

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 527 962**

51 Int. Cl.:

G06F 13/28 (2006.01)

G06F 9/46 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **14.03.2007** **E 07733977 (8)**

97 Fecha y número de publicación de la concesión europea: **03.12.2014** **EP 1999600**

54 Título: **Sistema y método para permitir la extracción rápida de datos de imagen entrelazados**

30 Prioridad:

14.03.2006 US 375069

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
02.02.2015

73 Titular/es:

CORE WIRELESS LICENSING S.À.R.L. (100.0%)
16, avenue Pasteur
2310 Luxembourg, LU

72 Inventor/es:

KAIKUMAA, TIMO

74 Agente/Representante:

CURELL AGUILÁ, Mireia

ES 2 527 962 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

Sistema y método para permitir la extracción rápida de datos de imagen entrelazados.

5 Campo de la invención

La presente invención se refiere en general al procesado de datos en dispositivos electrónicos. Más particularmente, la presente invención se refiere al desentrelazado de diferentes tipos de datos en dispositivos electrónicos móviles, tales como teléfonos móviles con módulos de cámara incorporados.

10 Antecedentes de la invención

En los últimos años, los teléfonos móviles vienen cada vez más equipados con módulos de cámara incorporados. Estos módulos de cámara permiten que un usuario saque una fotografía digital y que la transmita rápidamente a los amigos y la familia.

En la Figura 1 se muestra con la referencia 10 un sistema genérico de cámara digital que forma parte de un teléfono móvil u otro dispositivo electrónico. La cámara digital 10 incluye una caja 11 que contiene por lo menos una lente 12 y por lo menos un sensor de imágenes 18 el cual está conectado operativamente a una unidad de memoria 14 y a un procesador o controlador 16, estando situados los dos fuera del módulo de cámara 10 y dentro del teléfono móvil u otro dispositivo electrónico dentro del cual está alojado el módulo de cámara 10. La unidad de memoria 14 se puede usar para almacenar imágenes digitales y software de ordenador con el fin de llevar a cabo varias funciones en la cámara digital 10, así como para implementar la presente invención. El sensor de imágenes 18 puede ser un dispositivo acoplado por carga (CCD), un semiconductor complementario de óxido metálico (CMOS), u otro sistema.

Los módulos de cámara que se usan en teléfonos móviles se diseñan comúnmente de manera que sean de tamaño reducido y económicos por muchos motivos. Como consecuencia de este hecho, un sensor de imágenes típico 18 no incluye ninguna memoria de cuadro dentro del módulo de cámara 10. En su lugar, a los datos de imagen provenientes de una imagen capturada se les debe dar salida hacia el motor telefónico en cuanto son leídos del sensor. De lo contrario, los datos de imagen se perderían. El sensor de imágenes 18 se lee típicamente al estilo línea a línea, y los datos se envían en el mismo orden al motor telefónico, en donde existe suficiente memoria para almacenar los datos.

Para reducir el ancho de banda necesario para la transmisión de los datos de imagen entre el módulo de cámara 10 y el motor telefónico, frecuentemente el sensor de imágenes 18 comprime los datos antes de enviarlos a ningún otro sitio. Para reducir los requisitos de procesado en el motor telefónico, muchos módulos de cámara usan en esta fase la compresión JPEG, ya que es probable que en cualquier caso las imágenes acaben en un formato JPEG de manera que el motor telefónico no necesita consumir tiempo adicional para la codificación JPEG.

Desde una perspectiva de la facilidad de utilización, no es suficiente que las imágenes de cámara únicamente sean procesadas y guardadas en memoria. Esto es debido a que al usuario final también le gustaría ver la imagen que acaba de sacar en la pantalla del teléfono, por ejemplo, para decidir si la imagen tenía o no una calidad satisfactoria. Es también deseable desde la perspectiva del usuario para que no se produzca ningún retardo adicional cuando se muestra al usuario la imagen de vista previa.

No obstante, desafortunadamente, la elaboración de una imagen de vista previa a partir de los datos comprimidos no es una operación inmediata, sino que, por el contrario, ocupa inevitablemente cierto tiempo adicional. Existen varias soluciones potenciales diferentes a esta cuestión, aunque cada una de ellas posee inconvenientes importantes en sí mismos. Una de estas soluciones implica permitir que el módulo de cámara 10 dé salida a los datos de la imagen capturada en un formato tal que permita la construcción rápida de la imagen de vista previa de tamaño menor que se va a mostrar en el dispositivo de visualización del teléfono. En la práctica, esto significa el uso de un formato no comprimido para los datos de imagen. No obstante, el problema de este planteamiento es que con cámaras de resolución mayor, podría no haber ancho de banda suficiente para transferir los datos.

Otra solución potencial a este problema es el uso de un módulo de cámara que incluya suficiente memoria de cuadro para permitir que la misma imagen se transmita dos veces, primero como una imagen con el tamaño adecuado para la visualización, y a continuación una imagen de resolución completa con el fin de guardarla. El problema de este planteamiento es que la memoria adicional es simplemente demasiado cara para implementar de manera práctica esta solución.

Otra solución potencial al problema antes descrito implica el entrelazado de una imagen comprimida para ser guardada y una imagen de vista previa no comprimida para ser visualizada. Aun cuando no haya memoria para un cuadro de imagen completo en el módulo de cámara, un codificador de JPEG de la cámara requiere que se disponga de memoria para ocho líneas de la imagen. Esto hace que resulte posible entrelazar los datos no comprimidos de imágenes de vista previa dentro de los datos comprimidos de imágenes de resolución completa. Puesto que los datos de vista previa están destinados al dispositivo de visualización (cuya resolución es

sustancialmente menor que la correspondiente de los sensores de cámara) del teléfono móvil, el entrelazado de la imagen de vista previa reducida en tamaño no hace que aumenten los requisitos de ancho de banda en una gran cantidad. En la Figura 2 se representa gráficamente el proceso de entrelazado, en el que dos tipos de datos se dividen en fragmentos y se envían uno después de otro. En la Figura 2, los rombos blancos y negros representan

marcadores específicos que marcan el comienzo de cada tipo de fragmento. En la práctica, los marcadores se deben seleccionar de manera que no puedan aparecer en los propios datos. Alternativamente, se podría incluir en cada marcador información sobre la longitud del fragmento. No obstante, en la codificación JPEG, por ejemplo, no es posible en general conocer el tamaño de los datos de salida antes de que haya tenido lugar la codificación. Por lo tanto, la cámara debería estar equipada con memorias intermedias adicionales para este planteamiento.

