

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 378 357**

51 Int. Cl.:
G06T 15/08 (2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **02725716 .1**
96 Fecha de presentación: **17.04.2002**
97 Número de publicación de la solicitud: **1381998**
97 Fecha de publicación de la solicitud: **21.01.2004**

54 Título: **Dispositivo de renderizado de cuerpos volumétricos**

30 Prioridad:
18.04.2001 US 284716 P

45 Fecha de publicación de la mención BOPI:
11.04.2012

45 Fecha de la publicación del folleto de la patente:
11.04.2012

73 Titular/es:
LANDMARK GRAPHICS CORPORATION
2107 CityWest Blvd., Bldg. 2
Houston TX 77042, US

72 Inventor/es:
CALLEGARI, Andres, C.

74 Agente/Representante:
Curell Aguilá, Mireia

ES 2 378 357 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

Dispositivo de renderizado de cuerpos volumétricos.

5 **Antecedentes de la invención**1. Campo de la invención

10 La presente invención se refiere en general al procesamiento, la extracción de modelos, el perfeccionamiento de modelos y el renderizado (proceso de generación de imágenes en 3D) gráfico de datos digitales que representan volúmenes geológicos o de otro tipo y, más concretamente, al renderizado y al análisis de volúmenes de formas irregulares en los datos volumétricos basados en vóxeles.

15 2. Descripción de la técnica relacionada

Los geólogos, los geofísicos y los profesionales de otras disciplinas analizan los datos sísmicos para fines tales como la detección de la presencia y el cambio a lo largo del tiempo de los yacimientos de hidrocarburos u otras características del subsuelo. Se pueden recopilar datos sísmicos, por ejemplo, creando explosiones o liberando de alguna manera energía en el suelo y detectando y digitalizando la energía sísmica reflejada con una serie de geófonos o sensores similares. Los datos sísmicos volumétricos procesados representan una zona de subsuelo en tres dimensiones (3D), que por lo general se expresa en unidades de tiempo o de profundidad. Otros ejemplos de formas de recopilación y utilización de dichos datos volumétricos comprenden la medición gravitacional y magnética. Los datos pueden comprender cualquiera de los numerosos atributos que los profesionales del ámbito de la técnica consideran utilizables o deducibles a partir de las mediciones de la energía sísmica reflejada y mediciones sobre el terreno, el más común de los cuales tal vez sea la amplitud de las señales reflejadas.

Los conjuntos de datos de volumen 3D recopilados pueden interpretarse, almacenarse y manipularse en un formato cualquiera de los diversos posibles. Cada vez es más común un formato en el que cada dato representa un volumen. Dicho dato se denomina vóxel ("elemento de volumen") o celda. Por ejemplo, si la amplitud es el atributo que caracteriza a los datos recopilados, las muestras de atributos están representadas por vóxeles, cada uno de los cuales está caracterizado por una amplitud. En otras palabras, el conjunto de datos se compone de una pluralidad de vóxeles, cada uno de los cuales está caracterizado por una amplitud. Un conjunto de datos sísmicos de volumen comúnmente comprende millones o incluso miles de millones de vóxeles y puede requerir una capacidad de almacenamiento de datos del orden de los terabytes. Los formatos de vóxel se utilizan comúnmente no sólo en el análisis de datos sísmicos, sino también en imágenes médicas y otras disciplinas de formación de imágenes.

El análisis de los datos volumétricos normalmente conlleva unas etapas de renderizado, interpretación y perfeccionamiento para obtener un modelo del subsuelo o renderizar una vista 3D particular de la zona del subsuelo. La mayoría de motores gráficos 3D disponibles en el mercado (por ejemplo, tarjetas aceleradoras de gráficos para ordenadores de estaciones de trabajo) no presentan primitivas de vóxel, sino que solo pueden interpretar puntos, líneas y polígonos tales como triángulos, debido a que están diseñados para renderizar representaciones basadas en superficies de objetos 3D, es decir, armazones vacíos, en lugar de objetos que comprenden vóxeles. Aunque existen algunos aceleradores de gráficos basados en vóxeles, estos no pueden combinar de manera eficiente y rentable las primitivas 2D y 3D, que son necesarias para implementar ciertas características de visualización, tales como las transformaciones de Billboard de tipo esférico y animación.

Por lo tanto, los procedimientos de renderizado de datos de vóxeles conocidos, tales como el "raycasting" y el "splatting", sólo generan imágenes instantáneas en 3D de cierta perspectiva de visión predeterminada. En el raycasting, los rayos se proyectan desde el origen del observador, el punto de visión del usuario o a través de un plano de proyección y se extienden hasta que interseccionan con un punto de datos o una serie de puntos de datos. Los puntos situados a lo largo y alrededor de los rayos se procesan, y la imagen resultante se dibuja en la pantalla o plano de proyección. En la técnica de splatting, se calcula la contribución a la imagen final de cada vóxel de una vecindad determinada. Se "lanzan" vóxeles conceptualmente o virtualmente sobre el plano de la imagen, de tal forma que cada vóxel del espacio del objeto deja una huella en el espacio de la imagen. Tanto si se utilizan para renderizar representaciones basadas en superficies de objetos tridimensionales u objetos tridimensionales reales, es decir basadas en vóxeles, estos procedimientos sólo pueden renderizar una aproximación por vóxel plana, es decir, de forma circular o cuadrada sin ninguna vecindad entre vóxeles definida, y están estrictamente subordinadas a la visión. Debe tenerse en cuenta también que todos los procedimientos de renderizado de datos de vóxeles conocidos consumen muchos recursos de software o hardware y requieren elementos de hardware costosos y especializados que adolecen de diversas limitaciones de rendimiento y visualización. Además, el rendimiento de muchos de estos algoritmos se ve afectado por la forma en que recorren la memoria. Cuando la memoria se recorre en diferentes direcciones, el rendimiento del algoritmo puede variar según la eficacia del acceso a los datos del hardware. Cuanto más grande es el volumen de datos de trabajo menor es la capacidad de utilizar con eficacia las cachés de hardware.

