

Un primer aspecto de la invención proporciona un procedimiento implementado por ordenador para comprimir datos
de imagen como se define en la reivindicación 1.

40 Al separar los datos que representan las características más importantes de la imagen (los coeficientes DC DCT) y
los datos que representan las características menos importantes de la imagen (los coeficientes AC DCT), y procesarlos
por separado, se puede obtener una relación de compresión alta mientras se evitan los artefactos notables de
compresión de imagen en la imagen descomprimida.

45 La cuantificación de todos los conjuntos de coeficientes DCT puede incluir la aplicación de un factor de cuantificación
que da como resultado al menos un cuarto, al menos la mitad o aproximadamente tres cuartos de los coeficientes
DCT con un valor inferior a 1 y preferiblemente aproximadamente cero. La aplicación del factor de cuantificación puede
implicar dividir los coeficientes DCT de cada conjunto de coeficientes DCT por un factor de cuantificación que sea
mayor que al menos dos. El valor del factor de cuantificación puede depender del tamaño del bloque de píxeles.

50 El procedimiento puede comprender además comprimir la matriz DC para formar una matriz DC comprimida, en cuyo
caso la matriz DC comprimida se codifica posteriormente aritméticamente para formar los datos codificados
aritméticamente.

55 La compresión de la matriz DC puede incluir la aplicación de un proceso para hacer que los valores de los elementos
de la matriz DC comprimida sean más similares y/o aparezcan en la matriz DC comprimida con una frecuencia más
alta, en comparación con los elementos de la matriz DC. La compresión de la matriz DC puede incluir el cálculo de
una diferencia entre los valores de los elementos de la matriz DC para formar la matriz DC comprimida.

60 Cada componente clave puede tener un valor distinto. La clave puede comprender tres componentes clave, o un
número mayor o menor de componentes clave. Un mayor número de componentes puede ralentizar el proceso de
descompresión. Por lo tanto, existe un equilibrio entre aumentar la eficiencia de compresión mediante el uso de un

mayor número de componentes clave y no ralentizar el proceso de descompresión. El número de componentes clave puede estar en el rango de tres a ocho.

5 La compresión de la matriz AC reducida puede comprender además formar una primera matriz AC codificada que incluye los elementos distintos de cero de la matriz AC codificada y formar una segunda matriz AC codificada que incluye datos que codifican los elementos cero de la matriz AC codificada. Los datos que codifican los elementos cero de la matriz codificada pueden incluir datos que indican el número de ceros entre elementos adyacentes distintos de cero y/o datos que indican la posición de elementos distintos de cero. La segunda matriz AC codificada puede codificarse adicionalmente para aumentar la frecuencia de aparición de valores de elementos de una tercera matriz
10 AC codificada.

La formación de la matriz de datos limitada también puede incluir la determinación de la frecuencia de aparición de cada valle único en la matriz AC.

15 La clave y/o los datos limitados pueden almacenarse por separado y asociarse con el archivo de imagen comprimido. La clave y/o datos limitados pueden almacenarse en el archivo de imagen comprimido, por ejemplo, en un encabezado del archivo de imagen comprimido.

20 Los datos del encabezado se pueden separar de los datos comprimidos para crear efectivamente una versión criptográfica de los datos comprimidos. Un procedimiento de algoritmo de descompresión sería incapaz de descomprimir los datos sin los valores clave utilizados para comprimir los datos.

25 Cada uno de la pluralidad de bloques de píxeles no superpuestos puede tener un tamaño de $n \times n$. El valor de n puede ser al menos 4 y preferiblemente al menos 8. El valor de n puede seleccionarse de entre 4, 8, 16, 32, 64, 128, 256. El valor de n puede estar en el intervalo de 4 a 256.

30 La imagen puede ser una imagen en color y puede incluir una pluralidad de componentes de color. El procedimiento puede aplicarse para cada uno de los componentes de color. Los componentes de color pueden ser RGB o YCrCb o HSV o cualquier otro espacio de color generalmente conocido.

35 El procedimiento puede comprender además almacenar el archivo de imagen comprimido localmente en un primer dispositivo de procesamiento de datos y/o transmitir el archivo de imagen comprimido a una red de comunicaciones para su almacenamiento en un segundo dispositivo de procesamiento de datos remoto. El procedimiento puede incluir además generar los datos de imagen en el primer dispositivo de procesamiento de datos. Los datos de la imagen pueden corresponder a una fotografía, vídeo, gráfico o modelo 3D generado utilizando el primer dispositivo de procesamiento de datos. Los datos de imagen pueden ser generados por un dispositivo de captura de imágenes como una cámara o un escáner.

40 Un segundo aspecto de la invención proporciona un procedimiento implementado por ordenador para descomprimir datos de imagen como se define en la reivindicación 7.

45 El procedimiento puede comprender además descomprimir la matriz DC para formar una matriz DC adicional en la que los elementos son el conjunto de coeficientes DC DCT cuantificados. La descompresión de la matriz DC puede incluir el cálculo de un total entre los valores de los elementos de la matriz DC y el almacenamiento de los resultados en la matriz DC adicional.

50 Los valores de todas las combinaciones posibles de los elementos de la matriz de datos limitada cuando se combinan con la clave pueden calcularse y pueden almacenarse en una matriz de combinaciones de clave de datos limitada. Se puede aplicar un procedimiento de búsqueda a la matriz de combinaciones de claves de datos limitadas utilizando los valores de los elementos de la matriz AC codificada para encontrar valores coincidentes. Se puede determinar la combinación de elementos de la matriz limitada correspondiente a cada valor coincidente. La matriz AC reducida puede formarse a partir de cada combinación determinada de elementos del conjunto limitado.

55 El procedimiento puede comprender además clasificar la matriz de combinaciones de claves de datos limitadas por orden de tamaño para formar una matriz de combinaciones de claves de datos limitada clasificada. La clasificación puede ser de tamaño ascendente o descendente.

El procedimiento de búsqueda puede ser un procedimiento de búsqueda paralelo.

60 El procedimiento de búsqueda puede implementar un algoritmo de búsqueda binario.

La clave puede comprender tres o al menos tres componentes clave. Cada componente clave puede tener un tamaño

distinto.

El archivo de imagen comprimido puede incluir la clave y/o una matriz de datos limitada y pueden estar en un encabezado del archivo de imagen comprimido.

5

La clave y/o una matriz de datos limitada pueden asociarse con el archivo de imagen comprimido y almacenarse por separado de los datos codificados aritméticamente.

Cada uno de la pluralidad de bloques de píxeles no superpuestos puede tener un tamaño de nxn y n puede ser al menos 8. El valor de n puede seleccionarse entre 8, 16, 32, 64, 128, 256.

10

El procedimiento puede comprender además recuperar el archivo de imagen comprimido del almacenamiento local en un segundo dispositivo de procesamiento de datos y/o recibir el archivo de imagen comprimido a través de una red de comunicaciones desde un primer dispositivo de procesamiento de datos remoto.

15

La trama de datos de imagen puede ser una imagen 2D.

La trama de datos de imagen puede ser una imagen 3D. El archivo de imagen comprimido puede tener datos auxiliares asociados o almacenados en él y que permiten la reconstrucción y/o visualización de la imagen 3D.

20

La trama de datos de imagen puede ser una imagen fija o estática o una parte de un vídeo o una animación.

Un segundo aspecto de la invención proporciona un procedimiento implementado por ordenador para manejar datos de imagen como se define en la reivindicación 15.

25

El procedimiento puede comprender además transmitir el archivo de imagen comprimido desde un primer dispositivo de procesamiento de datos a un segundo dispositivo de procesamiento de datos antes de descomprimir el archivo de imagen comprimido. El archivo de imagen puede transmitirse a través de una red de área amplia o una red de área local.

30

El segundo dispositivo de procesamiento de datos puede descomprimir y mostrar el archivo de imagen comprimido o mostrarlo de otro modo.

El procedimiento puede comprender además almacenar el archivo de imagen comprimido en el segundo dispositivo de procesamiento de datos. El procedimiento puede comprender además transmitir el archivo de imagen comprimido desde el segundo dispositivo de procesamiento de datos a un tercer dispositivo de procesamiento de datos antes de descomprimir el archivo de imagen comprimido. El tercer dispositivo de procesamiento de datos puede descomprimir y/o mostrar el archivo de imagen comprimido o mostrarlo de otro modo. El segundo dispositivo de procesamiento de datos puede servir como un servidor de datos de imágenes, como un servidor de archivos de fotografía o vídeo. El primer y/o tercer dispositivo de procesamiento de datos pueden ser cada uno dispositivos del cliente.

35

40

Un cuarto aspecto de la invención proporciona un medio legible por ordenador que almacena instrucciones ejecutables por ordenador tal como se define en la reivindicación 16.

45

Un quinto aspecto de la invención proporciona un dispositivo de procesamiento de datos como se define en la reivindicación 17.

A continuación se describirá en detalle una realización de la invención, solo a modo de ejemplo, y con referencia a los dibujos adjuntos, en los que:

50

la figura 1 muestra un diagrama de flujo del proceso que ilustra un procedimiento general de procesamiento de imágenes según la invención e incluye etapas de compresión y descompresión de imágenes, cada una según la invención;

55

la figura 2 muestra un diagrama de flujo del proceso que ilustra un procedimiento de compresión de imagen según la invención y que corresponde al paso de compresión de imagen de la figura 1;

la figura 3 muestra una representación gráfica de una imagen que comprende una matriz de píxeles representados por datos de imagen;

60

la figura 4 muestra una representación gráfica del resultado de aplicar la transformación de coseno discreta a un bloque de píxeles nxn de los datos de imagen ilustrados en la figura 3 y la separación en una matriz de coeficientes

DC DCT y una matriz de coeficientes AC DCT;

la figura 5 muestra un diagrama de flujo del proceso que ilustra un algoritmo para comprimir la matriz de coeficientes AC DCT;

5

la figura 6 muestra una representación gráfica que ilustra el procedimiento de la figura 5 de comprimir la matriz de coeficientes AC DCT en una matriz;

la figura 7 muestra un diagrama de flujo del proceso que ilustra un procedimiento de compresión de matriz utilizado en el algoritmo ilustrado en la figura 5;

10

la figura 8 muestra una representación gráfica que ilustra un paso de compresión basado en clave del procedimiento de la figura 7;

la figura 9 muestra una representación gráfica que ilustra un paso de compresión adicional de eliminación de ceros del procedimiento de la figura 7;

15

la figura 10 muestra una representación gráfica de la estructura de datos para un archivo de datos de imagen comprimido;

20

La figura 11 muestra un diagrama de flujo del proceso que ilustra un procedimiento de descompresión de imagen según la invención y que corresponde al paso de descompresión de imagen de la figura 1;

La figura 12 muestra una representación gráfica de un paso de descompresión del coeficiente DC DCT de la figura 11;

25

la figura 13 muestra un diagrama de flujo del proceso que ilustra un procedimiento de descompresión del coeficiente AC DCT y que corresponde a un paso de descompresión del coeficiente AC DCT del procedimiento de la figura 11;

la figura 14 muestra una representación gráfica de una primera parte del procedimiento de la figura 13 para decodificar los datos AC DCT;

30

la figura 15 muestra un diagrama de flujo del proceso que ilustra un procedimiento de decodificación de datos AC DCT utilizado en el procedimiento de la figura 13;

35

la figura 16 muestra una representación gráfica del procedimiento de la figura 15 para decodificar los datos AC DCT;

la figura 17 muestra una representación gráfica de la cuantificación inversa y las porciones DCT inversas del procedimiento de descompresión de la figura 11;

40

la figura 18A muestra un ejemplo del resultado de aplicar el procedimiento de la invención para comprimir y descomprimir un archivo de imagen de rayos X;

la figura 18B muestra un ejemplo del resultado de aplicar el algoritmo JPEG para comprimir y descomprimir el mismo archivo de imagen de rayos X que el utilizado para la figura 18A al mismo tamaño;

45

la figura 18C muestra un ejemplo del resultado de aplicar el algoritmo JPEG2000 para comprimir y descomprimir el mismo archivo de imagen de rayos X que el utilizado para la figura 18A al mismo tamaño; y

la figura 19 muestra un diagrama de bloques esquemático de un dispositivo de procesamiento de datos según la invención configurado para extraer los procedimientos de compresión y/o descompresión de la invención.

50

Elementos similares en las distintas figuras comparten signos de referencia comunes a menos que se indique lo contrario.

55

La figura 1 muestra un diagrama de flujo que ilustra un procedimiento de procesamiento de datos de imagen general 100 según la invención y en el que pueden usarse los aspectos del procedimiento de compresión de imagen y del procedimiento de descompresión de imagen de la invención. Como se discute en mayor detalle a continuación, los procedimientos de compresión y descompresión de imágenes de la invención son aplicables a cualquier tipo de datos de imágenes que se puedan representar por color de píxel o componentes en escala de grises. En particular, la invención es aplicable a imágenes 2D tanto fijas como de vídeo. Además, la invención se puede aplicar a imágenes 3D. En ese caso, se pueden usar algunos datos auxiliares además de los datos de la imagen comprimida para

60

reconstruir la imagen 3D al descomprimir los datos de la imagen.

Sin embargo, la creación de datos de imágenes en 3D y la reconstrucción de los mismos es generalmente conocida en la técnica y se describe, por ejemplo, en: Jason Geng, Structured-light 3D surface imaging: a tutorial, *Advances in Optics and Photonics*, Vol 3, Issue 2, pp 128-160, 2011; MAIER-HEIN, L., MOUNTNEY, P., BARTOLI, A., ELHAWARY, H., ELSON, D., GROCH, A., KOLB, A., RODRIGUES, Marcos, SORGER, J., SPEIDEL, Suzanne and STOYANOV, D. (2013), Optical techniques for 3D surface reconstruction in computer-assisted laparoscopic surgery, *Medical Image Analysis*, 17 (8), 974-996; RODRIGUES, Marcos, KORMANN, Mariza, SCHUHLER, C and TOMEK, P (2013), Structured light techniques for 3D surface reconstruction in robotic tasks, in KACPRZYK, J, (ed.) *Advances in Intelligent Systems and Computing*, Heidelberg, Springer, 805-814; RODRIGUES, Marcos (2012) Fast 3D reconstruction with single shot technology: engineering and computing challenges, in *First national Conference on Technology and Management*, Malabe, Sri Lanka, 27 January, 2012, Malabe, Sri Lanka, Sri Lanka Institute of Information Technology; and RODRIGUES, Marcos, OSMAN, Abduslam and ROBINSON, Alan (2013), Partial differential equations for 3D Data compression and Reconstruction, *ADSA Advances in Dynamical Systems and Applications*, 8 (2), 303-315.