La transmisión de todos los datos de un cuadro de imagen requiere la transmisión de varios fragmentos de datos de vista previa y datos comprimidos. Un fragmento puede contener tanta información como pueda almacenar temporalmente el módulo de cámara en cualquier momento dado. Por ejemplo, si un codificador de JPEG requiere que haya disponibles 8 líneas en una memoria intermedia de un tipo dado, entonces es posible usar fragmentos que contienen datos por lo menos para esas 8 líneas. Debería indicarse que en el módulo de cámara también podría haber cierta capacidad adicional de almacenamiento temporal.

La Figura 2 ilustra también cómo fragmentos de datos de imagen de vista previa tienen siempre la misma longitud, aunque la longitud de fragmentos de datos comprimidos puede variar. Esto es debido a que la eficiencia de la compresión depende del contenido de las imágenes y puede variar incluso dentro de una imagen individual. Por otro lado, los datos de vista previa no están comprimidos y existe siempre, por lo tanto, la misma cantidad de datos de vista previa a enviar en un fragmento. Por este motivo, los marcadores después de los fragmentos de datos de vista previa (en el comienzo de los fragmentos de datos comprimidos) no son obligatorios, en la medida en la que la longitud del fragmento de datos de vista previa es siempre conocida.

Aunque el entrelazado de datos comprimidos y no comprimidos aborda muchas de las cuestiones que surgen en las otras soluciones potenciales descritas previamente, sigue presentando una deficiencia significativa. Con este planteamiento, aun cuando hay disponibles datos comprimidos para ser guardados y datos de vista previa para ser visualizados, los datos entrelazados en primer lugar se deben desentrelazar antes de que puedan ser usados. En otras palabras, los datos comprimidos se deben extraer a partir de los datos de vista previa, o los datos de vista previa se deben extraer a partir de datos comprimidos, antes de que los datos comprimidos se puedan guardar y los datos de vista previa puedan ser visualizados.

En la mayoría de teléfonos móviles convencionales con un módulo de cámara incorporado, una vez que se ha recibido la totalidad de los datos de imagen en una memoria dentro del motor telefónico, los mismos son procesados o copiados adicionalmente por el procesador de teléfono de tal manera que todos los datos contenidos dentro de la imagen son direccionados sucesivamente. En el caso de datos entrelazados, los datos se copian en una de las dos memorias intermedias tras el procesado. Una memoria intermedia está dedicada a guardar los datos de vista previa (no comprimidos), y la otra memoria intermedia está dedicada a almacenar los datos comprimidos. Debería observarse que, en la práctica, una de estas dos memorias intermedias puede ser la misma que la memoria intermedia original, en caso de que los datos entrelazados originales ya no sean necesarios sino que se puedan sobrescribir. No obstante, este proceso consume simultáneamente tiempo y poder de procesado.

La patente US n.º 6.708.233 describe un método para almacenar temporalmente datos de encabezamiento y de carga útil a partir de un flujo continuo entrante, en dos memorias intermedias independientes. El método que se describe en la patente US n.º 6.708.233 se basa en el DMA para la transferencia de datos, y requiere además un descriptor de DMA personalizado con campos específicos que, entre otras descripciones, deben identificar la longitud de secciones de encabezamiento y de datos a transferir.

Es un objetivo de la presente invención proporcionar un sistema que permita la transmisión y disponibilidad más rápidas y eficientes de datos de imagen tanto comprimidos como no comprimidos en un teléfono móvil u otro dispositivo electrónico.

Sumario de la invención

Este objetivo se logra a través de un método según se expone en la reivindicación 1, un producto de programa de ordenador según se expone en la reivindicación 9, mediante un dispositivo electrónico según se expone en la reivindicación 15 y de forma adicional alternativamente mediante un circuito integrado según se expone en la reivindicación 21.

En particular, se proporciona un sistema y un método para desentrelazar eficientemente datos entrelazados usando una combinación de una unidad de microprocesador (MPU) y hardware de acceso directo a memoria (DMA). Puesto que tanto la MPU como el hardware de DMA se usan para procesar los datos al mismo tiempo, el tiempo de procesado total se puede reducir significativamente en función de las longitudes relativas de los fragmentos de los datos comprimidos y los datos de vista previa. La presente invención proporciona varias ventajas con respecto a sistemas convencionales. Con la presente invención, no es necesario que la MPU procese en absoluto los datos de

vistas previas; puede omitir todos los fragmentos de datos de vista previa cuyo tamaño sea conocido de antemano. Como consecuencia, la MPU dispone de más tiempo para llevar a cabo otras funciones en el sistema. Adicionalmente, debido a que tanto el hardware de DMA como la MPU procesan datos al mismo tiempo, el desentrelazado acaba mucho más rápidamente que en la solución tradicional con solamente la MPU. Todavía
 5 adicionalmente, una vez que el hardware de DMA está configurado apropiadamente, hay poca necesidad de configuraciones adicionales durante la operación. En una situación óptima – en función del hardware de DMA – únicamente es necesario indicar la dirección de lectura nueva y volver a habilitar la transferencia de datos. La presente invención se puede usar conjuntamente con una amplia variedad de tipos de datos entrelazados, de entre los cuales uno de los tipos tiene una longitud fija, y no se limita únicamente a datos de imagen.

10 Estas y otras ventajas y características de la invención, junto con la organización y forma de funcionamiento de la misma, se pondrán de manifiesto a partir de la siguiente descripción detallada cuando la misma se considere conjuntamente con los dibujos adjuntos, en donde los mismos elementos tienen numerales iguales en la totalidad de los diversos dibujos que se describen posteriormente.