Además, sea cual sea el procedimiento de renderizado basado en vóxeles que se utiliza, el conjunto de datos de

volumen completo debe retenerse en memoria (hardware) y, finalmente, renderizarse mediante hardware para gráficos 3D. Para renderizar un volumen de datos, es necesario recuperar/traspasar el conjunto de datos completo desde la memoria de datos (es decir, memoria de acceso aleatorio, memoria en disco, etc.), renderizarlo mediante algoritmos que exigen el consumo de muchos recursos informáticos y enviarlo al hardware para gráficos 3D, que normalmente es limitado en cuanto a recursos. Como resultado de estas limitaciones, el renderizado puede llevar una cantidad de tiempo considerable, dependiendo del hardware utilizado. La calidad del renderizado, las opciones de visualización interactiva y respuesta interactiva a la visualización de última generación suscitan un gran interés entre profesionales tales como los geólogos y los geofísicos, que quizá necesiten ver un modelo desde muchas direcciones o puntos de visión diferentes, comparar diferentes modelos entre sí y manipularlos durante el análisis de las características del subsuelo en relación con los yacimientos de petróleo y gas.

Comúnmente es deseable identificar, aislar y centrarse en las características específicas y/o zonas anómalas de un volumen, tales como las relativas a yacimientos potenciales de petróleo y gas en los volúmenes sísmicos o las relativas a órganos, huesos y tumores en los volúmenes médicos. Dichas características pueden calificarse de volúmenes irregulares, porque no tienen una forma regular o predecible. Los sistemas de raycasting, point splatting e incluso los sistemas genuinos de renderizado basados en vóxeles no pueden separar o trabajar fácilmente con dichos volúmenes irregulares por separado del volumen de datos de trabajo completo, porque en la mayoría de casos (es decir, cuando se utiliza hardware 3D o texturizado 2D) el volumen de datos de trabajo completo finalmente se envía y renderiza en su totalidad mediante el hardware para gráficos 3D. La imagen creada por el hardware 3D puede aislar gráficamente y de forma parcial algunas características de la pantalla, aunque básicamente se trata de una vista efímera o instantánea del volumen de datos desde una única dirección específica. Es decir, aunque el usuario pueda ver una característica en la imagen, la característica no se representa por separado en la lógica del ordenador, independientemente de los datos de vecindad. Puede ser, sin embargo, que el usuario esté interesado solo en un volumen irregular o un grupo de volúmenes irregulares concreto y no en el volumen en su conjunto, o que esté interesado en realizar alguna operación de correlación de atributos concreta o ver solo un volumen irregular o un conjunto de volúmenes irregulares concreto. Por ejemplo, tal vez se desee tomar en consideración una serie de imágenes, cada una de las cuales representa el mismo volumen irregular o grupo de volúmenes irregulares en momentos diferentes, para evaluar cómo pueden haber cambiado los volúmenes irregulares a lo largo del tiempo, por ejemplo, a medida que se ha ido agotando el petróleo o el gas. Con los sistemas de renderizado convencionales, es necesario renderizar cada conjunto de datos de volumen en su totalidad para ver qué aspecto tenía el volumen irregular deseado representado dentro del mismo en el momento en que se recopiló el conjunto de datos.

Los profesionales de la técnica han intentado superar la incapacidad de trabajar con volúmenes irregulares individuales realizando una laboriosa identificación de dichos volúmenes por medio de las vistas instantáneas y extrayendo cada uno de ellos a partir de conjuntos de datos de volumen. Este procedimiento puede llegar a ser muy desagradable y tedioso cuando hay mucho ruido en los datos o si los objetos son numerosos y de pequeño tamaño. Por ejemplo, se han propuesto algoritmos mediante los cuales, utilizando diversas vistas instantáneas desde diversos ángulos, se identifica visualmente un punto semilla o vóxel semilla perteneciente a un volumen irregular y, a continuación, se utiliza software personalizado independiente para extraer todos los vóxeles vecinos que presumiblemente pertenecen al mismo volumen irregular. Sin embargo, la utilización de dicho procedimiento y flujo de trabajo para hallar un gran número de volúmenes irregulares resulta una tarea ineficaz, laboriosa y lenta. El problema se agrava mucho si lo que se desea es hallar todos los volúmenes irregulares de un conjunto de datos de muy gran tamaño. Por otra parte, no se ha propuesto ningún medio práctico y eficaz para la manipulación o el análisis adicional de los volúmenes irregulares extraídos de esta manera.

Sería conveniente disponer de un sistema eficiente para identificar, renderizar y realizar otras tareas con los volúmenes irregulares individuales representados en uno o más conjuntos de datos de volumen. La presente invención trata acerca de estos problemas y estas deficiencias y otras cuestiones de la manera descrita a continuación.