El método de procesamiento de datos de imagen generalmente comienza desde la etapa 102 de creación de imagen en el que se crean los datos de imagen. En algunas realizaciones, esto puede ser usando una cámara de acero o vídeo para capturar datos de imágenes. En otras realizaciones, los datos de imagen pueden surgir del escaneo de un elemento del mundo real. Alternativamente, en otras realizaciones, la creación de imágenes puede ser un proceso de modelado, por ejemplo, usando un sistema CAD, o usando un dibujo o una aplicación similar para crear los datos de la imagen. Independientemente de cómo se crean los datos de imagen en el paso 102, en el paso 104, los datos de imagen se comprimen usando el aspecto del procedimiento de compresión de imagen de la invención como se describe con mayor detalle a continuación. Después de comprimir los datos de la imagen, los datos de la imagen pueden almacenarse localmente o transmitirse y almacenarse de forma remota. Por ejemplo, en algunas realizaciones, el procedimiento de procesamiento de imágenes 100 puede funcionar todo, por ejemplo, en un teléfono inteligente para aumentar el número de imágenes que se pueden almacenar localmente en el teléfono inteligente desde una cámara incorporada. En otras realizaciones, los datos de imagen comprimidos pueden transmitirse a un sitio remoto para su almacenamiento. Por ejemplo, los datos de la imagen pueden ser sitios web para compartir fotos o un dispositivo de almacenamiento en la nube para almacenamiento remoto. En otras realizaciones adicionales, los datos de imagen pueden transmitirse a un dispositivo de procesamiento de datos adicional para su visualización y revisión. Por ejemplo, si la imagen comprende datos de imágenes de rayos X, entonces el dispositivo remoto puede ser un dispositivo digital de visualización de rayos X y los datos de imágenes comprimidas pueden enviarse a través de la red. Obviamente, los datos de la imagen comprimida pueden enviarse tanto a un repositorio para almacenamiento como a un dispositivo de visualización.

Cuando se desea ver los datos de la imagen, en la etapa 108, el procedimiento de descompresión de imagen de la invención se aplica a los datos de imagen almacenados. El formato del archivo que se almacena y/o transmite en el paso 106 se describe con mayor detalle a continuación con referencia a la figura 10.

Después de que la imagen se haya descomprimido en 108, la imagen se puede mostrar en la etapa 110. Por ejemplo, esto puede comprender simplemente recuperar una fotografía almacenada localmente de la memoria en un teléfono inteligente y mostrarla a un usuario. En otras realizaciones, puede comprender mostrar un modelo generado local o remotamente en un sistema CAD y/o CAM. En otras realizaciones, puede comprender mostrar una imagen de rayos X a un profesional médico. En otras realizaciones adicionales, puede comprender mostrar una imagen 3D, usando los datos de reconstrucción de imagen 3D asociados, a un usuario. En otras realizaciones, puede comprender mostrar un vídeo a un usuario en el que los datos de imagen se transmiten desde una fuente remota descomprimida en 108 y visualizada en 110 en un reproductor de vídeo.

Será evidente que, por lo tanto, la invención es aplicable en una amplia gama de campos y, en particular, durante datos de imagen en 2D, 3D y vídeo. Como se expone a continuación, los procedimientos de compresión y descompresión de la invención proporcionan relaciones eficientes de compresión de imagen y descompresión sin introducir artefactos de imagen importantes. Además, es adecuado tanto para imágenes fijas como para vídeos. Por lo tanto, la invención tiene una amplia gama de aplicabilidad y numerosos beneficios como se expone con más detalle a continuación. Por ejemplo, la invención puede aplicarse a aplicaciones de compresión y transmisión de vídeo. Otras aplicaciones se relacionan con la reconstrucción de superficies utilizando técnicas ópticas que utilizan luz blanca estructurada o infrarrojo cercano. Este enfoque se puede aplicar a aplicaciones como seguridad, gráficos y animación multimedia, descripciones geométricas de software de modelado 3D, verificación de geometría de formas en procesos industriales, prótesis y rehabilitación médica, ingeniería inversa y archivo y muchos más.

La figura 2 muestra un diagrama de flujo del proceso que ilustra un procedimiento de compresión de imagen según la invención y que corresponde generalmente a la etapa 104 de la figura 1. Inicialmente, en la etapa 202, la transformación discreta del coseno (DCT) se aplica a la imagen original. La figura 3 ilustra una representación gráfica

de una imagen 300 en forma de una matriz rectangular de píxeles individuales, por ejemplo, una matriz de 1280x1080 píxeles. Para una imagen en escala de grises, cada píxel está representado por un único elemento de intensidad de datos de escala de grises, por ejemplo, en el rango de 0 (correspondiente al negro) a 255 (correspondiente al blanco).

5 Para una imagen en color, cada píxel está representado por tres elementos de datos de color, que dependerán del espacio de color utilizado para representar la imagen en color. Los espacios de color comunes incluyen RGB, YCrCb y HSV, aunque generalmente se conocen otros. En el siguiente ejemplo, se supone un espacio de color RGB pero la invención no se limita a ese espacio de color específico. La figura 3 ilustra un solo componente de color y, por lo tanto, para una imagen en color y un espacio de color RGB, la imagen en color se separa en sus tres canales de color, rojo, verde y azul, y los datos de color para cada canal de color se procesan por separado.

10 En primer lugar, para cada canal de color, la imagen 2D 300 se divide en bloques no superpuestos, por ejemplo, 302, de $n \times n$ píxeles, por ejemplo, píxel 304. En el ejemplo ilustrado, el bloque 302 comprende 8×8 píxeles y una característica de la invención es que usa bloques grandes de píxeles, en comparación con JPEG, que típicamente usa bloques de píxeles más pequeños, por ejemplo, 2×2 , para su etapa DCT. De hecho, la invención generalmente puede usar bloques de píxeles con $n \geq 4$, por ejemplo, $n = 4, 8, 16, 32, 64$, etc. Por lo tanto, en el ejemplo actual, la imagen 15 300 se divide en 160×135 bloques 302 cada uno de 8×8 píxeles. En 210, cada bloque de píxeles 302, se transforma aplicando la DCT para producir coeficientes DCT descorrelacionados. La transformación produce el mismo número de coeficientes DCT que píxeles en el bloque, en su ejemplo 64.

20 En el caso de que el tamaño de la imagen no sea un múltiplo entero del tamaño del bloque de píxeles, los límites de la imagen se rellenan con ceros. Por ejemplo, para un tamaño de imagen de 1024×1024 , y si la imagen se divide usando bloques de 16×16 píxeles, los bloques de píxeles no cruzan los límites de la imagen. Sin embargo, para un tamaño de imagen de 1030×1030 y bloques de 16×16 píxeles, en este caso el procedimiento rellena ceros alrededor de la imagen para que correspondan a un múltiplo entero de bloques de píxeles, en este ejemplo 65×65 .

25 La figura cuatro proporciona una representación gráfica 310 de los 64 coeficientes DCT producidos mediante la aplicación de DCT. Cuando se transforma en el dominio de frecuencia, cada bloque de píxeles consiste en un valor de coeficiente DC en la primera ubicación 312 y otros valores de coeficiente AC 314, 316 (que suelen ser mucho más pequeños que el coeficiente DC 312), como se describe en la ecuación (1)

30

$$C(u, v) = a(u)a(v) \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2n} \right] \cos \left[\frac{(2y+1)v\pi}{2n} \right] \quad (1)$$

$$\text{donde } a(u) = \sqrt{\frac{1}{n}}, \text{ para } u = 0, a(u) = \sqrt{\frac{2}{n}}, \text{ para } u \neq 0$$

35 Y u y v denotan los componentes horizontal y vertical, respectivamente.

En el contexto de DCT, DC corresponde al valor medio de una señal, por lo que los componentes o coeficientes DC están relacionados con la intensidad de brillo promedio de cada bloque. En contraste con esto, todos los demás valores corresponden a variaciones superpuestas en la señal y, por lo tanto, se denominan AC. Por lo tanto, DC y AC se usan 40 para referirse al valor medio o promedio y las variaciones respectivamente.

Luego, en el paso 230, se cuantifican los coeficientes DCT para cada bloque, por ejemplo, el bloque 310. Cada bloque 310 de $n \times n$ coeficientes DCT se cuantifican usando una tabla de cuantificación, que se puede representar mediante la siguiente ecuación (2).

45

$$Q(i, j) = \text{Factor} * (i + j) \quad (2)$$

donde $i, j = 1, 2, \dots, n$, $\text{Factor} = 2, 3, \dots$ etc.

50 Cada $n \times n$ bloque 310 se cuantifica por «Q» usando una matriz de división de puntos que trunca los resultados. Este proceso 230 elimina coeficientes DCT insignificantes y aumenta el número de ceros en cada bloque 310. En la ecuación (2), el parámetro «Factor» se utiliza para aumentar o disminuir los valores de la «Q». Por lo tanto, los detalles de la imagen se reducen cuando $\text{Factor} > 2$. Para el coeficiente DC, $i=1$ y $j=1$ y, por lo tanto, Q tiene su valor más bajo y, por lo tanto, en la división por Q, el coeficiente DC se reduce proporcionalmente menos que los coeficientes AC.

55 Por ejemplo, para el último coeficiente AC 318, $i=8$ y $j=8$, y por lo tanto Q tiene su mayor valor. No hay un límite particular en el rango de Factor, porque su valor depende de los coeficientes DCT o la resolución de la imagen.

Después de la cuantificación de los coeficientes DCT, se forma una matriz 320 de los coeficientes DCT DC cuantificados. Como se ilustra en la línea 326 de la figura 4, el primer elemento 322 de la matriz 320 es el coeficiente DCT DC cuantificado para el primer bloque de píxeles. El segundo elemento 324 de la matriz 320 es el coeficiente DC DCT cuantificado para el segundo bloque de píxeles 306, etc. Como en este ejemplo, hay 21.600 bloques de píxeles, la matriz de coeficientes DC DCT cuantificados 320 tiene 21.600 elementos. La matriz de coeficientes DC DCT 320 se comprime entonces a 240 para formar una matriz DC comprimida 330. En particular, la diferencia se calcula entre dos valores adyacentes de la matriz 320, por ejemplo, 32 y 324, y el resultado se almacena como un elemento 332 en la matriz DC comprimida 330, como se ilustra con la flecha 334. Este proceso de compresión se conoce como proceso *DBV* o *diferencia entre dos valores*. Los coeficientes cuantificados DC DCT en la matriz 320 están correlacionados y generalmente sus valores son similares. Por lo tanto, sus diferencias, es decir, los elementos de la matriz comprimida 330, son pequeños y muchos de los datos en la matriz comprimida 330 se repetirán, lo que es más fácil de comprimir. La ecuación (3) describe este proceso DBV

$$D(i) = D(i) - D(i + 1) \quad (3)$$

donde $i=1,2,3, \dots, m-1$ y m es el tamaño original de la matriz DC 320, que en este ejemplo es 21.600.

Por ejemplo, la aplicación de (3) a la matriz DC de ejemplo de [13, 13, 15, 14, 14, 13, 15, 16, 17, 13, 14, 14] da como resultado una matriz DC comprimida de [0, -1, 1, 0, 1, -2, -1, -1, 4, -1, 0, 14]. Por lo tanto, el conjunto comprimido 320 tiene una mayor frecuencia de repetición de sus datos y, por lo tanto, es más fácil de comprimir.

Los coeficientes restantes de AC DCT cuantificados para el bloque de píxeles, por ejemplo, 314, 316 y que en este ejemplo totalizan 63 coeficientes, se convierten en una matriz unidimensional, como se ilustra en las líneas 344 y 346 en la figura 4, escaneando columna por columna, para formar una fila 342 de una matriz 340 referida generalmente aquí como la matriz AC. Este proceso se repite a 250 para cada bloque de píxeles hasta que se forma la matriz AC que tiene 21.600 filas, cada una de 63 columnas, y en donde cada elemento de la matriz AC corresponde a un coeficiente cuantificado AC DCT.

En 260, se aplica un proceso a la matriz AC 340 para reducir su tamaño eliminando elementos de datos que tienen un valor de cero. El algoritmo comprime el tamaño de la matriz AC 260 y la convierte en una matriz unidimensional y generalmente se denomina aquí algoritmo MMS o Minimize-Matrix-Size. La matriz AC 340 típicamente contiene muchos elementos de datos con valor cero y algunos elementos de datos que tienen un valor distinto de cero. El algoritmo MMS 260 generalmente elimina bloques de ceros y almacena bloques de datos distintos de cero en una matriz. Este algoritmo comprime la matriz AC 340 y también la convierte en una matriz unidimensional que es susceptible de compresión adicional utilizando un procedimiento de compresión adicional denominado generalmente en este documento como algoritmo *que elimina ceros y almacena datos distintos de cero* o *EZSN*.

En términos generales, el algoritmo MMS divide la matriz AC 340 en bloques de tamaño no superpuestos $K \times K$ donde $K \geq 8$, y luego comienza a buscar bloques de datos distintos de cero, es decir, busca al menos un elemento de datos que tenga un valor distinto de cero dentro de un bloque. Si un bloque contiene un elemento de datos que tiene un valor distinto de cero, entonces todos los elementos de datos en el bloque se almacenan en una matriz generalmente referida aquí como la matriz AC reducida. De lo contrario, el bloque se ignora y el algoritmo continúa buscando otros bloques distintos de cero.

El procedimiento MMS 350 se ilustra en el diagrama de flujo del proceso de la figura 5 y gráficamente en la figura 6. Antes de aplicar el procedimiento de tamaño de matriz minimizado 350, la probabilidad de los elementos de datos de la matriz AC reducida se calcula a partir de los elementos de datos de la matriz AC 340. Cada valor de datos único se determina a partir de todos los elementos de la matriz AC 340 junto con su frecuencia de ocurrencia a partir de la cual se puede determinar la probabilidad de cualquier elemento de datos distinto de cero: por ejemplo, el valor -3 ocurre 13568 veces de un total de 1360800 elementos de datos en la matriz AC 340.

Por ejemplo, si la matriz AC tenía la siguiente forma:

$$\begin{bmatrix} -1 & 0 & 5 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 0 & 2 & 1 \\ -2 & 0 & 2 & -2 \end{bmatrix}$$

Luego, cada valor de datos único se determina a partir de todos los elementos de la matriz para crear la matriz de datos limitada [-1,0,5,1,2,-2] y que tiene una frecuencia de ocurrencia o repetición de [1,7,1,2,3,2]. Estos valores de datos son las probabilidades asociadas a las que generalmente se hace referencia en este documento como datos

limitados y se usan en la etapa de descompresión. Los valores de los datos en la matriz limitada no se repiten ya que los datos limitados representan el espacio buscado por el procedimiento de descompresión para reconstruir la matriz AC original y, por lo tanto, los valores no deberán repetirse. El tamaño de los datos limitados se puede representar en bits, por lo que los datos limitados se almacenan en una parte del encabezado del archivo de imagen comprimido, como se describe con mayor detalle a continuación. Su efecto sobre el tamaño total del archivo comprimido es casi insignificante.