15 Breve descripción de los dibujos

La Figura 1 es una vista lateral en sección de un sistema genérico de cámara digital;

20 la Figura 2 es una representación que muestra cómo se pueden entrelazar tanto datos comprimidos como datos de vista previa de imágenes no comprimidos para su transferencia desde un sensor de imágenes a un motor telefónico;

la Figura 3 es una vista en perspectiva de un dispositivo electrónico que puede incorporar los principios de la presente invención;

25 la Figura 4 es una representación esquemática de la circuitería del dispositivo electrónico de la Figura 3;

la Figura 5 es una representación de los diversos componentes implicados en la implementación de una forma de realización de la presente invención; y

30 la Figura 6 es un diagrama de flujo que muestra la implementación de una forma de realización de la presente invención.

35 Descripción detallada de las formas de realización preferidas

La presente invención proporciona un sistema de desentrelazado de datos entrelazados de imágenes comprimidos y no comprimidos usando una combinación de MPU y capacidad de procesamiento de DMA. En la medida en la que la MPU y el hardware de DMA procesan los datos al mismo tiempo, el tiempo de procesamiento total se puede reducir en un cincuenta por ciento o más en función de las longitudes relativas de los fragmentos de los datos comprimidos y los datos de vista previa.

40 Las Figuras 3 y 4 muestran un teléfono móvil representativo 112 sobre el cual se puede implementar la presente invención. No obstante, es importante observar que la presente invención no se limita a ningún tipo de dispositivo electrónico y que se podría incorporar en dispositivos, tales como asistentes personales digitales, ordenadores personales, y otros dispositivos. Debería entenderse que la presente invención se podría incorporar en una amplia variedad de teléfonos móviles 112. El teléfono móvil 112 de las Figuras 2 y 3 incluye una caja 130, un dispositivo de visualización 132 en forma de una pantalla de cristal líquido, un teclado 134, un micrófono 136, un auricular 138, una batería 140, un puerto de infrarrojos 142, una antena 144, una tarjeta inteligente 146 en forma de una tarjeta de circuito integrado universal (UICC) según una forma de realización de la invención, un lector de tarjetas 148, circuitería de interfaz de radiocomunicaciones 152, circuitería de códec 154, el controlador 16 y la unidad de memoria 14. Los circuitos y elementos individuales son todos ellos de un tipo bien conocido en la técnica, por ejemplo de la gama Nokia de teléfonos móviles.

55 La Figura 5 es una representación de los diversos componentes implicados en la implementación de una forma de realización de la presente invención. La Figura 5 muestra una memoria intermedia de datos entrelazados 200, una memoria intermedia de datos comprimidos 210, y una memoria intermedia de datos no comprimidos 220. Debería observarse que o bien la memoria intermedia de datos comprimidos 210 o bien la memoria intermedia de datos no comprimidos 220 pueden usar la misma área de memoria que la memoria intermedia de datos entrelazados 200, si los datos originales ya no son necesarios después de la transmisión y se pueden sobrescribir. Cada una de la memoria intermedia de datos entrelazados 200, la memoria intermedia de datos comprimidos 210 y la memoria intermedia de datos no comprimidos 220 se puede incluir en la memoria 158 del dispositivo electrónico mostrado en las Figuras 3 y 4. El sistema incluye también una MPU 230 y hardware de DMA 240. La MPU 230, que puede comprender el controlador 16 del sistema mostrado en las Figuras 4, se usa para ejecutar el software del sistema, así como para llevar a cabo las operaciones especificadas por el software del sistema. El hardware de DMA 240 es un controlador de hardware que tiene la capacidad de mover datos desde un lugar de la memoria a otro lugar de la memoria de manera independiente con respecto a la MPU 230. Debería observarse que la MPU 230, el hardware de

DMA 240 y la unidad de memoria 14 pueden comprender componentes independientes, o los mismos se pueden combinar en un único circuito integrado.

La Figura 6 es un diagrama de flujo que muestra la implementación de una forma de realización de la presente invención. La etapa 300 de la Figura 6 representa el punto inicial para el desentrelazado de una unidad de datos. En este punto, se ha capturado una imagen y la memoria intermedia de datos entrelazados 200 contiene la unidad de datos entrelazados original. Por consiguiente, la memoria intermedia de datos comprimidos 210 y la memoria intermedia de datos no comprimidos 220 se han reservado para almacenar los datos comprimidos y datos de vista previa de manera concatenada. Aquellas configuraciones de DMA que no cambian en el transcurso de la operación también se materializan en este punto.

En la etapa 310 de la Figura 6, y suponiendo que la unidad de datos comienza con datos comprimidos, la MPU 230 comienza a examinar la unidad de datos entrelazados desde el comienzo de la unidad de datos. La MPU 230 busca un marcador que indica dónde comienza el siguiente fragmento de datos de vista previa. Mientras se lleva a cabo esta acción, la MPU 230 copia los datos comprimidos en la memoria intermedia de datos comprimidos 210 hasta el marcador, aunque sin incluirlo, que indica el comienzo de los datos de vista previa. Debería observarse que, si la unidad de datos comienza con datos de vista previa, esta etapa se omite.

En la etapa 320, se inicializa el hardware de DMA 240 para llevar a cabo una copia de memoria-a-memoria desde el comienzo recién encontrado de los datos de lista previa, a la memoria intermedia de datos no comprimidos 220. Debería observarse que los marcadores no están incluidos en esta copia de memoria. El tamaño de la transferencia de DMA es conocido, como lo es el tamaño fijo del fragmento de datos de vista previa.

En la etapa 330 y a medida que el hardware de DMA 240 está transfiriendo datos en segundo plano, la MPU 230 omite el fragmento de datos de vista previa (ya que su tamaño era conocido) y comienza a examinar los datos que suceden al fragmento de vista previa que se ha omitido en este momento. En caso de que se espere el marcador opcional para datos comprimidos y el mismo no se encuentre, o en caso de que el marcador opcional no se use, pero haya más datos a procesar, la MPU 230 continuará examinando los datos hasta que se encuentre un marcador para datos de vista previa. Mientras se lleva a cabo esta acción, la MPU 230 copia los datos comprimidos en la memoria intermedia de datos comprimidos 210, colocándolos inmediatamente después de los datos que se movieron previamente a la memoria intermedia de datos comprimidos 210.