En el documento de Gargantini *et al.* "Adaptive Display of Linear Octrees", Computers and Graphics, vol. 13, nº 3, pp. 337-343, 1989, se describe un algoritmo de visualización de renderizado de volúmenes que transforma un objeto sólido almacenado en forma de árbol "octree" lineal en una secuencia de coordenadas de píxeles de salida en un espacio de imagen de dos dimensiones. Se utiliza una geometría de proyección frontal para la presentación de las superficies.

Sumario de la invención

La presente invención se refiere a la identificación, el almacenamiento, el renderizado gráfico y otras operaciones volumétricas de volúmenes irregulares representados en formato de vóxeles o celdas tridimensionales. Los datos volumétricos de entrada pueden ser, por ejemplo, los que se recopilan en estudios geológicos en los que se desea identificar y trabajar con volúmenes irregulares que representan yacimientos de hidrocarburos, estructuras de sal o características geológicas relacionadas.

Para identificar un volumen irregular, el usuario introduce o establece de algún modo unos criterios que describen las propiedades de un volumen irregular deseado, y cada conjunto de datos de volumen se introduce y procesa

5 sometiéndolo a una búsqueda de volúmenes irregulares que satisfacen dichos criterios. El procesamiento implica la búsqueda automática de un vóxel semilla o una celda semilla que cumpla los criterios y por lo tanto pertenezca al volumen irregular deseado y, a continuación, la identificación de las celdas relacionadas con la celda semilla por una o más relaciones predeterminadas, celdas que por consiguiente también deberán agruparse en ese volumen irregular. Las relaciones predeterminadas pueden establecer, por ejemplo, que las celdas del volumen irregular lindan con otras celdas del volumen irregular o están a una distancia predeterminada de algún punto predeterminado, que las celdas del volumen irregular cumplen condiciones de wavelet predeterminadas, que las celdas pertenecen a un volumen irregular de un rango de tamaños predeterminado o cualquier otra relación predeterminada adecuada. La información que indica que cada una de dichas celdas está relacionada con otras celdas y pertenece a un volumen irregular puede ser de cualquier tipo adecuado y se almacena en una estructura de datos conveniente. También se almacena la ubicación o información de vecindad similar y otro tipo de datos que describen las propiedades o los atributos de la celda identificada.

15 No es necesario procesar el volumen de entrada en su totalidad de la forma descrita anteriormente. Por el contrario, el usuario puede indicar por sí mismo o especificar de alguna otra forma una extensión que abarca solo una parte del volumen de entrada. La división de un volumen de entrada en dichas partes o fragmentos permite el procesamiento independiente de estos (pudiendo comprender dicho "procesamiento" la visualización), por ejemplo, en un procesamiento paralelo, con varios subprocesos y varios canales o un sistema informático multitarea similar, o su procesamiento mediante otros sistemas informáticos o en momentos diferentes.

20 Debido a que es posible acceder a las estructuras de datos de la celda de forma individual, aleatoria y eficaz, las partes seleccionadas de los volúmenes irregulares hallados pueden renderizarse en una pantalla o someterse a otras operaciones, tales como la comparación mutua, la fusión, la división en volúmenes irregulares adicionales, la aplicación de filtros, la utilización como plantilla para aplicar operaciones a otro conjunto de volúmenes irregulares y cualquier otra operación booleana, aritmética y algorítmica adecuada. Cada vóxel o celda contiene toda la información necesaria para su renderizado completo. Los procedimientos convencionales que comprenden solo el renderizado, tales como el raycasting y el splatting, pueden modificarse para utilizar las nuevas estructuras de datos de la presente invención, pero debido a que la ubicación y otras características de cada vóxel en el volumen irregular o la parte del mismo que se va a renderizar están predeterminadas, es posible simplificar desde el punto de vista informático dichos procedimientos de renderizado convencionales, con lo cual estos pueden ejecutarse con mayor rapidez y eficacia.

35 Además, los volúmenes irregulares en los cuales se realizan dichas operaciones pueden ser los identificados dentro de un único volumen de entrada o diferentes volúmenes de entrada. En otras palabras, las operaciones se pueden realizar a través de los volúmenes irregulares identificados en varios conjuntos de datos de volumen.

40 En otro aspecto de la presente invención, es posible procesar previamente los datos para convertirlos en nuevas primitivas de vóxel. Cada vóxel de un volumen irregular está representado por un poliedro en el que los estados de las caras (por ejemplo, "on" u "off") se codifican mediante una matriz de bits. También se puede codificar otro tipo de información, tal como información de normal, posición y parámetros de renderizado similares conocidos en el ámbito de la técnica. Las caras se pueden convertir fácilmente en primitivas aún más simples, tales como triángulos o cuadrángulos, en el supuesto que se desee renderizar los datos utilizando una tarjeta aceleradora de gráficos que no dispone de primitiva de vóxel.

45 Aunque las formas de realización ilustradas de la presente invención se refieren a datos sísmicos de volumen, la presente invención es aplicable a cualquier tipo de datos de volumen adecuados, tales como los que se utilizan en la formación de imágenes médicas y otras disciplinas. Debe tenerse en cuenta que tanto la descripción general anterior como la descripción detallada siguiente son ejemplificativas e ilustrativas únicamente y no limitan la presente invención definida en las reivindicaciones.