La primera etapa del procedimiento MMS 350 generalmente elimina bloques de datos de la matriz AC 340 que contienen solo valores cero. La matriz de AC 340 se divide en bloques de datos de 8x8, por ejemplo, el primer bloque de datos de 8x8 370. El primer bloque 370 colocado en una posición de una primera fila 354 y una primera columna 356 se evalúa en 358 para determinar si alguno de sus 64 elementos de datos tiene un valor distinto de cero. Si se determina que todos los elementos de datos en el bloque tienen valor cero, entonces el procesamiento pasa a 360 y el bloque de datos se descarta. De lo contrario, en 362, todos los datos en el bloque de datos actual, incluso si la mayoría de sus valores son cero, se agregan en secuencia, fila por fila, a una matriz de datos unidimensionales 380, referida generalmente aquí como la matriz AC reducida. Como se ilustra en la figura 6, el primer elemento de datos del primer bloque 370 se almacena 372 como el primer elemento de la matriz AC reducida 380 y el último elemento de datos del primer bloque 370 se almacena 374 como el 64.º elemento de la matriz AC reducida 380.

En 364, se selecciona un siguiente bloque de datos en la fila actual, en este ejemplo, el bloque de datos 376 y el procesamiento continúa como se describe anteriormente. En este ejemplo, todos los elementos de datos del segundo bloque de datos 376 tienen un valor de cero y, por lo tanto, el segundo bloque de datos 376 se descarta en 360 y ninguno de sus datos se almacena en la matriz AC reducida 380. Los bucles de procesamiento vuelven a 354 y el tercer bloque de datos 378 se evalúa y determina en 358 para incluir algunos datos distintos de cero y, por lo tanto, todos sus elementos de datos se agregan 382 a la matriz AC reducida 380 en 362 y se concatenan con los datos existentes. El proceso se repite hasta que se haya evaluado la primera fila del bloque de datos y luego en 366 el proceso se repite nuevamente para la siguiente fila de bloques de datos y así sucesivamente hasta que se haya evaluado toda la matriz AC 340 y se hayan eliminado todos los bloques de datos cero.

En el pseudocódigo, la parte de eliminar ceros, almacenar no ceros del procedimiento MMS 350 se puede representar de la siguiente manera:

```

K=8; %% tamaño de bloque = [K X K]
  I=1 LOC=1
  Mientras que (I<tamaño de columna para subbanda de alta frecuencia)
    J=1;
    Mientras que (J<tamaño de fila para subbanda de alta frecuencia)
      Block[1..K*K]= Read_Block_from_Matrix(I, J); %% leer un bloque 8x8 de una subbanda de alta
      frecuencia
      IF(Check_Block(Block) == 'nonzero') %% esta función comprueba el contenido «Block» = ¿distinto de
      cero?
      POSTION [LOC] =I; POSTION [LOC+I] =J; %% Ubicación de seguridad para bloques que contienen datos
      distintos de cero
      LOC =LOC+2;
    45 Para n=I: Block_Size * Block_Size
      Reduced Array[P] = Block[n]; %% datos distintos de cero de seguridad en una nueva matriz

```

```

      ++P;
    End;
  End;
  J=J+ K;
End;
  I=I+ K;
End;

```

Después de que la matriz AC 340 haya eliminado sus bloques de datos de ceros y los datos restantes se hayan

transformado en la matriz AC reducida 380, el procedimiento MMS 350 comprime la matriz reducida en la etapa 368 usando una clave y los resultados se almacenan en una nueva matriz a la que se hace referencia en este documento generalmente como la matriz AC codificada. En el siguiente ejemplo, la compresión en el paso 368 es por un factor de 1/3, pero el valor estará en los valores clave utilizados y los valores de los coeficientes en la matriz AC reducida 380.

La etapa de compresión de la matriz AC reducida 368 del procedimiento MMS 350 se ilustra mediante el diagrama de flujo del proceso de la figura 7 y gráficamente en las figuras 8 y 9. El procedimiento 400 de compresión de matriz AC reducida incluye una primera etapa, en la que se usa una tecla para comprimir los datos en una tercera y una segunda etapa opcional en la que los datos comprimidos pueden comprimirse aún más separando los valores de datos distintos de cero e iguales a cero.

La primera etapa de compresión comienza en 402 generando una clave de compresión usando un valor máximo de los datos en la matriz AC reducida 380. La clave de compresión, Clave, utilizada en este ejemplo tiene tres componentes: Clave [1], Clave [2] y Clave [3]. Cada matriz AC reducida dará lugar a sus propios valores clave, ya que los valores clave dependen del valor máximo en la matriz AC reducida. El principio de la clave es similar a los pesos utilizados en la red neural Perceptron: $P=AW_1+BW_2+CW_3$, donde W_i son valores de peso que se generan aleatoriamente y «A», «B» y «C» son datos. El resultado de esta suma es «P» y solo hay una combinación posible para los valores de datos para W_i dada.

En 402, se utiliza un procedimiento generador de claves para generar los valores clave. Se selecciona el valor máximo de los elementos de datos en la matriz AC reducida 380, y los tres componentes clave se generan según las etapas siguientes:

M= MAX VALUE +(MAX VALUE/2); %% máximo valor elegido y dividido por «2»
 Key[1]=0.1, %% primer valor clave <=1
 Key[2]=(0.1×M) ×Factor; %% Factor=1,2,3,..etc.
 Key[3]=(KEY(1) ×M+KEY(2) ×M)×Factor;
 KEY= [Key[1]. Key[2], Key[3]]

y que son los valores clave finales utilizados en los procedimientos de compresión y descompresión

En 404, se seleccionan los primeros tres elementos de datos 382, 384, 286 de la matriz AC reducida 380 y se combinan en un solo valor según la ecuación siguiente:

$$CodedArr(P) = Key[1]* RA(L) + Key[2]* RA(L + 1) + Key[3]* RA(L + 2)$$

donde «RA» representa datos de matriz AC reducida;

$L=1, 2, 3, \dots N-3$, donde «N» es el tamaño de la matriz AC reducida

$P=1,2,3 \dots N/3$, donde «P» denota la posición dentro de la matriz AC codificada

y que se almacena como el primer elemento de datos 392 de la matriz AC codificada 390. En 408, los siguientes tres elementos de datos, del 4.º al 6.º, de la matriz AC reducida 380 se seleccionan y el procesamiento retorna 412 a la etapa 404 y se comprimen usando la misma Clave para generar el segundo elemento de datos 394 de la matriz AC codificada 390. El procesamiento continúa en bucle de esta manera hasta que toda la matriz AC reducida se haya comprimido para generar la matriz AC codificada 390 y que tiene un tercio del número de elementos de datos.

En el seudocódigo, la primera etapa de compresión 402-412 del procedimiento 400 de compresión de matriz AC reducida se puede expresar de la siguiente manera:

Let Pos-1
 MAX_Value=Find_Max_value(matriz AC reducida);
 W==Key-Generator (0.1, MAX_Value, 2);
 I=1;
 Mientras que (I< tamaño de matriz reducida)
 Para K=0 a 2
 Coded AC Array[Pos]= Coded AC Array[Pos] + (Reduced-Array[I+K] × W[K]);

```

    End; %% final para
    Pos++;
    l=l+3;
End; %% final mientras que

```

5

La matriz AC codificada generalmente contendrá muchos ceros con algunos elementos de datos distintos de cero. Todos los ceros se pueden borrar y solo se pueden mantener los datos distintos de cero. Por lo tanto, como segunda etapa de compresión adicional del procedimiento 400 de compresión de matriz AC reducida se puede usar
 10 opcionalmente. Este procedimiento de compresión adicional generalmente implica separar los datos de cero de los datos distintos de cero. Los datos de cero se pueden calcular fácilmente calculando el número de elementos de datos de cero entre dos elementos de datos distintos de cero.

Por ejemplo, suponga el siguiente ejemplo de matriz AC codificada = [0,5, 0, 0, 0, 7,3, 0, 0, 0, 0, -7] como se ilustra
 15 en la figura 9. En el paso 420, la matriz codificada 390 se separa en una matriz que representa los datos cero 398 y una matriz 386 que incluye todos los datos distintos de cero = [0,5, 7,3, -7]. La matriz que representa los datos cero, la matriz cero, es = [0, 3, 0, 5, 0] donde los 0 indican la posición de los datos distintos de cero en las posiciones correspondientes en la matriz AC codificada 390 y los números finitos indican el número de ceros entre dos elementos de datos consecutivos distintos de cero. Con el fin de aumentar la relación de compresión, en 422, los datos en la
 20 matriz cero se pueden convertir en valores de datos que tienen una mayor frecuencia de ocurrencia para una mayor compresión. Por ejemplo, el número «5» en la matriz cero 398 se puede dividir en los números «3» y «2» para aumentar la probabilidad de datos redundantes, por ejemplo, ya que el número «3» ahora aparece más frecuentemente en la matriz 399. Por lo tanto, el nuevo equivalente de matriz cero 399 es [0, 3, 0, 3, 2, 0].

25 El paso final del procedimiento de compresión 200 es la codificación aritmética 270 para la compresión sin pérdidas de la matriz DC 330 y las matrices AC codificadas 396, 399. La codificación aritmética es generalmente conocida en la compresión de datos. La codificación aritmética toma un flujo de datos y los convierte en un único valor de coma flotante. Los valores de salida están en el rango de menos de uno y mayor que cero y cuando se decodifica el valor de coma flotante, devuelve exactamente el mismo flujo de datos. La etapa 270 de codificación aritmética calcula la
 30 probabilidad de que todos los elementos de datos en el flujo de datos se codifiquen y asigna un valor en el rango de cero a uno para cada elemento de datos con el fin de generar flujos de bits comprimidos. Los principios de la codificación aritmética se describen, por ejemplo, en *Introduction to Data Compression*, 2nd Edition, Sayood., K., Academic Press. Por lo tanto, en 270 la matriz DC comprimida 330 está codificada aritméticamente, la matriz AC 396 codificada distinta de cero está codificada aritméticamente y la matriz AC codificada cero 399 está codificada
 35 aritméticamente.

El resultado de aplicar el procedimiento de compresión a los datos de la imagen original es un archivo de imagen comprimido 430 que tiene la estructura del archivo y los elementos de datos ilustrados en la figura 10. Una parte del encabezado del archivo 430 almacena la Clave 432 utilizada durante la compresión de la matriz AC reducida y también
 40 la matriz limitada de datos 434 generada al comienzo del procedimiento MMS. Una parte del cuerpo del archivo de imagen comprimido almacena los datos codificados aritméticamente para los componentes rojo 436, azul 438 y verde 440 de los datos de píxeles originales. Obviamente, para otros sistemas de color, se utilizarán los componentes de color correspondientes y para una imagen en escala de grises solo se usa un único campo para el valor de escala de grises codificado aritméticamente para los datos de píxeles originales. Para una imagen fija, por ejemplo, una
 45 fotografía, se proporciona un único archivo 430. Para las imágenes de vídeo, también se usa un solo archivo, pero con el cuerpo incluye datos para cada cuadro consecutivo de datos de imágenes de vídeo. Para las imágenes en 3D, se utiliza el mismo procedimiento de compresión, porque una superficie en 3D es efectivamente una imagen en 2D que se convierte en una estructura en 3D mediante un proceso posterior.

50 La figura 11 muestra un diagrama de flujo del proceso que ilustra el procedimiento de descompresión 450 de la invención y que corresponde generalmente al paso 108 de la figura 1. El procedimiento de descompresión 450 generalmente es el inverso del procedimiento de compresión 200 y consta de tres etapas principales: decodificación de la matriz DC; decodificación de la matriz AC reducida; y la reconstitución de los datos de píxeles mediante la aplicación de cuantificación inversa y un DCT inverso. En 452, la parte del encabezado del archivo comprimido 430
 55 se lee para obtener la clave y la matriz de datos limitada utilizada durante la compresión de la matriz AC y que se utilizan para reconstituir los elementos de la matriz AC.

La primera etapa del proceso de descompresión propiamente dicha ocurre en 454 y la matriz DC codificada aritméticamente recibida es aritméticamente. Luego, en 456, la matriz DC decodificada 470 se descomprime en una
 60 matriz de coeficientes DC DCT cuantificados 472, como se ilustra gráficamente en la figura 12. El procedimiento de descompresión utilizado en 456 es esencialmente el inverso del proceso *DBV* o *Diferencia entre dos valores* descrito anteriormente. Se aplica una adición entre dos valores o proceso ABV a la matriz DC decodificada 470 en 456 para

devolver los valores de coeficiente DCT DC cuantificados originales. ABV representa una ecuación inversa a DBV y se aplica a pares de valores adyacentes en la matriz DC decodificada 470, por ejemplo, 476, 478, y toma el último valor en la posición m , 476, y lo agrega al valor anterior, 478 en la posición $m-1$ y almacena el total actual en la última posición 480 de la matriz DC 472. Luego, el total actual se agrega al siguiente valor anterior 480 en la posición $m-2$, y el resultado almacenado en la penúltima posición 482 de la matriz DC 472, y así sucesivamente. La siguiente ecuación define el proceso de descompresión ABV:

$$D(i-1) = D(i-1) + D(i)$$

10 Donde $i = m, (m-1), (m-2), (m-3), \dots, 2$ y m es el número de elementos en la matriz DC decodificada 470.

Por ejemplo, usando la misma matriz DC codificada que la anterior, [0, -1, 1, 0, 1, -2, -1, -1, 4, -1, 0, 14], el último elemento (m^{th}) en la matriz decodificada es 14, el $m-2^{\text{th}}$ es $0 + 14 = 14$, el elemento $m-3^{\text{th}}$ es $-1 + 14 = 13$, el elemento $m-4^{\text{th}}$ es $4 + 13 = 17$, etc., y por lo tanto, la matriz DC original está decodificada. Por lo tanto, el resultado de la etapa 15 456 es una matriz de coeficientes DC DCT cuantificados.

La decodificación aritmética descomprime la matriz AC cero y la matriz AC distinta de cero, y luego combina la matriz AC cero con la matriz AC distinta de cero para generar la matriz AC codificada incluyendo ceros. Por ejemplo, la matriz AC descomprimida distinta de cero es = [0,5, 7,3, -7] y la matriz AC descomprimida cero es = [0, 3, 0, 5, 0], por lo que 20 la combinación resultante es la matriz AC codificada que es [0,5, 0, 0, 0, 7,3, 0, 0, 0, 0, -7].