En la etapa 340, o bien se puede suponer que el hardware de DMA 240 ha acabado de copiar el fragmento de datos de imagen que se estaba copiando en la etapa 320, o bien el sistema puede comprobar específicamente el estado del hardware de DMA 240 y puede esperar hasta que finalice la transferencia. En cualquiera de los casos anteriores, el hardware de DMA 240 se reinicializa para llevar a cabo una copia de memoria-a-memoria para el siguiente fragmento de datos de vista previa, copiando los datos a la memoria intermedia de datos no comprimidos 220, y colocándolos inmediatamente después de los datos que se movieron previamente a la memoria intermedia de datos no comprimidos 220.

Debería observarse que, debido a que el hardware de DMA está diseñado en particular para mover datos de manera rápida, y puesto que la MPU debe examinar adicionalmente los datos que está copiando, el hardware de DMA 240 debería poder procesar más datos en el mismo periodo de tiempo que la MPU 230. En caso de que exista un riesgo de que no se acabe la transferencia de DMA en el instante en el que debería iniciarse una transferencia de DMA nueva, una de las soluciones alternativas implica el uso de dos o más canales de DMA de manera que se realicen siempre solicitudes de DMA consecutivas para canales diferentes.

En la etapa 350, se determina si se ha copiado la unidad de datos completa. Si se ha copiado la unidad de datos completa, entonces en la etapa 360 se puede visualizar la imagen de vista previa, y los datos comprimidos se guardan en la memoria intermedia de datos comprimidos 210. Si no se ha copiado la unidad de datos completa, entonces las etapas 330 y 340 se repiten hasta que la unidad de datos completa haya sido procesada y colocada en la memoria intermedia de datos comprimidos 210 y la memoria intermedia de datos no comprimidos 230, respectivamente.

Tal como se ha mencionado previamente, el proceso representado en la Figura 6 también puede funcionar en la situación en la que la memoria intermedia de datos comprimidos 210 es la misma que la memoria intermedia de datos entrelazados 200. Debería observarse, que si la unidad de datos se ordena tal como se representa en la Figura 2, la MPU 230 comenzará a sobrescribir el primer fragmento de datos de vista previa aproximadamente al mismo tiempo que el hardware de DMA 240 está transfiriendo ese fragmento de datos a la memoria intermedia de datos no comprimidos 220. No obstante, en la mayoría de casos, se puede suponer ciertamente que el hardware de DMA 240 funciona más rápido que la MPU 230, en la medida en la que el hardware de DMA 240 puede copiar a ciegas los datos y no tiene que consumir tiempo alguno analizándolos. Por lo tanto, el hardware de DMA 240 puede leer los datos de vista previa antes de que los mismos se sobrescriban con datos comprimidos del siguiente fragmento.

Puesto que el tamaño de la transferencia de DMA es siempre el mismo, la inicialización de una transferencia de DMA nueva puede que requiera solamente dos acciones después de que se realice la configuración inicial. La primera acción es configurar la dirección de inicio de lectura de DMA, mientras que la segunda acción es habilitar la transferencia de DMA. Esto supone que el hardware de DMA 240 contiene un puntero de escritura que se incrementa automáticamente mientras progresa la transferencia. Por lo tanto, el puntero de escritura apunta automáticamente a la posición correcta de la memoria intermedia de datos no comprimidos 220 cuando se copia el siguiente fragmento de datos de vista previa.

En una forma de realización alternativa de la invención, hay fragmentos de datos de más de dos tipos diferentes. En esta situación, la MPU 230 puede gestionar todos los fragmentos cuyo tamaño puede variar, mientras que el hardware de DMA 240 continúa ocupándose de todos los fragmentos de tamaño fijo. Se puede usar más de un canal de DMA en el hardware de DMA 240 si hay más de un tipo de fragmentos con una longitud fija. Estos canales múltiples se usan con el fin de garantizar que no hay necesidad de esperar a que se complete una transferencia de DMA antes de que pueda comenzar la siguiente transferencia, ya que cada canal de DMA puede funcionar de manera independiente con respecto a otros canales.

Debería observarse que se puede usar una amplia variedad de marcadores de fragmentos conjuntamente con la presente invención, y que la presente invención no se limita a ningún tipo específico de marcador. Adicionalmente, aun cuando se han usado marcadores de fragmentos en el comienzo de cada fragmento en el proceso representado en la Figura 6, los marcadores también se podrían usar al final de cada fragmento o tanto en el comienzo como al final de cada fragmento.

La presente invención se describe en el contexto general de etapas de método, las cuales se pueden implementar en una forma de realización por medio de un producto de programa que incluye instrucciones ejecutables por ordenador, tales como código de programa, ejecutados por ordenadores en entornos en red.

En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etcétera, que llevan a cabo tareas particulares o implementan tipos particulares de datos abstractos. Las instrucciones ejecutables por ordenador, las estructuras de datos asociadas, y los módulos de programa representan ejemplos de código de programa para ejecutar etapas de los métodos dados a conocer en la presente. La secuencia particular de dichas instrucciones ejecutables o estructuras de datos asociadas representa ejemplos de acciones correspondientes para implementar las funciones descritas en dichas etapas.

Las implementaciones de software y web de la presente invención se podrían lograr con técnicas de programación convencionales con una lógica basada en reglas y otra lógica destinada a materializar las diversas etapas de búsqueda en bases de datos, etapas de correlación, etapas de comparación y etapas de decisión. Debería indicarse también que los vocablos "componente" y "módulo" tal como se usan en la presente, y en las reivindicaciones, están destinados a abarcar implementaciones que usan una o más líneas de código de software, y/o implementaciones de hardware, y/o equipos para recibir entradas manuales.

La descripción anterior de formas de realización de la presente invención se ha presentado a título ilustrativo y descriptivo. No está destinada a ser exhaustiva o a limitar la presente invención a la forma precisa dada a conocer, y son posibles modificaciones y variantes teniendo en cuenta las enseñanzas anteriores o las mismas se pueden asimilar al llevar a la práctica la presente invención. Las formas de realización se seleccionaron y describieron con el fin de explicar los principios de la presente invención y su aplicación práctica para permitir que los versados en la técnica utilicen la presente invención en diversas formas de realización y con varias modificaciones según resulte adecuado para el uso particular contemplado.