50 **Breve descripción de los dibujos**

Los dibujos adjuntos ilustran una o más formas de realización de la presente invención y, junto con la descripción, son útiles para explicar los principios de la presente invención. Siempre que sea posible, se utilizarán los mismos números de referencia en todos los dibujos para indicar elementos idénticos o similares de una forma de realización, y en los que:

la figura 1 ilustra un sistema informático de un ejemplo de forma de realización según la presente invención;

60 la figura 2 es un diagrama de flujo que ilustra un procedimiento para identificar y almacenar volúmenes irregulares y someterlos a renderizado y otro tipo de operaciones;

la figura 3A es un diagrama de flujo de la etapa de segmentación o preprocesamiento de la figura 2;

65 la figura 3B es una continuación de la figura 3A;

la figura 3C es una continuación de las figuras 3A a 3B;

la figura 3D es una continuación de las figuras 3A a 3C;

5 la figura 3E es una continuación de las figuras 3A a 3D;

la figura 3F es una continuación de las figuras 3A a 3E;

10 la figura 3G es una continuación de las figuras 3A a 3F;

la figura 3H es una continuación de las figuras 3A a 3G; y

la figura 3I es una continuación de las figuras 3A a 3H.

15 Descripción detallada

Se puede utilizar un ordenador, ilustrado de forma general en la figura 1, para aplicar los procedimientos de identificación, procesamiento y renderizado de volúmenes irregulares (VI) y otros procedimientos de la presente invención. Como se indica más adelante, gracias a los nuevos algoritmos y procedimientos de la presente invención, no es necesario que el ordenador sea una estación de trabajo gráfica potente como las que se utilizan normalmente para renderizar conjuntos de datos tridimensionales complejos, por ejemplo las empleadas comúnmente en los análisis geológicos del subsuelo, sino que en su lugar algunas formas de realización de la presente invención pueden estar constituidas por un ordenador personal corriente o incluso un ordenador portátil provisto de una memoria, una potencia de gráficos y una capacidad de procesamiento relativamente limitadas. Un ordenador adecuado presenta, por ejemplo, un procesador 10, una memoria principal 12 en la que se almacenan los programas y los datos durante su funcionamiento, un control de entrada/salida 14, un disco duro 16 o un dispositivo similar donde se almacenan los programas y los datos en memoria no-volátil, un teclado 18, un ratón o un dispositivo señalador similar 20 y un monitor de vídeo u otro dispositivo de presentación visual 22 en el que se pueden renderizar los VI 23. También puede añadirse un acelerador de gráficos 24 del tipo que presentan comúnmente los ordenadores personales para permitir el renderizado tridimensional. Como tendrán en cuenta los expertos en la materia a la que se refiere la presente invención, el ordenador comprende otros elementos de hardware y software que no se ilustran para mayor claridad, pero que comúnmente están presentes en dichos ordenadores. Aunque solo se ilustra un procesador 10 para mayor claridad, puede disponerse de varias unidades centrales de procesamiento (CPU) y el sistema puede ser de varios canales o agrupado, de tal forma que el procesamiento puede ser realizado por varias CPU u ordenadores, y el renderizado de primitivas de véxeles (descrito más adelante) puede ser realizado por varios sistemas o subsistemas de presentación de gráficos independientes.

Se representan conceptualmente los siguientes elementos de software como elementos residentes o almacenados en la memoria 12: procesador de segmentación 26; datos semilla 28; estructuras de datos 30; postprocesador 31 y dispositivo de renderizado 32. Como observarán los expertos en la materia, estos elementos de software se representan como elementos residentes en la memoria 12 con la finalidad de ilustrar conceptualmente el ordenador programado, y es posible que realmente no sean residentes de forma simultánea o en su totalidad, sino que en su lugar se creen, almacenen, eliminen o aparezcan en la memoria partes de programas y datos relativos a la presente invención en función de las necesidades, fundamentalmente bajo control del procesador 10 conforme a su programación. Dichos elementos de software o partes de estos pueden transferirse entre el disco duro 16 y la memoria 12 cuando sea necesario, de la manera convencional conocida por los expertos en la materia. La programación (código) puede almacenarse en el disco duro 16, una vez copiada desde un CD-ROM (no representado) u otro tipo de disco extraíble, un ordenador remoto a través de una conexión de red (no representada) u otra fuente de código de programa ejecutable por ordenador. El procesador de segmentación 26, el postprocesador 31 y el dispositivo de renderizado 32 son elementos fundamentales en la programación. Otros elementos más convencionales de la programación, tales como una interfaz de usuario adecuada, no se representan para mayor claridad, aunque serán fácilmente deducibles por los expertos en la materia con referencia a la información dada a conocer en presente memoria. La presente invención puede adoptar, pues, no solo la forma de realización de un procedimiento y un sistema, sino también de un soporte o un producto informático utilizable por un ordenador, en los cuales se almacenan elementos de software. Como se ha indicado anteriormente, en algún momento concreto habrá otros elementos de software residentes en la memoria 12 por lo menos en parte que no se representan para mayor claridad y que comprenden, por ejemplo, un sistema operativo adecuado tal como MICROSOFT WINDOWS, UNIX o LINUX.

Como se ilustra en la figura 2, las etapas principales del procedimiento de la presente invención comprenden una etapa de segmentación o preprocesamiento 34, una o más operaciones de volúmenes irregulares (VI) 36 y una etapa de renderizado 38. Antes de la etapa de preprocesamiento 34, se introduce un conjunto de datos de volumen en la etapa 40. El conjunto de datos de volumen de la forma de realización ilustrada de la presente invención puede ser del tipo comúnmente utilizado en los análisis geológicos del subsuelo. Dicho conjunto de datos comprende habitualmente un número elevado, a menudo de muchos millones o miles de millones, de puntos o celdas de datos en un espacio tridimensional, cada uno de los cuales representa el valor de un atributo tal como la amplitud de una

señal acústica reflejada por las características del subsuelo en un punto o una celda de un volumen tridimensional. Dicho de otro modo, el conjunto de datos representa un volumen del subsuelo. El volumen puede contener uno o más VI que representan yacimientos de hidrocarburos u otras características volumétricas que pueden diferenciarse del volumen en su conjunto basándose en los valores de los atributos. Como se describe en mayor detalle más adelante, la presente invención permite identificar cualquiera de dichos VI y almacenar sus representaciones en estructuras de datos, de una manera que facilita la realización eficaz y rápida de operaciones en los mismos y su renderizado.