La decodificación de la matriz AC reducida en 460 utiliza un procedimiento que encuentra los valores de datos de alta frecuencia originales. El archivo de datos comprimido 430 contiene información sobre las claves de compresión utilizadas, $k[1]$, $k[2]$, $k[3]$ 432 y datos 434 sobre la probabilidad de los valores de datos individuales y esos valores de 25 datos individuales, los datos de matriz limitada, seguido de flujos de datos AC DCT comprimidos 436, 438, 440. Por lo tanto, el procedimiento de descompresión de datos AC DCT utilizado en 460 recoge cada elemento de datos comprimidos k de la matriz AC codificada para reconstruir los elementos de datos AC DCT originales $k \times 3$ de la matriz AC reducida. El procedimiento utilizado en el paso 460 puede denominarse un procedimiento *Parallel-Fast Match Search* porque utiliza flujos de procesamiento paralelos, cada uno con un algoritmo de búsqueda binario.

30 El procedimiento Parallel-FMS se ilustra en los diagramas de flujo del proceso de las figuras 13 y 15 y gráficamente en las figuras 14 y 16. La figura 13 muestra un diagrama de flujo del proceso que ilustra el procedimiento de descompresión de matriz AC codificada 500 utilizado en el paso 460 para regenerar la matriz AC DCT. En 502, se generan tres conjuntos 520, 522, 524 de los datos limitados, es decir, los valores únicos de los elementos de datos 35 que aparecen al menos una vez en la matriz AC reducida. En la matriz 520, L11 representa el primer elemento de datos de la primera matriz, L12, representa el segundo elemento de datos de la primera matriz, etc., hasta L1m, que representa el último elemento de datos m^{th} de la primera matriz. En la matriz 522, L21 representa el primer elemento de datos de la segunda matriz, L22, representa el segundo elemento de datos de la segunda matriz, etc., hasta L2m, que representa el último elemento de datos m^{th} de la segunda matriz. En la matriz 524, L31 representa el primer 40 elemento de datos de la tercera matriz, L32, representa el segundo elemento de datos de la tercera matriz, etc., hasta L3m, que representa el último elemento de datos m^{th} de la tercera matriz.

Inicialmente, las tres matrices 520, 522, 524 de datos limitados contienen los mismos valores, es decir $L11=L21=L31$, $L21=L22=L32$, etc., y en 502 el procedimiento 500 calcula todas las combinaciones posibles de los elementos de 520, 45 con $K[1]$, los elementos de 522 con $K[2]$ y los elementos de 524 con $K[3]$, y el resultado de cada combinación se almacena como un elemento de la matriz de combinaciones de claves de datos limitada 526. Por ejemplo, el primer elemento 528 de 526 está dado por $L11*k[1] + L21*k[2] + L31*k[3]$, el segundo elemento 530 de 526 está dado por $L11*k[1] + L21*k[2] + L32*k[3]$, etc. Después de que todas las combinaciones posibles del elemento 520, 522 y 524 con la clave se hayan calculado y almacenado en la matriz 526, la matriz 526 se ordena en orden ascendente en el paso 508 50 para formar la matriz de combinaciones de claves de datos limitada 532.

Luego, en 510, se busca el conjunto ordenado 532 utilizando los elementos de la matriz AC codificada 534 para reconstruir la matriz AC reducida. El procedimiento de búsqueda utilizado en el paso 510 implementa el algoritmo de búsqueda binaria en forma paralela. En particular, el paso 510 usa procesos de búsqueda l -binaria en paralelo para 55 reconstruir los elementos de datos originales $l \times 3$ utilizados para reconstruir la matriz AC reducida 536 como se ilustra en la figura 16. La figura 15 muestra un diagrama de flujo del proceso que ilustra un hilo del procedimiento de búsqueda binaria 540, y l de estos operan en paralelo. Cada uno de los procedimientos de búsqueda l -binaria selecciona 542 un elemento de datos AC comprimido de la matriz AC codificada 534, por ejemplo, el elemento de datos 570, y compara el elemento de datos AC 570 con el elemento central o medio de la matriz de combinaciones de claves de datos limitada ordenada 532. Si se determina que los valores coinciden en 544, entonces se ha encontrado un elemento 60 coincidente y se selecciona en 548 y el triplete correspondiente de los elementos de datos 574 se extrae usando la

clave, k, en 550, por ejemplo, el triplete L15, L27, L34. El triplete descomprimido de los elementos de datos AC se almacena 552 en su posición correspondiente en la matriz AC reducida 536. Por ejemplo, el elemento de datos 570 es de la tercera posición en la matriz AC codificada 534, y por lo tanto se coloca en la tercera posición en la matriz AC reducida 536. El siguiente elemento de datos de matriz AC codificada se selecciona 554 y procesando bucles.

5 De lo contrario, si en 544 se determina que el elemento de datos 570 es mayor que el elemento central de la matriz 532, entonces el procedimiento selecciona 558 la mitad de la matriz que tiene mayores valores de datos y la etapa 544 repite su acción en la submatriz a la derecha del elemento central. Alternativamente, si se determina que el valor no es mayor en 556, entonces en 560 el procedimiento selecciona la mitad del conjunto que tiene valores de datos
 10 menores y la etapa 544 repite su acción en la submatriz a la izquierda del elemento central. Por lo tanto, los pasos 544, 556, 558 y 560 se repiten para dividir en dos la matriz 532 hasta que se encuentre un valor de datos coincidente. Por lo tanto, el resultado del procedimiento 540 es reconstruir la matriz AC reducida 536 y descomprimir la matriz AC codificada 534 por un factor de tres para la clave de tres componentes utilizada en este ejemplo.

15 Luego, en 512, la matriz AC completa se reconstruye utilizando el procedimiento inverso del EZSN. La matriz AC reducida 536 almacena todos los datos para todos los bloques de datos distintos de cero de la matriz AC. Por lo tanto, los bloques de 8x8 de datos distintos de cero se reemplazan en sus posiciones correspondientes en la matriz AC 560 como se muestra en la figura 17 y los bloques de 8x8 restantes tienen valores de datos cero, ya que originalmente se eliminaron sobre esa base. Las posiciones (i, j) para bloques distintos de cero se almacenan en una matriz llamada
 20 LOC generada durante el procedimiento de compresión de matriz AC que elimina los bloques cero y almacena bloques distintos de cero y que contiene información sobre la ubicación de cada bloque distinto de cero. La matriz LOC se puede representar en unos pocos bytes y se guarda en el archivo de encabezado y sus efectos en el tamaño del archivo son insignificantes. Por lo tanto, al final del procedimiento 500, la matriz AC 340 ha sido reconstruida como matriz AC 560.

25 Volviendo a la figura 11 y con referencia adicional a la figura 17, en 462 los elementos de la matriz AC 560 se combinan con los elementos de la matriz DC 472 seguido de cuantización inversa (multiplicación de puntos con la ecuación (2) anterior), para recuperar la matriz de 8x8 de los coeficientes DCT 570, incluido el coeficiente DC DCT 572 y los coeficientes AC DCT, por ejemplo, 574, 576, para cada uno de los bloques de 8x8 píxeles. Finalmente, en
 30 464, el DCT inverso se aplica a la matriz de coeficientes DCT 570 usando la ecuación (6) que se muestra a continuación, para recuperar los datos de píxeles para el bloque 580 de 8x8 píxeles correspondiente de la imagen 582. Esto se lleva a cabo para cada canal de color de un bloque de píxeles y para cada bloque de píxeles de la imagen 582 a fin de recuperar todos los datos de color de píxeles para toda la imagen 582.

35
$$f(x, y) = \sum_{u=0}^{Block-1} \sum_{v=0}^{Block-1} a(u)a(v)C(u, v) \cos \left[\frac{(2X+1)u\pi}{2Block} \right] \cos \left[\frac{(2y+1)v\pi}{2Block} \right] \quad (6)$$

Como se señaló anteriormente, los procedimientos de compresión y descompresión pueden manejar imágenes 2D utilizadas en aplicaciones de imágenes 3D. El principio de funcionamiento del escaneo de superficie 3D es proyectar
 40 patrones de luz sobre la superficie objetivo cuya imagen es grabada por una cámara. La forma del patrón capturado se combina con la relación espacial entre la fuente de luz y la cámara, para determinar la posición 3D de la superficie a lo largo del patrón. Las principales ventajas del procedimiento son la velocidad y la precisión; se puede escanear una superficie desde una sola imagen 2D y procesarla en una superficie 3D en unos pocos milisegundos.

La siguiente tabla compara el procedimiento de compresión y descompresión de la invención con JPEG y JPEG2000
 45 para cinco imágenes 2D distintas. La calidad de la imagen descomprimida se refleja en los valores de error cuadrático medio (RMSE) enumerados en la tabla.

Nombre de la imagen	Tamaño de bloque utilizado por DCT	Tamaño de imagen original	Tamaño de imagen comprimida	RMSE 2D de la invención	RMSE 2D de JPEG	RMSE 2D de JPEG2000
radiografía	8x8	588 Kbytes	10 Kbytes	5,0	11,88	3,2
Ojo	64x64	9 Mbytes	14,2 Kbytes	4,89	15,3	4,1
Niña	16x16	2,25 Mbytes	21,2 Kbytes	10,48	21,1	6,4
Celda	64x64	8,5 Mbytes	9,8 Kbytes	4,2	16	2,5
Bebé	32x32	3 Mbytes	18,3 Kbytes	5,3	15,5	3,5

Como se puede ver, para la misma relación de compresión, la invención tiene un RMSE superior al de JPEG y un RMSE comparable al de JPEG2000. Sin embargo, la invención también se puede utilizar para comprimir imágenes fijas y vídeo sin introducir desenfoque, a diferencia de JPEG2000.

5

Las figuras 18A a 18C muestran visualmente los resultados de la primera línea de la tabla anterior, que compara la invención con JPEG y JPEG2000. La figura 18A muestra la imagen 600 resultante de comprimir una imagen de rayos X de 588 Kbytes a 10K bytes usando el procedimiento de la invención y luego descomprimirla usando el procedimiento de la invención. La figura 18B muestra la imagen 602 resultante de comprimir una imagen de rayos X de 588 Kbytes a 10 Kbytes y luego descomprimirla usando JPEG. La figura 18C muestra la imagen 604 resultante de comprimir una imagen de rayos X de 588 Kbytes a 10 Kbytes y luego descomprimirla usando JPEG2000. Como se puede ver, para la misma relación de compresión, la imagen JPEG 602 incluye artefactos de compresión, no visibles en la imagen 600. Además, la imagen 600 generada usando los procedimientos de la invención es visualmente indistinguible de la imagen 604 generada usando JPEG.

15

Los procedimientos de la invención realizan compresión de imagen de alta calidad para imágenes 2D y reconstrucción de imágenes 3D. Los procedimientos se basan en DCT junto con la separación de los coeficientes DC DCT y la minimización del tamaño de la matriz de coeficientes AC DCT. Los procedimientos de la invención pueden producir una mejor calidad de imagen a relaciones de compresión más altas que JPEG y JPEG2000 y es capaz de reconstruir con precisión 3D a relaciones de compresión más altas. Además, las imágenes 2D reconstruidas con relaciones de compresión más altas son visualmente equivalentes a JPEG2000 y mejores que el algoritmo de JPEG.

20

Varias características de los procedimientos de la invención son dignas de mención. La DCT se aplica utilizando un tamaño de bloque de píxeles grande, de al menos 4, pero preferiblemente ≥ 8 . La imagen transformada final se separa en los valores del coeficiente DC DCT y los valores del coeficiente AC DCT usando distintas matrices. Los coeficientes DC DCT pueden representarse utilizando solo unos pocos bytes, ya que todos los valores son muy frecuentes.

25

Dado que la matriz de coeficientes AC DCT cuantificados contiene muchos ceros, se utiliza una estrategia para reducir la cantidad de datos a comprimir eliminando los ceros y manteniendo datos distintos de cero. Esta estrategia mantiene la información más importante mientras reduce el tamaño de la matriz hasta en un 75 % o más.

30

Los datos distintos de cero se comprimen aún más, por un factor de tres, en un único valor de coma flotante. Este proceso convierte la matriz AC en una matriz unidimensional, con una relación de compresión aumentada mientras mantiene la calidad de los coeficientes AC.

35

Para la descompresión, se utiliza un algoritmo de búsqueda eficiente y rápido para encontrar los datos originales exactos de una matriz unidimensional, es decir, la matriz AC codificada, que luego se convierte de nuevo en la matriz AC, y que utiliza los valores clave y datos limitados.

40

Los valores clave y los datos limitados se utilizan para comprimir y descomprimir una imagen. Sin estos, las imágenes no pueden ser reconstruidas. Por lo tanto, estos elementos también se pueden usar para proporcionar cifrado, seguridad o gestión de derechos digitales, lo que evita la descompresión no autorizada de los datos de la imagen.

El procedimiento de compresión cuando se usa en imágenes en color verdadero (es decir, rojo, verde y azul), obtiene relaciones de compresión más altas y alta calidad de imagen para imágenes en 3D que contienen líneas de rayas verdes. Además, el procedimiento se puede usar en imágenes YCbCr con buena calidad a relaciones de compresión más altas para imágenes 2D y 3D.

45

Los procedimientos de la invención son particularmente adecuados para aplicaciones en tiempo real, tales como objetos de datos 3D y transmisión de datos de vídeo a través de redes informáticas, como Internet.

50

Generalmente, las realizaciones de la presente invención, y en particular los procesos implicados en la compresión y descompresión de datos de imágenes emplean diversos procesos que implican datos procesados, almacenados o transferidos a través de uno o más sistemas informáticos. Las realizaciones de la presente invención también se refieren a uno o más aparatos de procesamiento de datos para realizar estas operaciones. El o cada aparato puede estar especialmente construido para los fines requeridos, o puede ser una computadora de propósito general activada selectivamente o reconfigurada por un programa de ordenador y/o estructura de datos almacenados en el ordenador. Los procesos presentados aquí no están inherentemente relacionados con ningún ordenador en particular u otro aparato. En particular, se pueden usar varias máquinas de uso general con programas escritos según las enseñanzas del presente documento, o puede ser más conveniente construir un aparato más especializado para realizar los pasos del procedimiento requeridos. Una estructura particular para una variedad de estas máquinas aparecerá a partir de la descripción dada a continuación.