REIVINDICACIONES

1. Método de procesamiento de una unidad de datos entrelazados que incluye una pluralidad de fragmentos de datos de un primer tipo y una pluralidad de fragmentos de datos de un segundo tipo, que comprende:

usar una unidad de microprocesador (230) para copiar un primer fragmento de datos del primer tipo en la unidad de datos a una primera memoria intermedia (210);

usar hardware de acceso directo a memoria (240) para copiar un primer fragmento de datos del segundo tipo en la unidad de datos a una segunda memoria intermedia (220);

al completarse la copia del primer fragmento de datos del primer tipo, usar la unidad de microprocesador (230) para copiar el siguiente fragmento de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210);

después de completarse la copia del primer fragmento de datos del segundo tipo, usar el hardware de acceso directo a memoria (240) para copiar el siguiente fragmento de datos del segundo tipo en la unidad de datos a la segunda memoria intermedia (220); y

continuar usando la unidad de microprocesador (230) y el hardware de acceso directo a memoria (240) para copiar fragmentos de datos subsiguientes del primer tipo y el segundo tipo, respectivamente, de la unidad de datos a la primera memoria intermedia (210) y la segunda memoria intermedia (220) hasta que se hayan copiado todos los fragmentos de datos de la unidad de datos.

2. Método según la reivindicación 1, que comprende por lo menos una de entre:

la pluralidad de fragmentos de datos del primer tipo comprende fragmentos de datos comprimidos,

la pluralidad de fragmentos de datos del primer tipo comprende fragmentos de datos que tienen longitudes no uniformes,

la pluralidad de fragmentos de datos del segundo tipo comprende fragmentos de datos no comprimidos.

3. Método según la reivindicación 2, en el que los fragmentos de datos no comprimidos comprenden datos de vista previa de imágenes.

4. Método según la reivindicación 3, que comprende además, después de que todos los fragmentos de datos del segundo tipo en la unidad de datos se hayan copiado a la segunda memoria intermedia (220), usar los datos de vista previa de imágenes para visualizar una imagen para un usuario.

5. Método según cualquiera de las reivindicaciones anteriores, en el que la copia de fragmentos de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210) incluye usar la unidad de microprocesador (230) para buscar un marcador que indique dónde comienza el siguiente fragmento de datos no copiados del primer tipo.

6. Método según cualquiera de las reivindicaciones anteriores, en el que la pluralidad de fragmentos de datos del primer tipo y el segundo tipo se copia a partir de una memoria intermedia de datos entrelazados (200).

7. Método según la reivindicación 6, en el que la memoria intermedia de datos entrelazados (200) comprende la primera memoria intermedia de datos (210), y siendo los fragmentos de datos individuales del segundo tipo sobrescritos en la memoria intermedia de datos entrelazados (200) durante la copia de fragmentos de datos del primer tipo.

8. Método según cualquiera de las reivindicaciones anteriores, en el que la unidad de datos se genera a partir de la captura de una imagen por parte de un módulo de cámara.

9. Producto de programa de ordenador para procesar una unidad de datos entrelazados que incluye una pluralidad de fragmentos de datos de un primer tipo y una pluralidad de fragmentos de datos de un segundo tipo, que comprende:

un código de ordenador para usar una unidad de microprocesador (230) con el fin de copiar un primer fragmento de datos del primer tipo en la unidad de datos a una primera memoria intermedia (210);

un código de ordenador para usar hardware de acceso directo a memoria (240) con el fin de copiar un primer fragmento de datos del segundo tipo en la unidad de datos a una segunda memoria intermedia (220);

un código de ordenador para, al completarse la copia del primer fragmento de datos del primer tipo, usar la unidad de microprocesador (230) para copiar el siguiente fragmento de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210);

5 un código de ordenador para, después de completarse la copia del primer fragmento de datos del segundo tipo, usar el hardware de acceso directo a memoria (240) para copiar el siguiente fragmento de datos del segundo tipo en la unidad de datos a la segunda memoria intermedia (220); y

10 un código de ordenador para continuar usando la unidad de microprocesador (230) y el hardware de acceso directo a memoria (240) con el fin de copiar fragmentos de datos subsiguientes del primer tipo y el segundo tipo, respectivamente, de la unidad de datos a la primera memoria intermedia (210) y la segunda memoria intermedia (220) hasta que se hayan copiado todos los fragmentos de datos en la unidad de datos.

10. Producto de programa de ordenador según la reivindicación 9, que comprende por lo menos una de entre:

15 la pluralidad de fragmentos de datos del primer tipo comprende fragmentos de datos comprimidos,

la pluralidad de fragmentos de datos del primer tipo comprende fragmentos de datos que tienen longitudes no uniformes,

20 la pluralidad de fragmentos de datos del segundo tipo comprende fragmentos de datos no comprimidos.

11. Producto de programa de ordenador según la reivindicación 9 o 10, que comprende además un código de ordenador para, después de que todos los fragmentos de datos del segundo tipo en la unidad de datos se hayan copiado a la segunda memoria intermedia (220), usar los datos no comprimidos para visualizar una imagen a un usuario.

12. Producto de programa de ordenador según cualquiera de las reivindicaciones 9 a 11, en el que la copia de fragmentos de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210) incluye usar la unidad de microprocesador (230) para buscar un marcador que indique dónde comienza el siguiente fragmento de datos no copiados del primer tipo.

13. Producto de programa de ordenador según cualquiera de las reivindicaciones 9 a 12, en el que la pluralidad de fragmentos de datos del primer tipo y el segundo tipo se copia a partir de una memoria intermedia de datos entrelazados (200).

14. Producto de programa de ordenador según la reivindicación 13, en el que la memoria intermedia de datos entrelazados (200) comprende la primera memoria intermedia de datos (210), y siendo los fragmentos de datos individuales del segundo tipo sobrescritos en la memoria intermedia de datos entrelazados (200) durante la copia de fragmentos de datos del primer tipo.