La etapa 40 también puede comprender la definición de una “extensión”. Es decir, el usuario puede seleccionar solo una parte o un subconjunto de un conjunto de datos de volumen para la etapa de preprocesamiento 34. De esta manera, bajo control del usuario o un algoritmo automatizado, un conjunto de datos de volumen puede dividirse en partes o subconjuntos para que de ese modo se procesen solo los que se deseen o se puedan procesar las diversas partes por separado unas de otras. La división de conjuntos de datos de volumen de esta manera resulta especialmente propicia para los sistemas informáticos multitarea o multiprocesamiento, en los que varios subprocesos esencialmente paralelos entre sí pueden realizar el procesamiento de la etapa 34, y en los que un subproceso puede procesar una parte mientras otro subproceso está procesando otra parte.

La etapa de preprocesamiento 34 comprende una etapa 42 en la que se recorren las celdas del conjunto de datos de volumen de entrada hasta que se halla una celda semilla que cumple ciertos criterios predeterminados del VI que desea el usuario. A continuación, en la etapa 44, se determinan otras celdas que están relacionadas con la celda semilla de alguna manera predeterminada. Por ejemplo, las celdas pueden presentar una relación de adyacencia. Como se describe a continuación, la etapa 44 puede realizarse mediante un algoritmo recursivo. La información que describe cada celda determinada se almacena en una estructura de datos en la etapa 46. La información puede comprender, por ejemplo, el valor de un atributo tal como la amplitud que describe las propiedades de la celda, una indicación de la ubicación de la celda identificada en el conjunto de datos de volumen global u otro marco de referencia, y un identificador que establece de manera unívoca que la celda identificada pertenece al VI identificado. Por lo tanto, cada una de dichas celdas se marca para indicar el VI al cual pertenece. El término “pertenece” significa en la presente memoria, que la celda está relacionada con las demás celdas del VI de una manera predeterminada, por ejemplo a través de la adyacencia espacial (es decir, las celdas se hallan a una distancia predeterminada o seleccionada unas de otras, o un punto o unas celdas de referencia se hallan dentro de una vecindad predeterminada o seleccionada), las características de wavelet u otro tipo de atributo o limitación. Las etapas 42, 44 y 46 se realizan hasta que se identifican todas las celdas que pertenecen a cada VI del conjunto de datos de volumen de entrada. Debe observarse que es posible procesar de esta manera más de un conjunto de datos de volumen o una sección de este. Los VI resultantes que se identifican y almacenan pueden ser, pues, VI que se hallan dentro de volúmenes diferentes. Por ejemplo, puede ser deseable preprocesar una pluralidad de conjuntos de datos de volumen que representan instantáneas del mismo volumen de subsuelo en momentos diferentes, de una manera que permita comparar los VI que se hallan dentro de estos para ver cómo pueden haber cambiado a lo largo del tiempo características tales como los yacimientos de hidrocarburos.

Es en la etapa de operaciones con los VI 36 cuando se pueden realizar dichas comparaciones y otras operaciones. Por ejemplo, para comparar dos VI que representan instantáneas del mismo volumen del subsuelo en momentos diferentes a fin de determinar cómo ha cambiado dicho volumen, se puede realizar una operación booleana AND o de intersección. Otras operaciones pueden comprender una operación booleana OR o de unión, operaciones aritméticas, operaciones gradiente y cualquier otra operación conocida dentro del ámbito de la técnica aplicable a datos de volúmenes geológicos o datos de volúmenes similares. Estas operaciones pueden realizarse de manera muy eficaz, ya que las estructuras de datos contienen toda la información necesaria, tal como la ubicación de cada celda y sus valores de atributos, no siendo por lo tanto necesario acceder de nuevo al conjunto de datos original.

Aunque los expertos en la materia a la cual se refiere la presente invención conocen sobradamente las operaciones gradiente, por ejemplo, debe tenerse en cuenta que las etapas pueden comprender (en caso de que la primitiva de vóxel sea, como se describe más adelante, un hexaedro o un cubo), la determinación de las normales de cada cara del hexaedro y el cálculo del gradiente (f_{normal}) mediante el conocido procedimiento de diferencias centrales.