55

60

Además, las realizaciones de la presente invención se refieren a medios legibles por ordenador o productos de programas de ordenador que incluyen instrucciones de programa y/o datos (incluidas estructuras de datos) para realizar diversas operaciones implementadas por ordenador. Los ejemplos de medios legibles por ordenador incluyen, entre otros, medios magnéticos como discos duros, disquetes y cintas magnéticas; medios ópticos tales como discos CD-ROM; medios magnetoópticos; dispositivos de memoria de semiconductores y dispositivos de hardware que están especialmente configurados para almacenar y realizar instrucciones del programa, como dispositivos de memoria de solo lectura (ROM) y memoria de acceso aleatorio (RAM). Ejemplos de códigos de programa incluyen tanto códigos de máquina, como los producidos por un compilador y archivos que contienen un código de nivel más alto que se pueden ejecutar por ordenador utilizando un intérprete.

La figura 19 ilustra un sistema informático típico que, cuando se configura o diseña adecuadamente, puede servir como un aparato de procesamiento de datos de esta invención. El sistema informático 800 incluye cualquier número de procesadores 802 (también conocidos como unidades centrales de procesamiento o CPU) que están acoplados a dispositivos de almacenamiento, incluido el almacenamiento primario 806 (típicamente una memoria de acceso aleatorio o RAM), almacenamiento primario 804 (típicamente una lectura de solo memoria o ROM). La CPU 802 puede ser de varios tipos, incluidos microcontroladores y microprocesadores, como dispositivos programables (por ejemplo, CPLD y FPGA) y dispositivos no programables, como ASIC de matriz de puerta o microprocesadores de uso general. Como es bien sabido en la técnica, el almacenamiento primario 804 actúa para transferir datos e instrucciones unidireccionalmente a la CPU y el almacenamiento primario 806 se usa típicamente para transferir datos e instrucciones de una manera bidireccional. Ambos dispositivos de almacenamiento primario pueden incluir cualquier medio legible por ordenador adecuado, como los descritos anteriormente. Un dispositivo de almacenamiento masivo 808 también está acoplado bidireccionalmente a la CPU 802 y proporciona capacidad de almacenamiento de datos adicional y puede incluir cualquiera de los medios legibles por ordenador descritos anteriormente. El dispositivo de almacenamiento masivo 808 puede usarse para almacenar programas, datos y similares, y es típicamente un medio de almacenamiento secundario tal como un disco duro. Se apreciará que la información retenida dentro del dispositivo de almacenamiento masivo 808 puede, en los casos apropiados, incorporarse de manera estándar como parte del almacenamiento primario 806 como memoria virtual. Un dispositivo de almacenamiento masivo específico como un CD-ROM 814 también puede pasar datos unidireccionalmente a la CPU.

La CPU 802 también está acoplada a una interfaz 810 que se conecta a uno o más dispositivos de entrada/salida, como monitores de vídeo, bolas de seguimiento, ratones, teclados, micrófonos, pantallas táctiles, lectores de tarjetas de transductores, lectores de cinta magnética o de papel, tabletas, lápices, reconocedores de voz o de escritura a mano, u otros dispositivos de entrada conocidos como, por supuesto, otros ordenadores. Finalmente, la CPU 802 opcionalmente se puede acoplar a un dispositivo externo como una base de datos o un ordenador o red de telecomunicaciones usando una conexión externa como se muestra generalmente en 812. Con tal conexión, se contempla que la CPU podría recibir información de la red, o podría enviar información a la red en el curso de realizar las etapas del procedimiento descritas en este documento.

Aunque lo anterior ha descrito generalmente la presente invención según procesos y aparatos específicos, la presente invención tiene un rango de aplicabilidad mucho más amplio. En particular, los aspectos de la presente invención no se limitan a ningún tipo particular de datos de imagen. Un experto en la materia reconocerá otras variantes, modificaciones y alternativas dado el caso de la exposición anterior.

REIVINDICACIONES

1. Un procedimiento implementado por ordenador para comprimir datos de imágenes (104, 200) que comprende:
5 aplicar (210) una transformación de coseno discreta (DCT) a cada uno de una pluralidad de bloques de píxeles no superpuestos que abarcan un marco de datos de imagen para generar un conjunto de coeficientes DCT para cada bloque de píxeles que comprende un coeficiente DC DCT y una pluralidad de coeficientes AC DCT;
10 cuantificar (230) cada conjunto de coeficientes DCT para generar un conjunto de coeficientes DC DCT cuantificados y un conjunto de coeficientes AC DCT cuantificados;
formar (240) una matriz DC a partir del conjunto de coeficientes DC DCT cuantificados;
15 formar (250) una matriz AC a partir del conjunto de coeficientes AC DCT cuantificados;
formar (352) una matriz de datos limitada que comprende elementos que tienen valores que corresponden solo a cada valor único de los elementos de la matriz AC;
20 comprimir (350) la matriz AC eliminando bloques de datos de la matriz AC que tienen solo valores cero, en donde cada bloque de datos tiene el mismo tamaño fijo conocido, y formando una matriz de AC reducida a partir de bloques de datos de la matriz AC que incluye valores cero;
almacenar la posición en la matriz AC de cada bloque de datos de la matriz AC, incluidos los valores distintos de cero
25 en una matriz de ubicación;
generar una clave (402) utilizando el valor máximo de los elementos de la matriz AC reducida;
comprimir (368, 400) la matriz AC reducida utilizando la clave para formar una matriz AC codificada, donde la clave comprende una pluralidad de componentes clave y el mismo número de elementos de la matriz AC reducida que el número de componentes clave se combinan utilizando la clave para formar un solo elemento de la matriz AC codificada;
30 codificar aritméticamente (270) la matriz DC y la matriz AC codificada para formar datos codificados aritméticamente;
35 y
formar un archivo de imagen comprimido (430), que incluye los datos codificados aritméticamente (436, 438, 440) y almacena la matriz de ubicación en un encabezado del archivo de imagen comprimido, y almacena la clave (432) y la matriz de datos limitada (434).
40
2. El procedimiento (104, 200) de la reivindicación 1, en el que cuantificar cada conjunto de coeficientes DCT incluye dividir los coeficientes DCT de cada conjunto de coeficientes DCT por un factor de cuantificación que es mayor que al menos dos.
- 45 3. El procedimiento (104, 200) de la reivindicación 1 o 2, y que comprende además comprimir la matriz DC (240) para formar una matriz DC comprimida y donde la matriz DC comprimida se codifica posteriormente aritméticamente para formar los datos codificados aritméticamente.
4. El procedimiento (104, 200) de la reivindicación 3, donde la compresión de la matriz DC (240) incluye el
50 cálculo de una diferencia entre los valores de los elementos de la matriz DC y el que forma la matriz DC comprimida.
5. El procedimiento (104, 200) de cualquier reivindicación anterior, donde la compresión de la matriz AC reducida (400) comprende además formar (420) una primera matriz AC codificada que incluye los elementos distintos de cero de la matriz AC codificada y formar una segunda matriz AC codificada que incluye datos que codifican los
55 elementos cero de la matriz AC codificada.
6. El procedimiento (104, 200) de cualquier reivindicación anterior y que además comprende almacenar el archivo de imagen comprimido (106) localmente en un primer dispositivo de procesamiento de datos y/o transmitir el archivo de imagen comprimido (106) a una red de comunicaciones para almacenarlo en un segundo dispositivo remoto
60 de procesamiento de datos.
7. Un procedimiento implementado por ordenador para descomprimir datos de imágenes (108, 450) que

comprende:

abrir (452) un archivo de imagen comprimido que incluye datos codificados aritméticamente, en el que un encabezado del archivo de imagen incluye una matriz de ubicación;

5

decodificar aritméticamente los datos codificados aritméticamente para formar una matriz DC (454) y una matriz AC codificada (458), y donde los elementos de la matriz DC corresponden a un conjunto de coeficientes DC DCT cuantificados;

10 recuperar una clave y una matriz de datos limitada;

descomprimir la matriz AC codificada (460, 500) utilizando la clave y la matriz de datos limitada para formar una matriz AC reducida, donde la matriz de datos limitada comprende elementos que tienen valores que corresponden solo a cada valor único de una matriz AC original de coeficientes AC DCT cuantificados antes de la compresión y en el que la clave comprende una pluralidad de componentes clave y se generó utilizando un valor máximo de la matriz AC original;

15

reconstruir una matriz AC (512) utilizando la matriz de ubicación que almacena la posición en la matriz AC original de cada bloque de datos de la matriz AC original, incluidos los valores distintos de cero, formando bloques de datos en la matriz AC, incluidos los valores distintos de cero y obtenidos de la matriz AC reducida y formando el resto de la matriz AC con bloques de datos que solo tienen valores cero, en el que cada bloque de datos tiene el mismo tamaño fijo conocido y en el que los elementos de la matriz AC son un conjunto de coeficientes AC DCT cuantificados;

20

aplicar una cuantificación inversa (462) al conjunto de coeficientes DC DCT cuantificados y al conjunto de coeficientes AC DCT cuantificados para generar una pluralidad de conjuntos de coeficientes DCT, cada uno correspondiente a uno respectivo de una pluralidad de bloques de píxeles, cada conjunto de coeficientes DCT que comprende un coeficiente DC DCT y una pluralidad de coeficientes AC DCT; y

25

aplicar una transformación DCT inversa (464) a cada uno de la pluralidad de conjuntos de coeficientes DCT para formar una pluralidad de bloques de píxeles no superpuestos que abarcan un marco de datos de imagen.

30

8. El procedimiento (108, 450) de la reivindicación 7, y que comprende además descomprimir (456) la matriz DC para formar una matriz DC adicional en la que los elementos son el conjunto de coeficientes DC DCT cuantificados.

35

9. El procedimiento (108, 450) de la reivindicación 8, donde descomprimir la matriz DC incluye calcular un total entre los valores de los elementos de la matriz DC y almacenar los resultados en la matriz DC adicional.

10. El procedimiento (108, 450) de las reivindicaciones 7 a 9, en el que la descompresión de la matriz AC codificada incluye:

40

calcular (502) los valores de todas las combinaciones posibles de los elementos de la matriz de datos limitada cuando se combinan con la clave y almacenarlos en una matriz de combinaciones de claves de datos limitada;

45 aplicar un procedimiento de búsqueda (510) a la matriz de combinaciones de claves de datos limitada utilizando los valores de los elementos de la matriz AC codificada para encontrar valores coincidentes;

determinar la combinación de elementos de la matriz limitada correspondiente a cada valor coincidente; y

50 formar la matriz AC reducida (510) a partir de cada combinación determinada de elementos de la matriz limitada.

11. El procedimiento (108, 450) de la reivindicación 10, que comprende además clasificar (508) la matriz de combinaciones de claves de datos limitada por orden de tamaño para formar una matriz de combinaciones de claves de datos limitada ordenada.

55

12. El procedimiento (108, 450) de la reivindicación 10 u 11, donde el procedimiento de búsqueda implementa un algoritmo de búsqueda binario paralelo.

13. El procedimiento (104, 200, 108, 450) de cualquiera de las reivindicaciones 1 a 12, donde cada uno de la pluralidad de bloques de píxeles no superpuestos (302) tiene un tamaño de $n \times n$ y donde n es al menos 4.

60

14. El procedimiento (108, 450) de cualquiera de las reivindicaciones 7 a 13 y que además comprende

recuperar el archivo de imagen comprimido del almacenamiento local en un segundo dispositivo de procesamiento de datos y/o recibir el archivo de imagen comprimido a través de una red de comunicaciones desde un primer dispositivo de procesamiento de datos remoto.

- 5 15. Un procedimiento implementado por ordenador (100) para manejar datos de imágenes, que comprende:
- generar al menos una trama de datos de imagen (102);
- comprimir la trama de datos de imagen (104) usando el procedimiento de cualquiera de las reivindicaciones 1 a 6 para
10 formar un archivo de imagen comprimido;
- descomprimir el archivo de imagen comprimido (108) usando el procedimiento de cualquiera de las reivindicaciones 7 a 14 para formar la al menos una trama de datos de imagen; y
- 15 visualizar la al menos una trama de datos de imagen (110) en un dispositivo de visualización.
16. Un medio legible por ordenador que almacena instrucciones ejecutables por ordenador que configuran un dispositivo de procesamiento de datos para llevar a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 15.
20
17. Un dispositivo de procesamiento de datos que comprende al menos un procesador de datos y el medio legible por ordenador de la reivindicación 16.