15. Dispositivo electrónico, que comprende:

45 una unidad de microprocesador (230);

una unidad de memoria conectada operativamente a la unidad de microprocesador (230), incluyendo la unidad de memoria una primera memoria intermedia (210) y una segunda memoria intermedia (220); y

50 hardware de acceso directo a memoria (240) conectado operativamente a la unidad de memoria,

incluyendo la unidad de memoria un producto de programa de ordenador para procesar una unidad de datos entrelazados que incluye una pluralidad de fragmentos de datos de un primer tipo y una pluralidad de fragmentos de datos de un segundo tipo, que comprende:

55 un código de ordenador para usar la unidad de microprocesador (230) con el fin de copiar un primer fragmento de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210);

un código de ordenador para usar el hardware de acceso directo a memoria (240) con el fin de copiar un primer fragmento de datos del segundo tipo en la unidad de datos a la segunda memoria intermedia (220);

60 un código de ordenador para, al completarse la copia del primer fragmento de datos del primer tipo, usar la unidad de microprocesador (230) para copiar el siguiente fragmento de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210);

un código de ordenador para, después de completarse la copia del primer fragmento de datos del segundo tipo, usar el hardware de acceso directo a memoria (240) para copiar el siguiente fragmento de datos del segundo tipo en la unidad de datos a la segunda memoria intermedia (220); y

un código de ordenador para continuar usando la unidad de microprocesador (230) y el hardware de acceso directo a memoria (240) con el fin de copiar fragmentos de datos subsiguientes del primer tipo y el segundo tipo, respectivamente, de la unidad de datos a la primera memoria intermedia (210) y la segunda memoria intermedia (220) hasta que se hayan copiado todos los fragmentos de datos de la unidad de datos.

16. Dispositivo electrónico de la reivindicación 15, que comprende por lo menos una de entre:

la pluralidad de fragmentos de datos del primer tipo comprende fragmentos de datos comprimidos,

la pluralidad de fragmentos de datos del primer tipo comprende fragmentos de datos que tienen longitudes no uniformes,

la pluralidad de fragmentos de datos del segundo tipo comprende fragmentos de datos no comprimidos.

17. Dispositivo electrónico de la reivindicación 15 o 16, en el que el producto de programa de ordenador comprende además un código de ordenador para, después de que todos los fragmentos de datos del segundo tipo en la unidad de datos se hayan copiado a la segunda memoria intermedia (220), usar los datos no comprimidos para visualizar una imagen a un usuario.

18. Dispositivo electrónico según la reivindicación 15, 16, o 17, en el que la copia de fragmentos de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210) incluye usar la unidad de microprocesador (230) para buscar un marcador que indique dónde comienza el siguiente fragmento de datos no copiados del primer tipo.

19. Dispositivo electrónico según la reivindicación 15, 16, 17, o 18, en el que la pluralidad de fragmentos de datos del primer tipo y el segundo tipo se copia a partir de una memoria intermedia de datos entrelazados (200).

20. Dispositivo electrónico según la reivindicación 19, en el que la memoria intermedia de datos entrelazados comprende la primera memoria intermedia de datos (210), y siendo los fragmentos de datos individuales del segundo tipo sobrescritos en la memoria intermedia de datos entrelazados durante la copia de fragmentos de datos del primer tipo.

21. Circuito integrado, que comprende:

una parte de microprocesador;

una parte de memoria conectada operativamente a la parte de multiprocesador, incluyendo la parte de memoria una primera memoria intermedia (210) y una segunda memoria intermedia (220); y

una parte de hardware de acceso directo a memoria (240) conectada operativamente a la parte de memoria,

incluyendo la parte de memoria un producto de programa de ordenador para procesar una unidad de datos entrelazados que incluye una pluralidad de fragmentos de datos de un primer tipo y una pluralidad de fragmentos de datos de un segundo tipo, que comprende:

un código de ordenador para usar la parte de microprocesador con el fin de copiar un primer fragmento de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210);

un código de ordenador para usar la parte de hardware de acceso directo a memoria (240) con el fin de copiar un primer fragmento de datos del segundo tipo en la unidad de datos a la segunda memoria intermedia (220);

un código de ordenador para, al completarse la copia del primer fragmento de datos del primer tipo, usar la parte de microprocesador para copiar el siguiente fragmento de datos del primer tipo en la unidad de datos a la primera memoria intermedia (210);

un código de ordenador para, después de completarse la copia del primer fragmento de datos del segundo tipo, usar la parte de hardware de acceso directo a memoria (240) para copiar el siguiente fragmento de datos del segundo tipo en la unidad de datos a la segunda memoria intermedia (220); y

un código de ordenador para continuar usando la parte de microprocesador y la parte de hardware de acceso directo a memoria (240) con el fin de copiar fragmentos de datos subsiguientes del primer tipo y el segundo

tipo, respectivamente, de la unidad de datos a la primera memoria intermedia (210) y la segunda memoria intermedia (220) hasta que se hayan copiado todos los fragmentos de datos en la unidad de datos.