Los VI almacenados o cualquier VI resultante de las operaciones de la etapa 36 pueden visualizarse en la etapa de renderizado 38. Puede utilizarse cualquier procedimiento de renderizado tridimensional adecuado conocido dentro del ámbito de la técnica. No obstante, la etapa 38 puede comprender de manera alternativa o adicional el nuevo procedimiento de renderizado de la presente invención, que convierte o transforma los datos almacenados en primitivas gráficas (poliédricas) de vóxel y, en algunas formas de realización de la presente invención, en primitivas gráficas más simples (por ejemplo, polígono, punto, línea). Por ejemplo, en la etapa 48 el usuario puede seleccionar el VI que desea renderizar. En la etapa 50, el VI puede transformarse en una representación poliédrica basada en vóxel. Por ejemplo, cada celda puede representarse como un vóxel cúbico o como unos conos (para facilitar una indicación vectorial). En la presente memoria de patente, el término “vóxel” se utiliza para hacer referencia a una representación de software de un objeto volumétrico tridimensional que comprende suficiente información para renderizar dicho objeto y aplicarle otro tipo de procesamiento sin recurrir a los algoritmos de renderizado tridimensionales aproximados convencionales, tales como los sistemas de raycasting o texturizado. El término “celda” se refiere al correspondiente objeto volumétrico no procesado que no comprende dicha información de

renderizado, aunque puede comprender información de atributos y otro tipo de información. En la forma de realización ilustrada de la presente invención, se define una cadena de bits de ejemplo que comprende seis bits que indican el estado de una cara del cubo cada uno. Si el valor del bit es "1", la cara se visualizará, es decir, está activada ("on"). Si el valor del bit es "0", la cara no se visualizará, es decir, está desactivada ("off"). Otro bit de la cadena puede indicar si el vóxel en su conjunto está activado o desactivado ("on" u "off"), puesto que a veces es deseable visualizar todas las caras. Otro bit de la cadena puede indicar si el vóxel está seleccionado. La selección se refiere al tipo de operación que el usuario puede realizar para seleccionar una parte que quiere someter a una operación. La parte seleccionada de un VI puede aparecer, por ejemplo, en un color diferente al de las partes no seleccionadas. Cuando se transforma una celda en una primitiva poliédrica de vóxel, puede generarse otra cadena de bits (por ejemplo, de 48 bits) que indica la ubicación del vóxel, así como otra cadena (por ejemplo, de 12 bytes) que aporta la información normal. Como bien se sabe dentro del ámbito de la técnica, los motores gráficos 3D utilizan la información de normal para determinar el sombreado adecuado basándose en la luz reflejada desde el objeto con respecto al punto de visión y una o más fuentes de luz.

En las formas de realización de la presente invención, en las que el acelerador de gráficos 24 acepta primitivas de vóxel como entrada, los datos de vóxeles resultantes pueden facilitarse directamente al acelerador de gráficos 24 para el renderizado en la etapa 54. En las formas de realización en las que el acelerador de gráficos 24 acepta solo las primitivas más convencionales, tales como puntos, líneas y polígonos, cada vóxel puede transformarse en dichas primitivas en la etapa 52. Por ejemplo, cada cara de un cubo puede dividirse en dos triángulos. Los triángulos constituyen una primitiva generalmente aceptada en muchos aceleradores de gráficos 24 disponibles en el mercado.

También puede realizarse un vaciado en la etapa 50. El término "vaciado" se refiere a una operación de ajuste de escala que renderiza los datos utilizando una cantidad de vóxeles que es inferior cuando el punto de visión del usuario está más alejado de la imagen y superior cuando el punto de visión del usuario está más cerca. Dicho de otro modo, si la vista es más lejana, un grupo de vóxeles puede renderizarse como un vóxel único. Por lo tanto, la etapa 50 es sensible a las entradas de interfaz de usuario que indican la distancia desde el punto de visión hasta la imagen. Cada celda puede vaciarse asincrónicamente en el momento de realizar el procesamiento/renderizado, ya que cada celda contiene información de posición y de estado. El vaciado puede tener lugar durante el renderizado o el procesamiento, dependiendo de los resultados deseados. Por ejemplo, tal vez sea deseable mezclar varios atributos en un VI, renderizándolos en ubicaciones de vaciado diferentes.

Las figuras 3A a 3I ilustran la etapa de preprocesamiento 34 en mayor detalle. En la etapa 56, el usuario identifica o selecciona el conjunto de datos de volumen con el cual va a trabajar. El usuario también puede seleccionar una extensión o una parte del conjunto de datos con la cual va a trabajar. Por lo tanto, el usuario puede dividir selectivamente el conjunto de datos en varias partes y procesar solo algunas de ellas o hacer que el ordenador las procese por separado, por ejemplo, mediante multiprocesamiento. Aparte de dividirse por acción del usuario, el conjunto de datos puede dividirse automáticamente en un número predeterminado de partes.

En la etapa 58, si el usuario ha indicado que deseaba realizar una operación de unión entre dos o más VI, se realiza la etapa 60. La etapa 60 se ilustra en mayor detalle en la figura 3G. Si, como se ilustra en la figura 3G, se determina en la etapa 62 que un volumen booleano ya no existe, entonces se asigna uno a la memoria en la etapa 64. En caso de que exista uno de dichos volúmenes, este se restablece en la etapa 66. En la etapa 68, se inicializan y activan celdas de gestión. Esto significa que se marcan una serie de campos de una estructura de datos para indicar que se va a realizar una operación booleana. El procesamiento vuelve a la etapa 70 de la figura 3A.

Se utiliza una matriz recursiva de estructuras para reducir la cantidad de memoria de pila que en su ausencia necesitaría el sistema de hardware. Es decir, cada vez que se activa una función recursiva, se crea, en la memoria dinámica, un marco que contiene los argumentos de la función. La matriz recursiva de estructuras permite desplazar con eficacia la carga de memoria a la memoria dinámica, donde la memoria se asigna y desasigna dinámicamente y, por lo tanto, no es un recurso tan limitado como la pila. Si en la etapa 70 se determina que dicha matriz todavía no existe, se crea una en la etapa 72.