Fig. 1

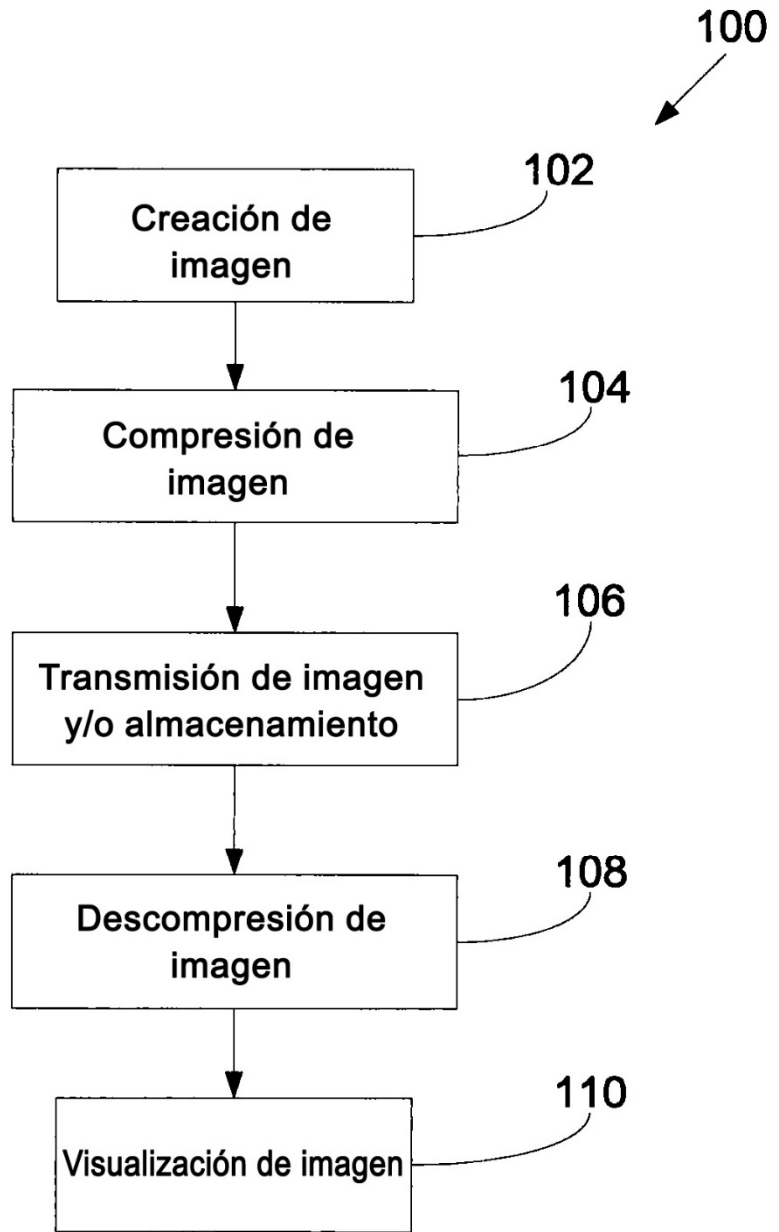


Fig. 2

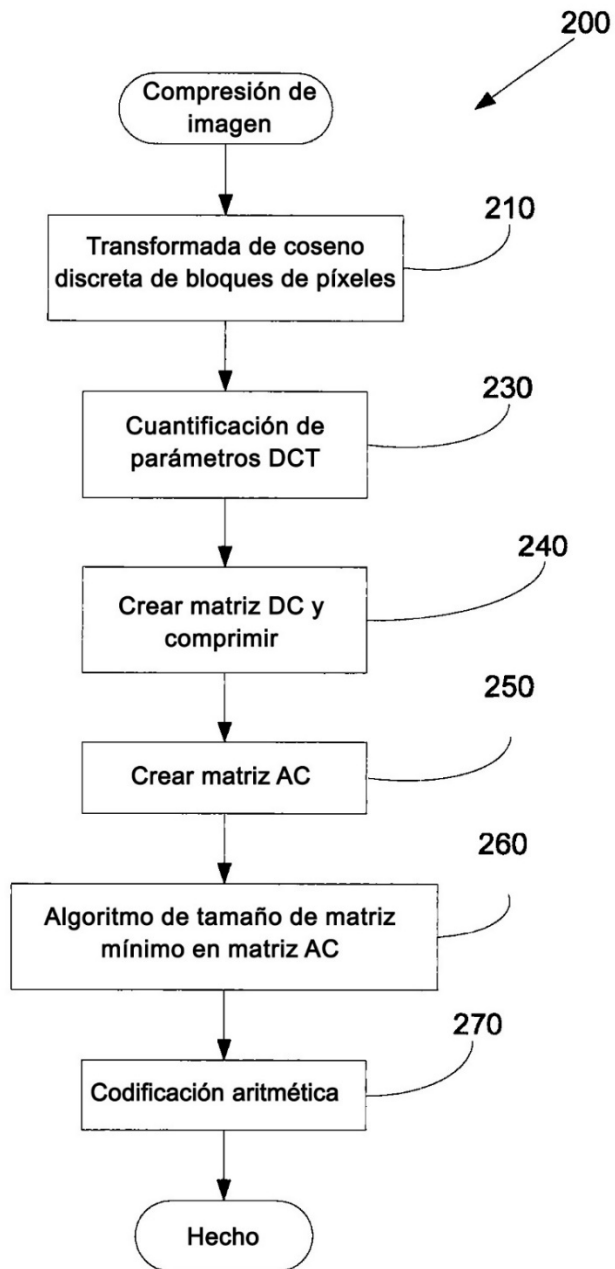


Fig. 10

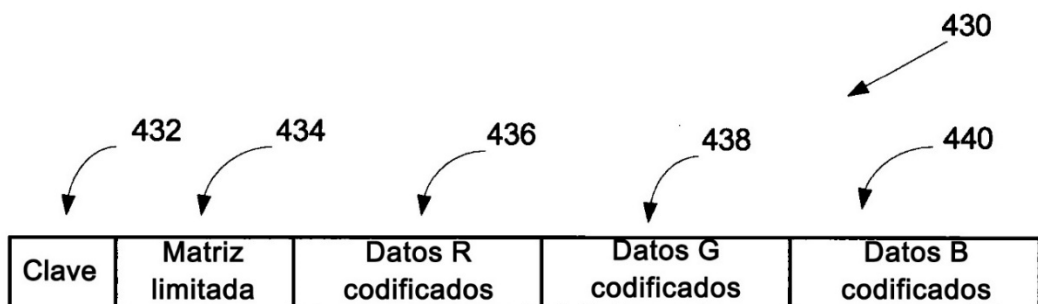


Fig. 3

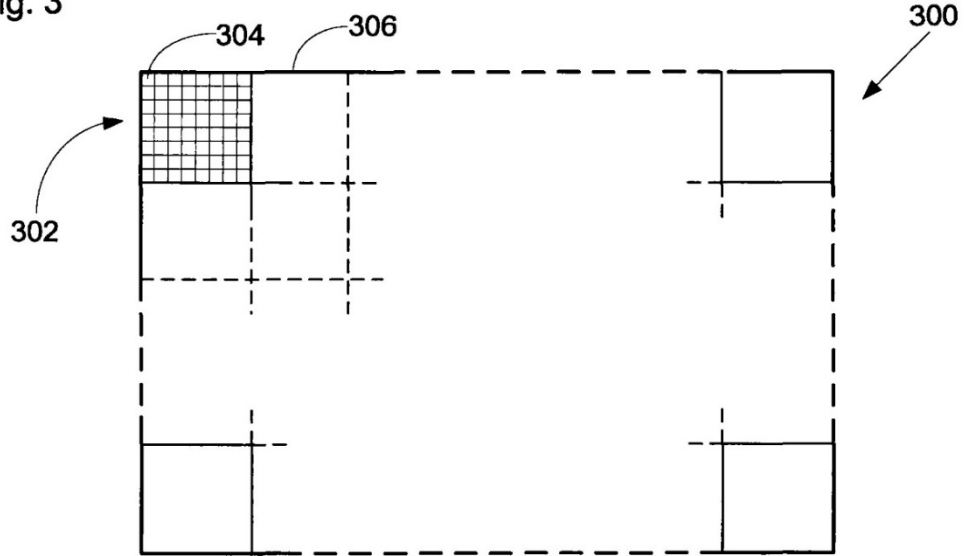


Fig. 4

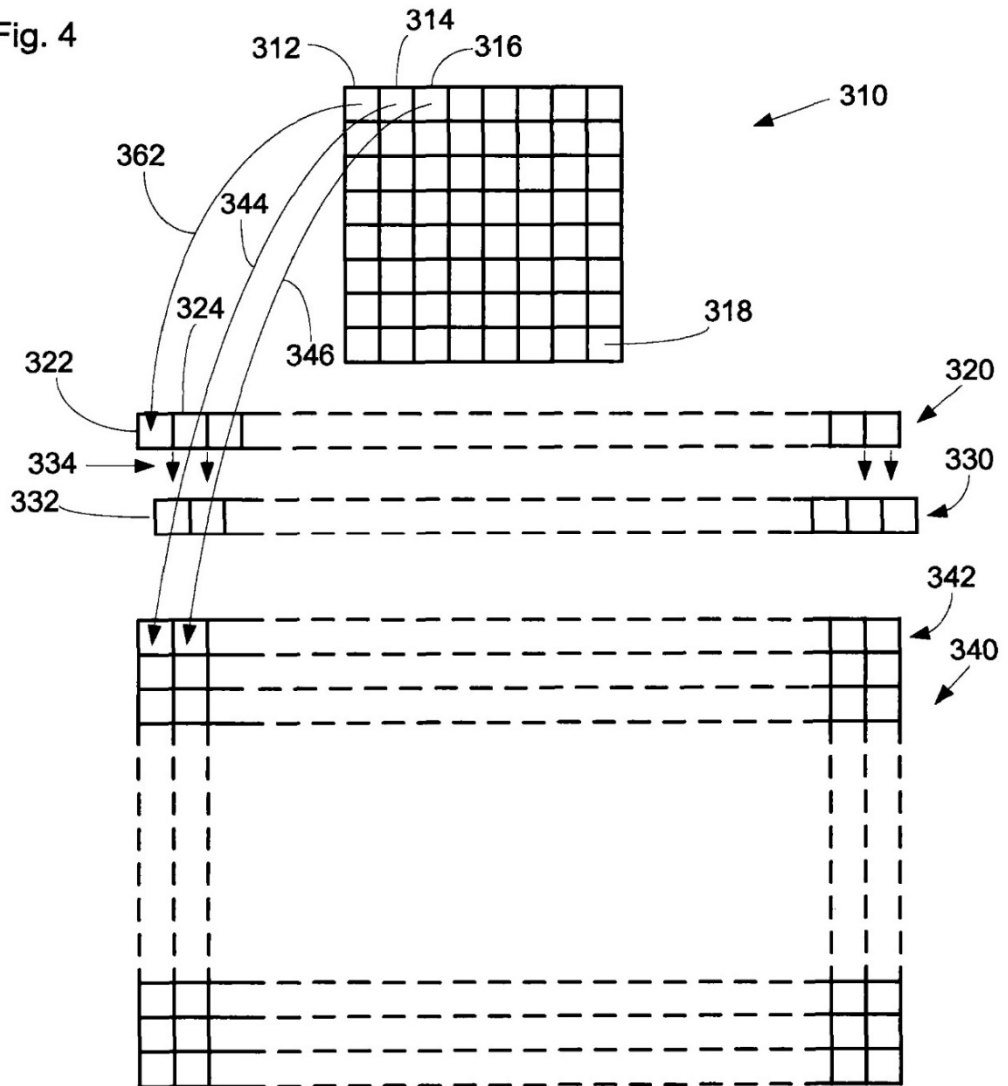


Fig. 5

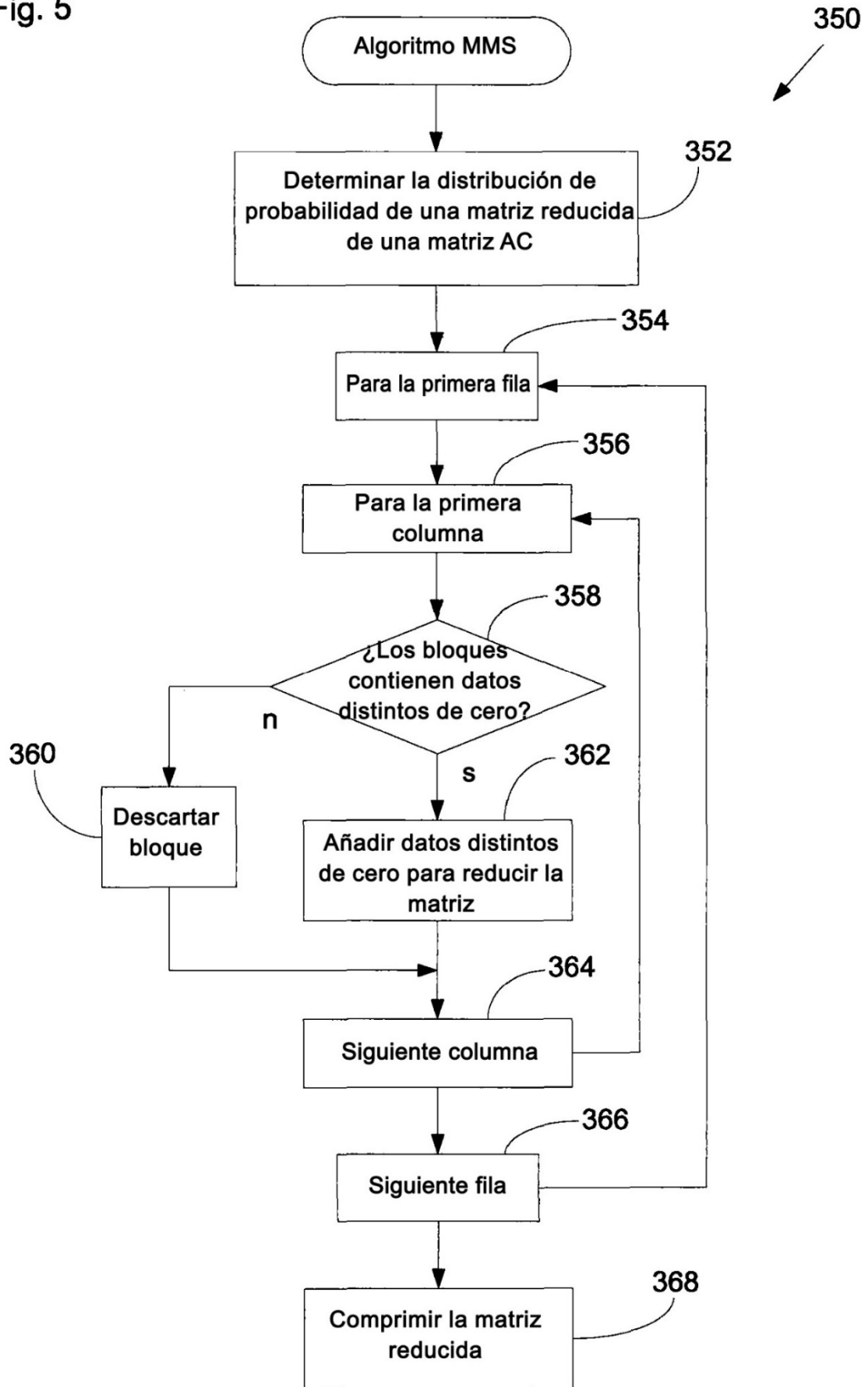


Fig. 6