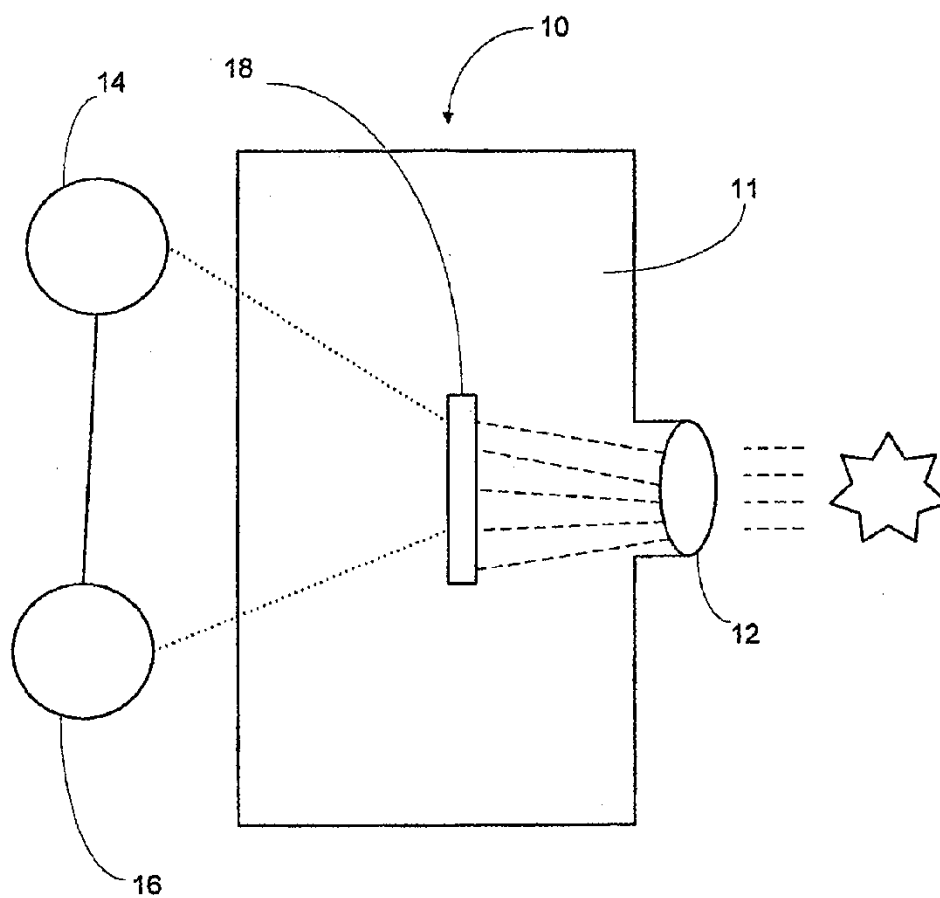


FIG. 1



FIG. 2

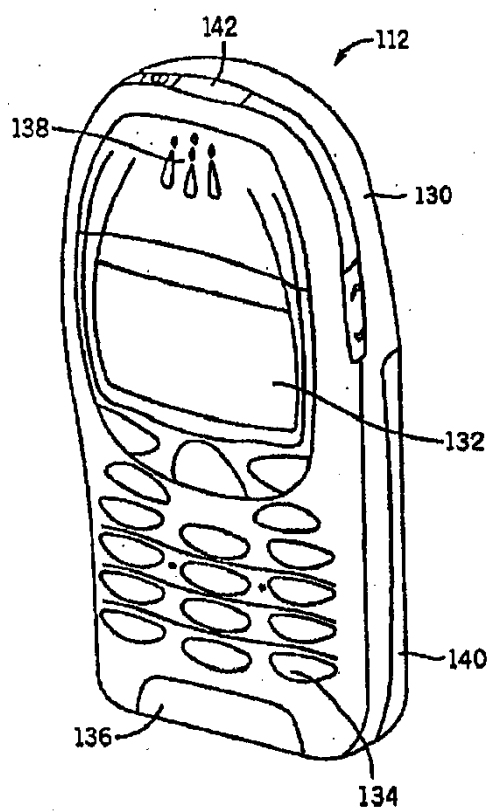


FIG. 3