Un volumen de bits es un objeto en el que sus elementos binarios presentan una correspondencia biunívoca con cada muestra o elemento de la matriz de volumen de datos original. Es decir, se utiliza un elemento de 1 bit por cada elemento de muestra de 8, 16, 32, 64, etc. bits. El objeto de matriz de bits puede utilizarse para no tener que realizar búsquedas o selecciones, y aporta una estructura para mantener el estado de travesía. Se utiliza un volumen de bit de recuento, en conjunción con el volumen de datos original para eliminar todavía más ruido, manteniendo un registro del cálculo del tamaño irregular inicial. Esto brinda también una manera de restringir el tamaño de los VI y las celdas que se van a aceptar. Dicho volumen puede utilizarse como una máscara para áreas no procesadas, también. El volumen de bit de recuento es operativo para proveer un sistema de comprobación de estado activado/desactivado y para eliminar la necesidad de disponer de sistemas de búsqueda que resultan prohibitivos cuando se utilizan grandes volúmenes de datos. Es decir, debido a que el acceso es aleatorio, el tiempo necesario para acceder a una celda no depende de la cantidad ni el tamaño de los datos. Si en la etapa 74 se determina que existe un volumen de bit de recuento, este se restablece en la etapa 76. Si no existe ninguno, se crea uno en la etapa 78.

Se utiliza un volumen de bit de estado, en conjunción con el volumen de datos original, para mantener el estado de travesía de procesamiento (para estar informado acerca de las celdas que ya se han procesado a fin de no volver a procesarlas). El volumen de bit de estado también facilita la realización de operaciones booleanas entre los VI y el procesamiento conjunto de grupos de VI procedentes de conjuntos de datos de volumen diferentes. Dicho volumen también se utiliza para combinar diversos conjuntos de datos de volumen irregular, que se crean subdividiendo el conjunto de datos original en conjuntos de datos más pequeños que se procesan en subprocesos separados o de forma independiente. Si en la etapa 80 se determina que existe un volumen de bit de estado, este se restablece en la etapa 82. Si no existe ninguno, se crea uno en la etapa 84.

En la etapa 85, se inicia un bucle en el que se procesan las muestras de celdas del conjunto de datos de volumen de entrada. En la etapa 86, se provee información de estado, que indica por ejemplo si se han sobrepasado los requisitos de memoria, por medio de la interfaz de usuario. En la etapa 88, se determina si la muestra actual en ese momento cumple los criterios predeterminados para las celdas semilla. Se considera que la muestra es una celda de semilla si: (1) el atributo cumple unos requisitos mínimos predeterminados; (2) el bit de estado no está activado en el volumen de bit estado; (3) el bit de recuento no está activado en el volumen de bit de recuento; (4) la celda situada a la izquierda de la celda de muestra actual en ese momento no es una celda válida; (5) la celda situada encima de la celda de muestra actual en ese momento no es una celda válida y (6) la celda situada delante de la celda de muestra actual en ese momento no es una celda válida. La condición (1) conlleva la comparación del atributo con uno o más umbrales predeterminados. Un umbral puede ser, por ejemplo, el valor de un único atributo, tal como la amplitud. Por ejemplo, tal vez sea deseable ignorar las muestras cuya amplitud no se halla por encima de cierto umbral. La condición (2) indica que el algoritmo todavía no ha procesado la celda y que por lo tanto esta sigue estando pendiente de identificación como celda del VI actual o celda semilla de un VI diferente. Si el bit de recuento al cual se refiere la condición (3) está activado, significa que otro proceso de búsqueda ya ha considerado esta muestra como posible celda semilla, y o bien que la celda ya se ha procesado o bien que el VI al cual pertenece la celda no satisface un umbral de tamaño mínimo predeterminado. (Un VI muy pequeño puede representar ruido, en lugar de la característica de subsuelo que interesa al usuario.) Las condiciones (4), (5) y (6) se refieren al orden en que se procesan las celdas del conjunto de datos de volumen de entrada: de izquierda a derecha, a continuación de arriba abajo y finalmente de atrás adelante. Si ya se ha procesado y verificado la validez de la celda superior, izquierda o delantera, la celda de muestra actual en ese momento ya habrá sido procesada y puede rechazarse como posible celda semilla. Debe tenerse en cuenta que las condiciones descritas anteriormente que definen los criterios de las celdas semilla solo son ejemplificativas, y que es posible utilizar más o menos criterios.

Debe tenerse en cuenta asimismo que, aunque el recorrido a través de las celdas tiene lugar de una celda a otra adyacente hasta que se encuentra una semilla, el procedimiento no se limita a este tipo de recorrido. Aparte de recorrer las celdas de una a otra basándose en la contigüidad espacial, el procedimiento puede pasar de una celda a otra basándose en un desplazamiento o una búsqueda de proyección de vecindad o cualquier otro medio conveniente para recorrer las celdas de una a otra (no siendo necesario que estas sean espacialmente adyacentes).

Si se encuentra una celda semilla, entonces se inicia un proceso recursivo en la etapa 90, en el que se asigna un identificador principal al volumen irregular al cual pertenece la celda semilla, y las celdas de ese VI se identifican y se añaden a un campo de recuento de celdas del VI. La información que describe cada celda identificada, tal como su ubicación y el identificador que indica el volumen irregular al cual pertenece, se almacena en una estructura de datos correspondiente a esa celda. El procesamiento recursivo de VI a partir de la celda semilla se describe en mayor detalle a continuación. El proceso de búsqueda de celdas semilla y de procesamiento del VI al cual pertenece cada celda identificada se repite hasta que se determina, en la etapa 92, que todas las celdas del conjunto de datos de volumen de entrada se han sometido al muestreo.