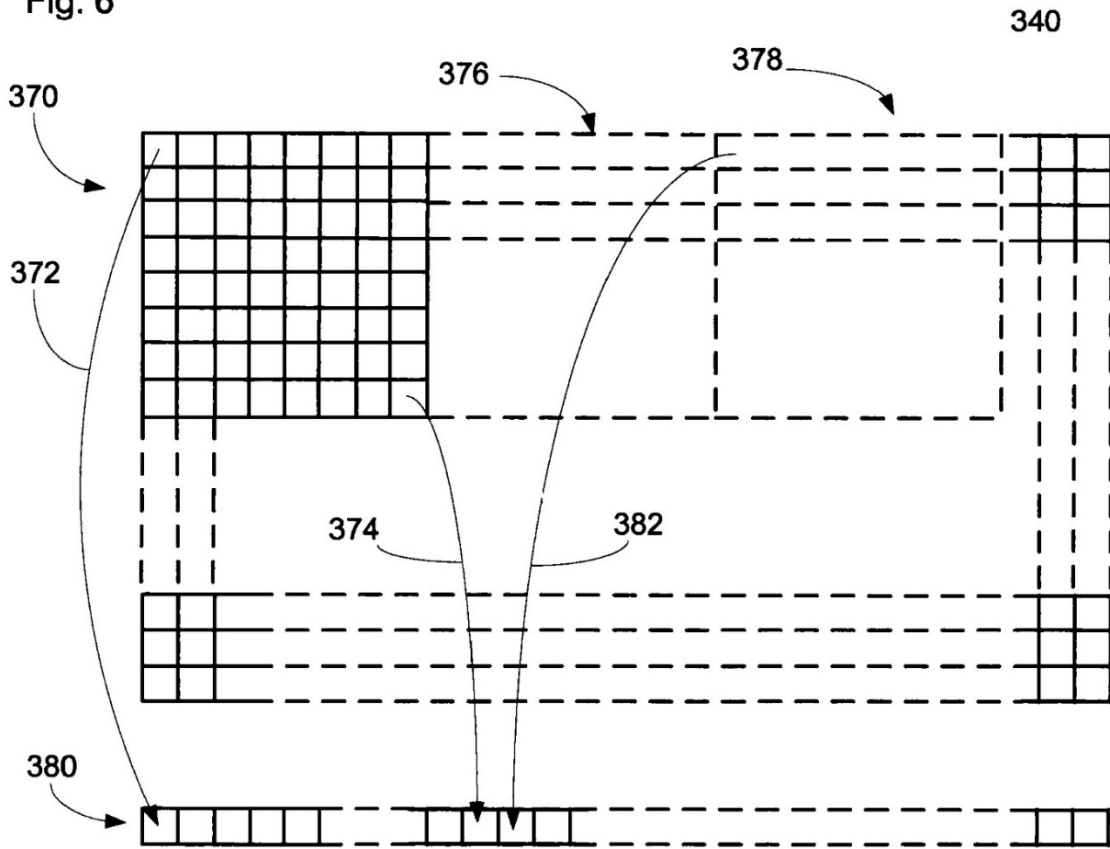


Fig. 8

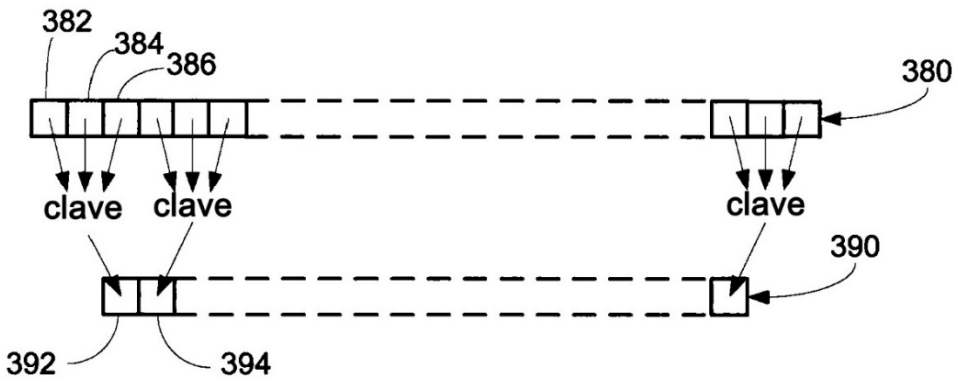


Fig. 9

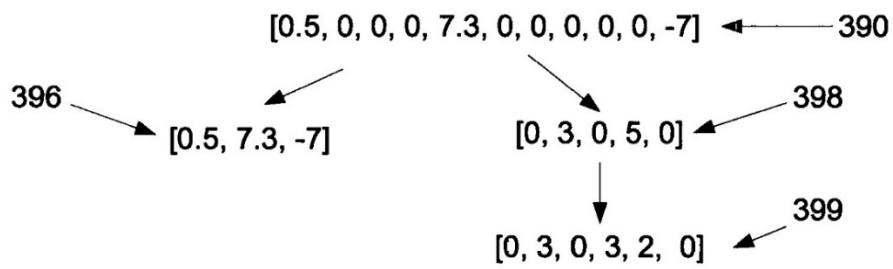


Fig. 7

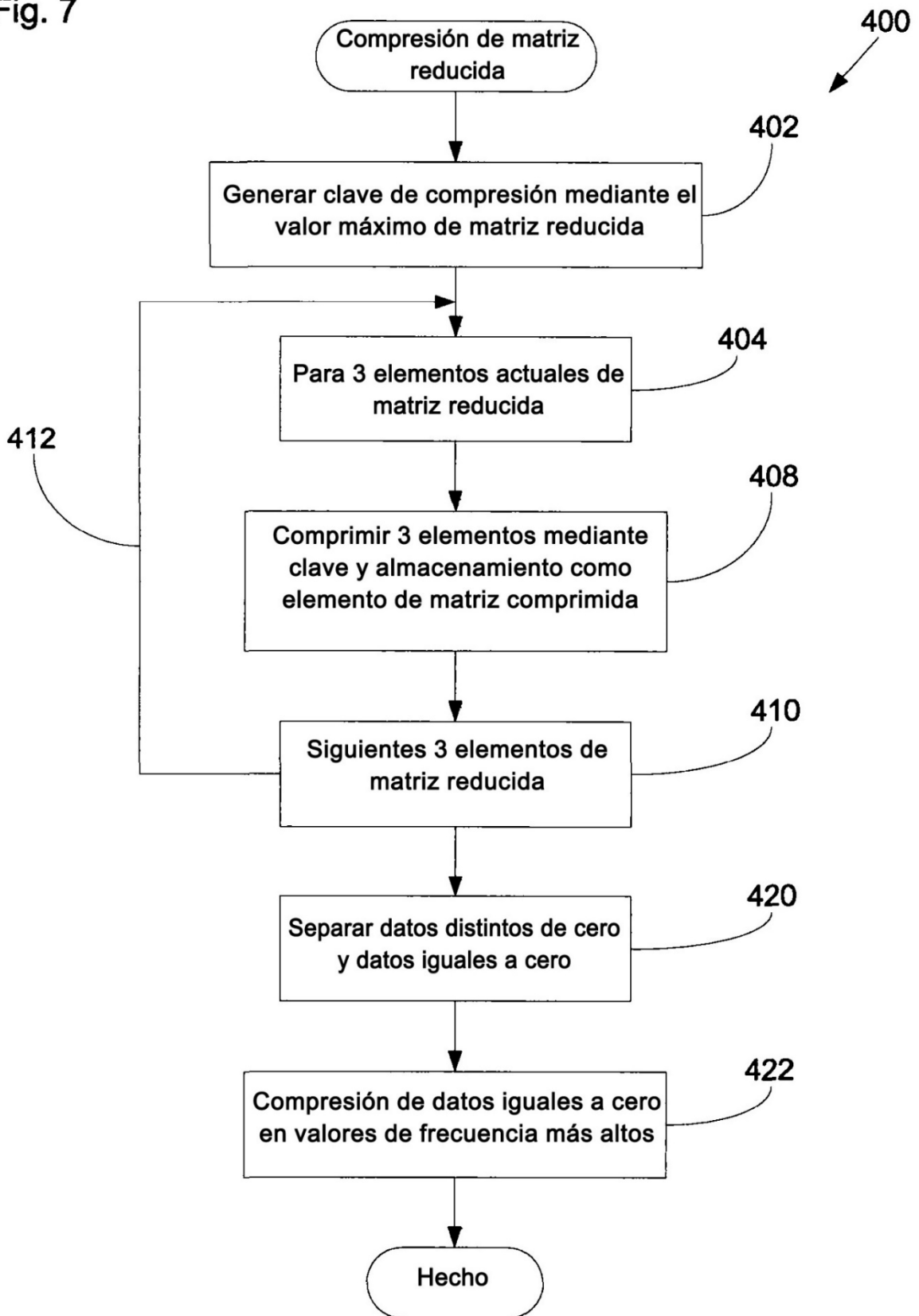


Fig. 11

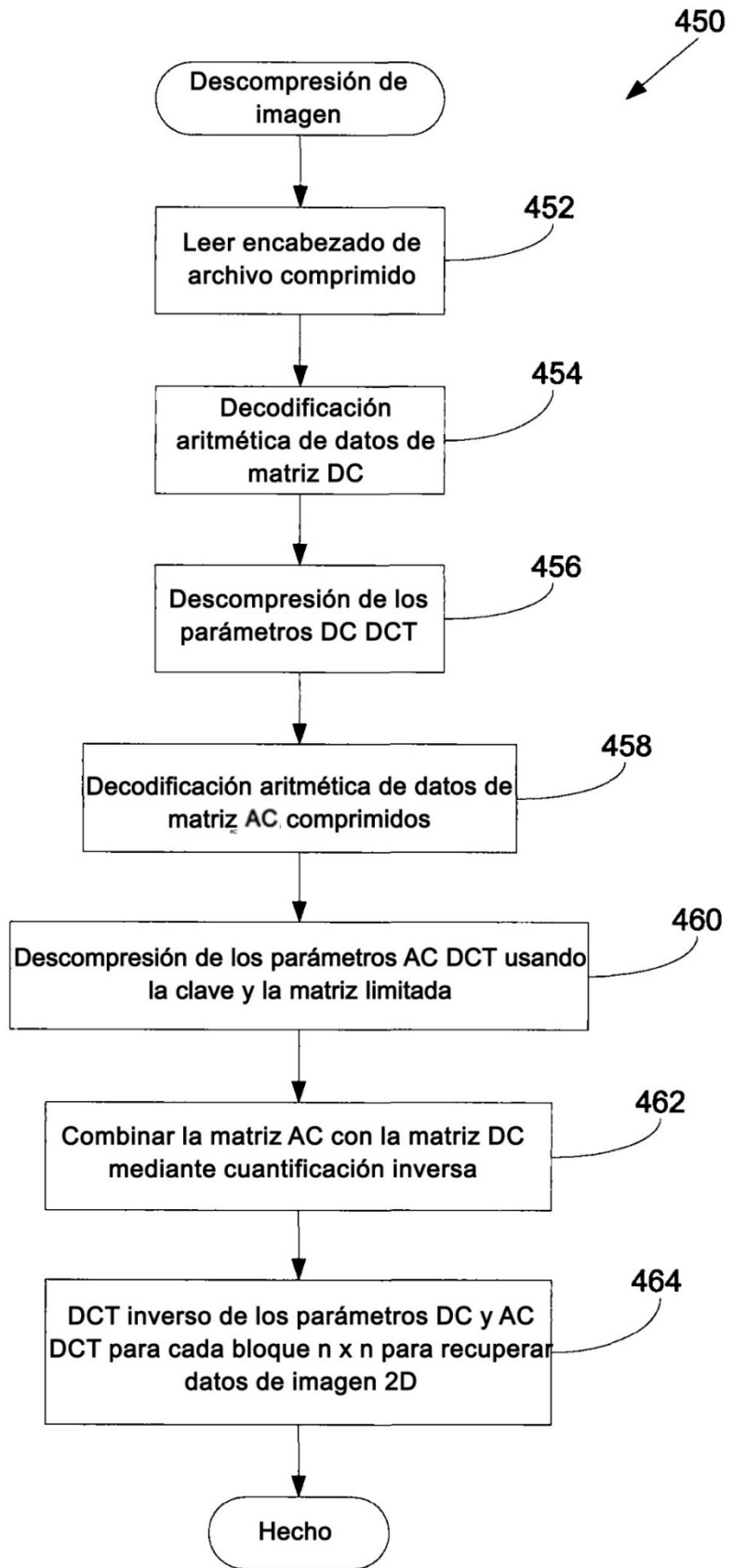


Fig. 12

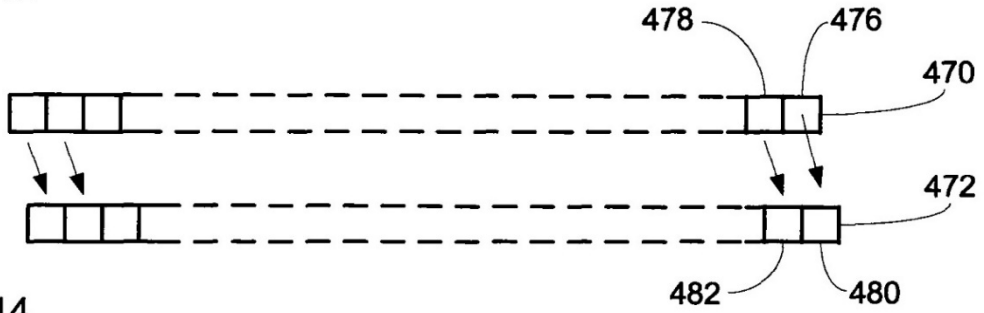


Fig. 14