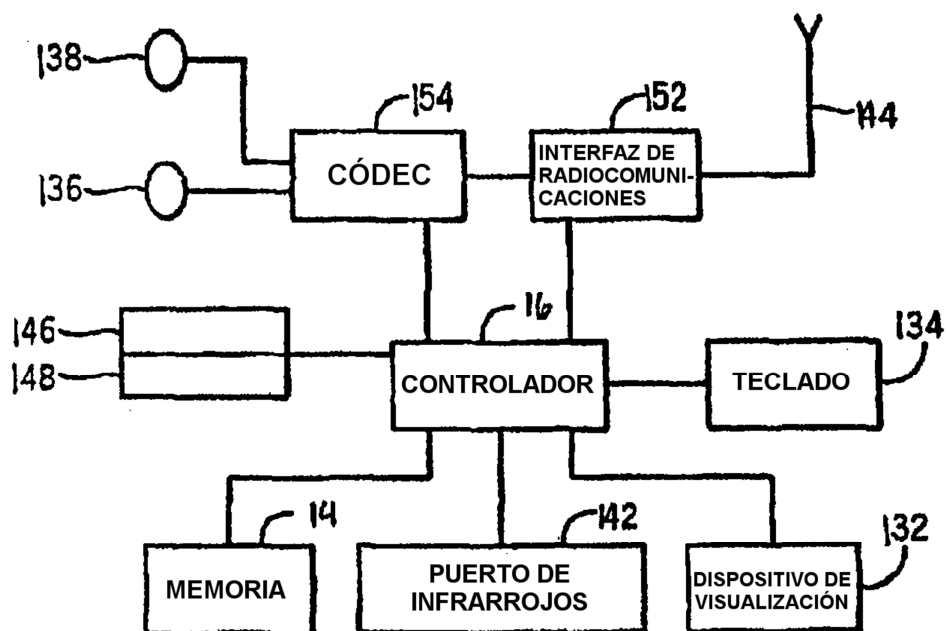


FIG. 4

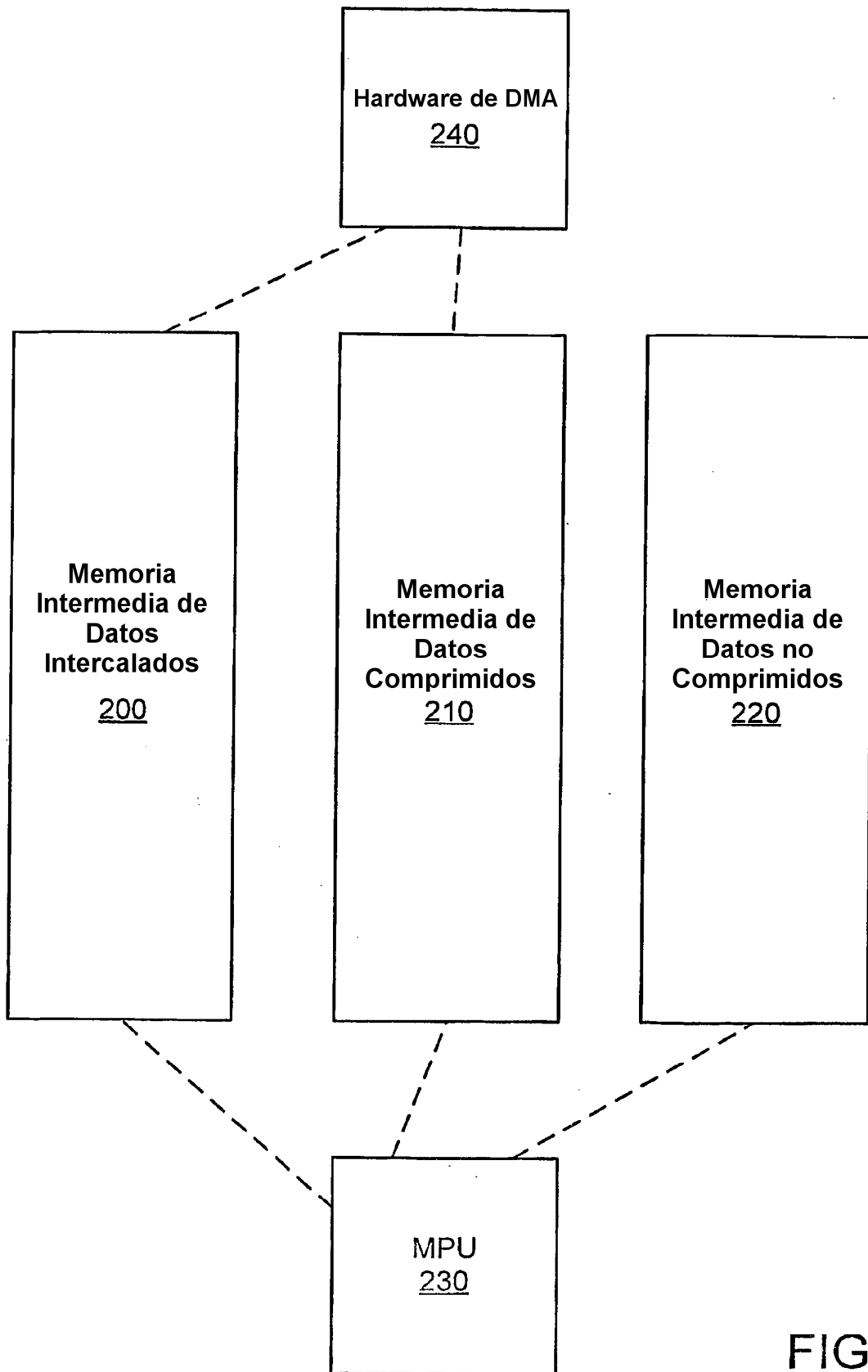


FIG. 5

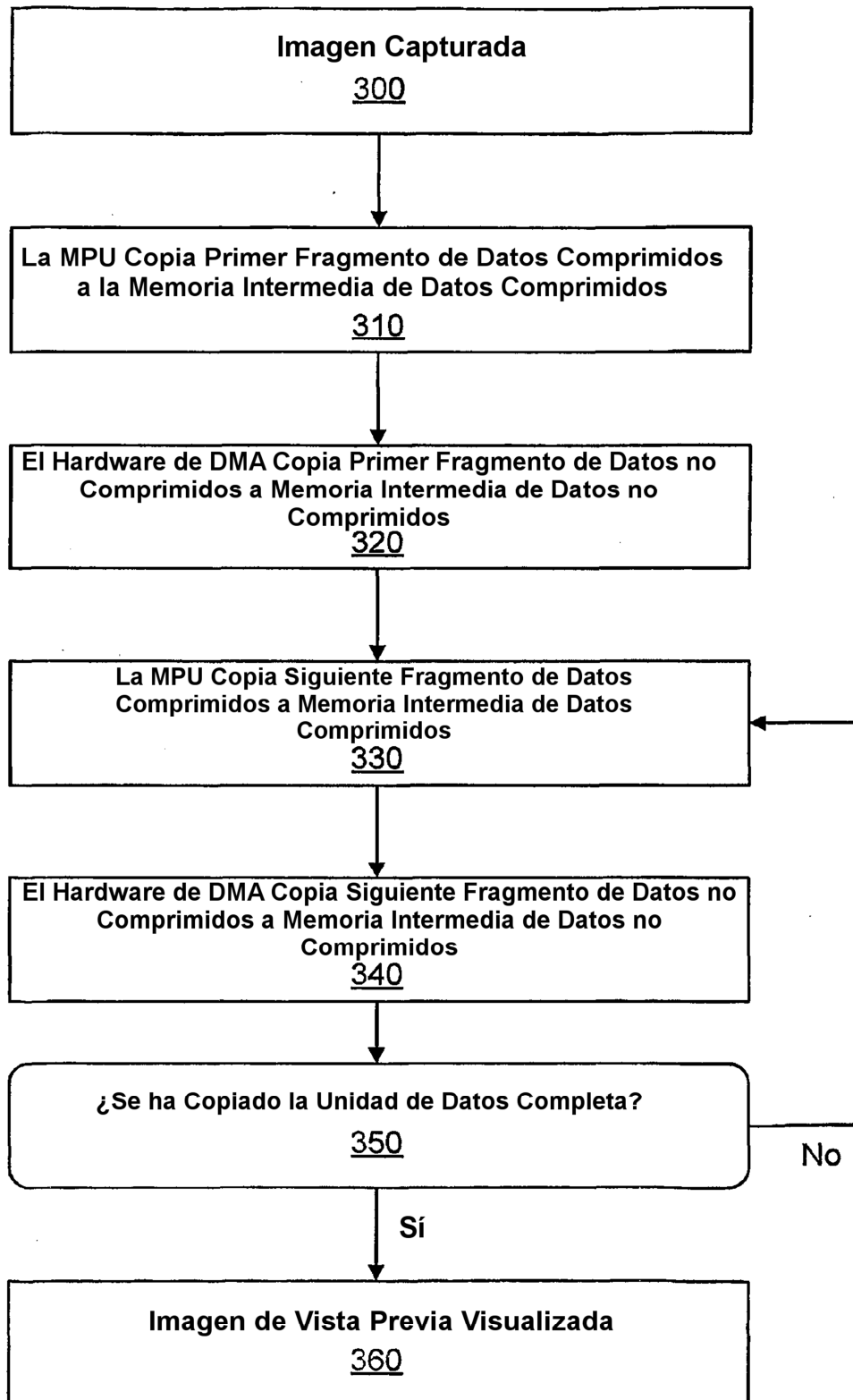


FIG. 6