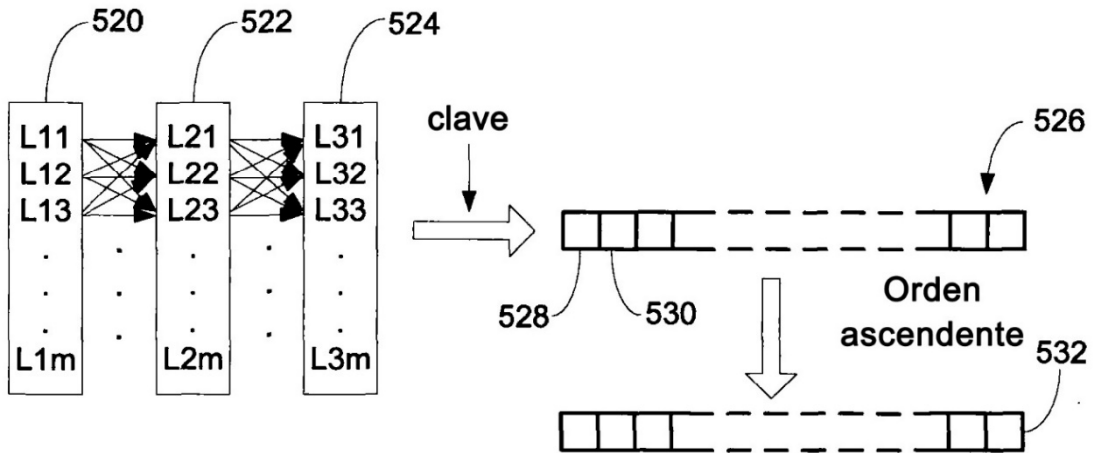


Fig. 16

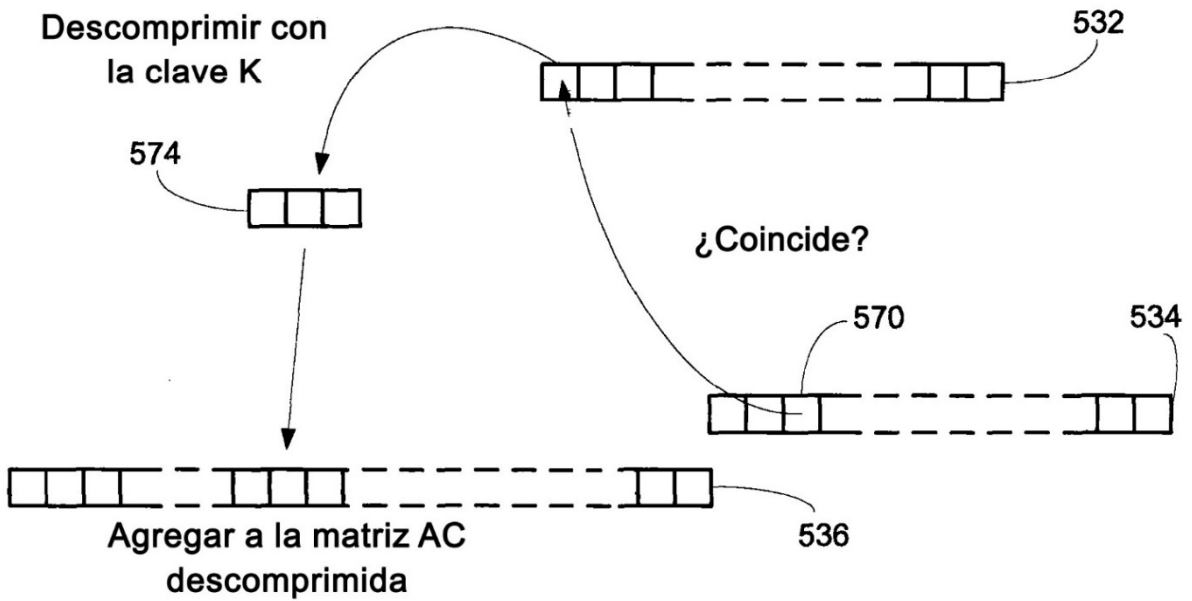


Fig. 13

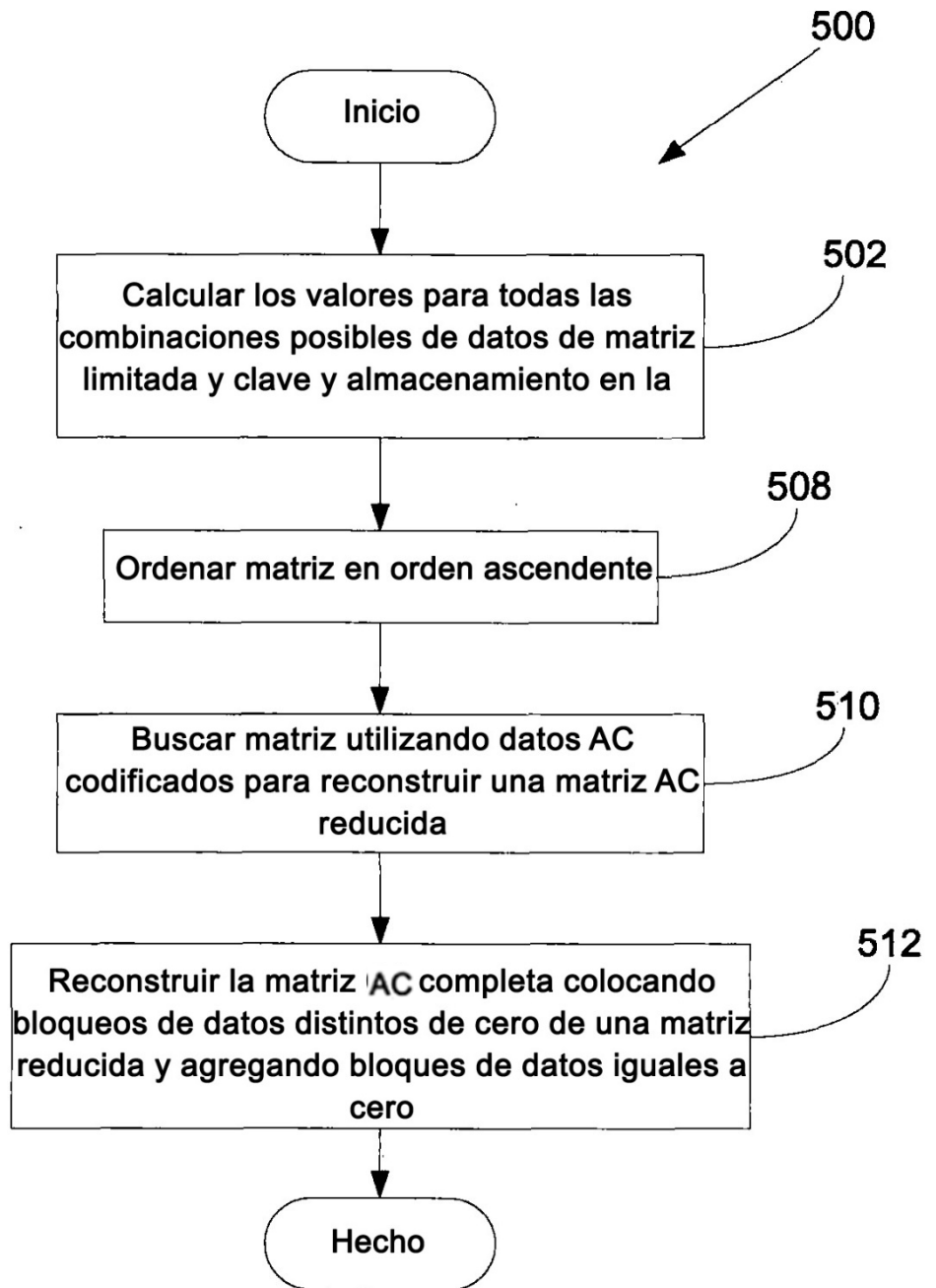


Fig. 15

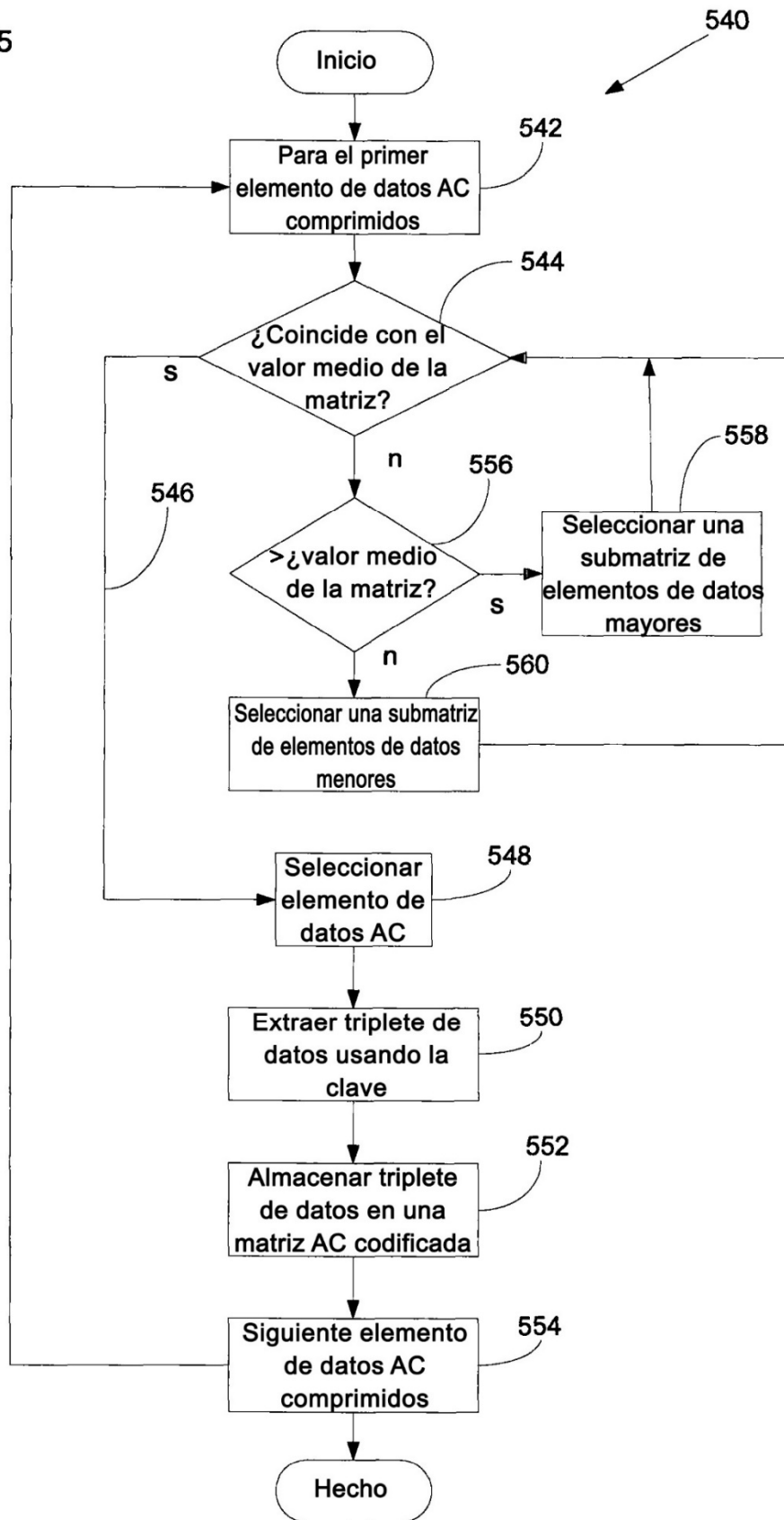


Fig. 17

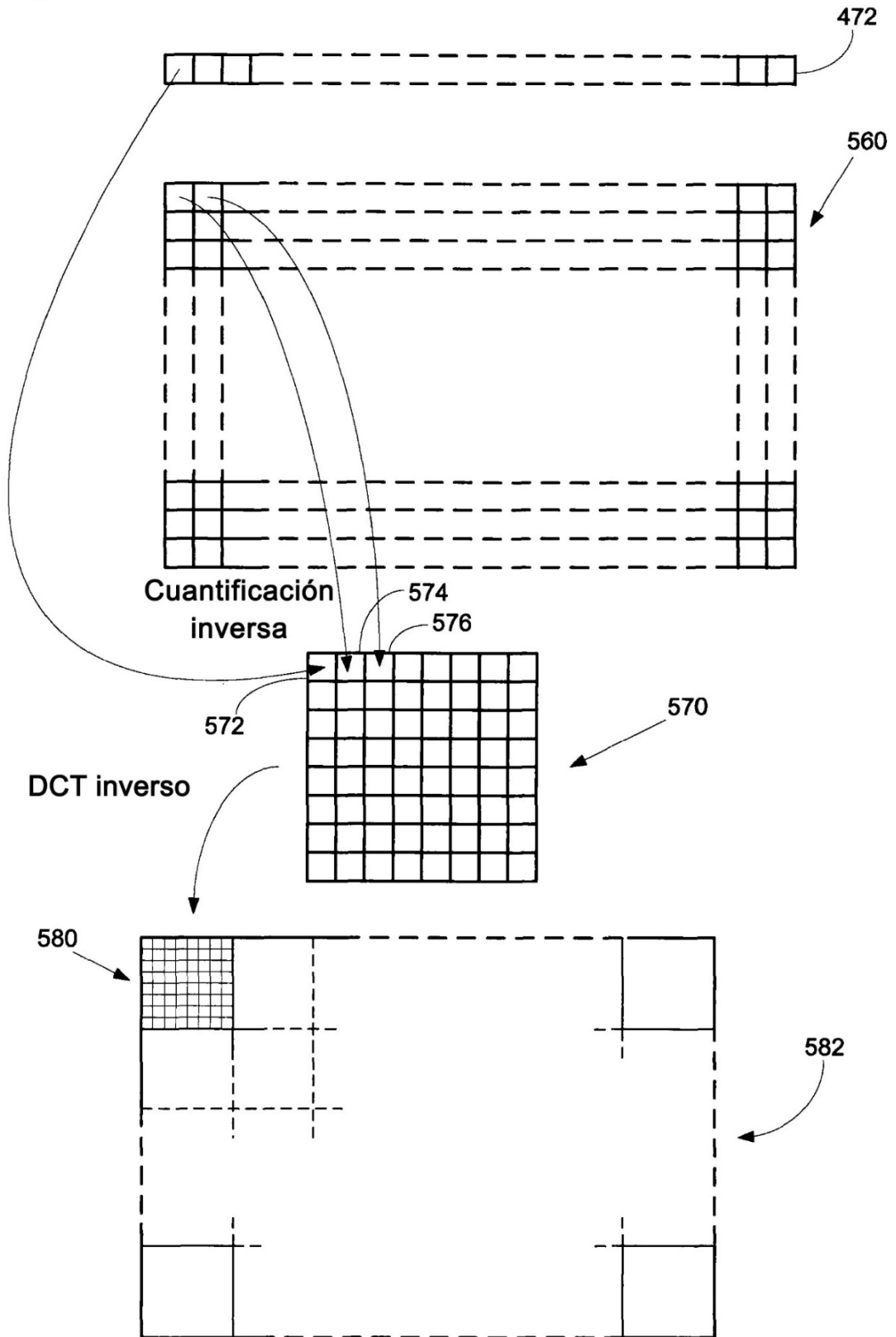


Fig. 18A

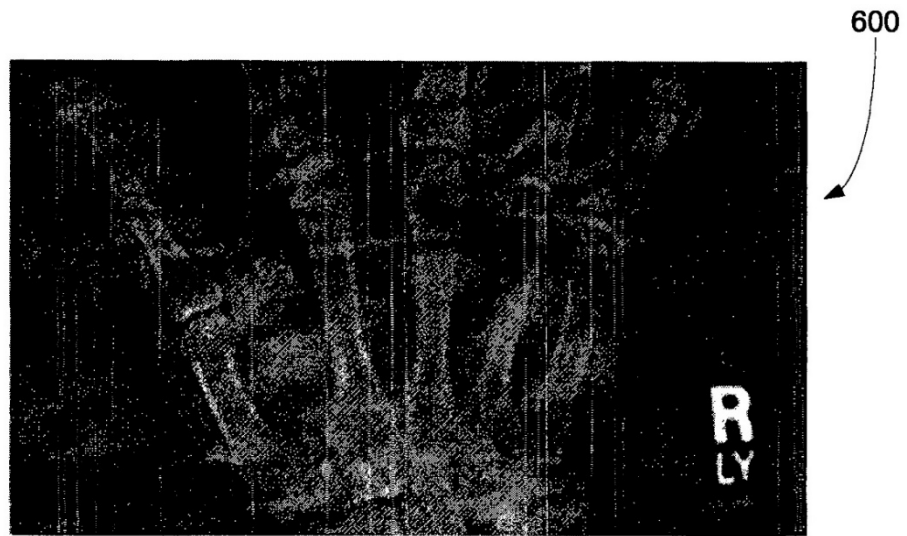


Fig. 18B



Fig. 18C



Fig. 19