Pueden realizarse fácilmente varias operaciones de postprocesamiento una vez que se han identificado y almacenado los VI del volumen de entrada. Estas operaciones son en esencia las que, en el ámbito de la técnica anterior, se consideran de interés para los geólogos y otro tipo de usuarios de dicho sistema. Por ejemplo, las operaciones de unión y de proyección son muy conocidas. No obstante, el preprocesamiento permite realizar dichas operaciones con mucha más eficacia y rapidez que cuando se utilizan algoritmos convencionales para realizarlas, ya que no hay necesidad de reprocesar volúmenes de datos completos y crear volúmenes resultantes del mismo tamaño. Por ejemplo, si en la etapa 94 se determina que el usuario ha indicado que va a realizar una operación de unión, en la etapa 96 se realiza el procesamiento de unión que se beneficia del preprocesamiento, tal como se describe a continuación. Si en la etapa 98 se determina que el usuario ha indicado que va a realizar una operación de proyección, en la etapa 100 se realiza una operación de proyección que se beneficia del preprocesamiento. En los sistemas convencionales, se utiliza el algoritmo de raycasting u otros algoritmos complejos para realizar operaciones de proyección y de visualización. En la presente invención, la proyección es una etapa sencilla que los expertos en la técnica a la que se refiere la presente invención podrán implementar con facilidad. Para descargar en el hardware y renderizar de manera continua un único conjunto de información geométrica, basta con cambiar los parámetros de visión. Como se indica en la etapa 102, es posible realizar cualquier otra de dichas operaciones conocidas antes de la descarga.

Por ejemplo, una de las operaciones que puede realizarse en la etapa 102 con un VI identificado es la visualización de la proyección de los atributos mínimo o máximo. El VI se procesa de tal forma que todas las celdas que tienen

una propiedad que las identifica como celdas superficiales, es decir, como celdas situadas en la superficie del vóxel externo del VI, reciben un nuevo valor de atributo. Las columnas de celdas del VI se procesan determinando el número de vóxeles presentes entre la celda superior, es decir, la celda superficial, y una celda inexistente en el volumen de bits. El volumen de bits se marca con todas las celdas existentes. Los valores de atributo de las celdas superiores se comparan a continuación con las siguientes celdas continuas situadas en la columna. El valor máximo o, en otros casos, el valor mínimo del atributo sustituye al valor de atributo de la celda superior. Se realiza la misma operación para las celdas inferiores. Esta proyección puede realizarse utilizando un vector normal o de visión/proyección que se proporciona.

En la etapa 104, pueden realizarse tareas de mantenimiento, tales como la supresión de los volúmenes de bit de estado y bit de recuento y los recursos de memoria dinámica/de pila, antes de que termine el procesamiento.

El procesamiento de unión de la etapa 96 se ilustra en las figuras 3H a 3I. En la etapa 106, se identifica la siguiente celda que se va a procesar en el VI seleccionado y se examina la correspondiente estructura de datos del volumen booleano. Por ejemplo, en la etapa 108 se determina si el bit "AND" de la estructura de datos está activado, hecho que indica que va a realizarse una operación booleana después del procesamiento, y por lo tanto todas las celdas deben marcarse adecuadamente para su correcta correlación con las celdas de procesamiento adecuadas. Si el bit está activado, entonces se activa el bit de intersección de la correspondiente ubicación en el volumen resultante, es decir, un volumen que representa los resultados de la intersección. Si en la etapa 112 se determina que se van a procesar más celdas del VI seleccionado, el procesamiento vuelve a la etapa 106. Cuando se han procesado todas las celdas, en la etapa 114 se determina si el bit m_ORA está activado y si existe la celda en la respectiva posición de volumen de bit booleano. Si dicho bit está activado, en la etapa 116 se ponen a cero ciertos campos de la celda que no son de interés, es decir, que no se van a procesar. Uno de dichos campos puede ser el campo "visibility/ACTIVE" que indica si la celda está activada, es decir, si está visible cuando se renderiza. Otro campo puede ser una marca que se coloca cuando el procesamiento indica que la celda debe suprimirse pero todavía está presente. ORA es un campo que indica que solo hay una celda en esa posición espacial en ese momento. Ningún otro VI contiene una celda en esa posición. Esta inicialización se repite para cada celda, como se indica en las etapas 118 y 120. Cuando se han procesado todas las celdas, en la etapa 122 se combinan los identificadores de VI si el conjunto de datos de entrada se ha procesado en partes (es decir, definiendo extensiones). Otra posibilidad es que la combinación tenga lugar más tarde o de alguna otra manera, por ejemplo en otro ordenador.

En la etapa 124, se determina si las celdas AND están activadas, es decir, si van a estar visibles o no en el VI resultante cuando se presente. Si estas celdas están activadas, la etapa 126 representa una rutina o procedimiento de activación "activateORAcellsOnly" adecuado. Esta rutina aísla y marca todas las celdas que se identifican mediante la operación de celda OR, que se pueden visualizar para el procesamiento. En la etapa 128, se determina si las celdas "ORA" están activadas, es decir, si van a estar visibles o no en el VI resultante cuando se presente. Si las celdas están activadas, la etapa 130 representa una rutina o procedimiento de activación "selectANDcellsOnly" adecuado. Esta rutina identifica celdas que satisfacen una condición booleana AND, que puede ser el caso de dos celdas presentes en dos VI diferentes de uno o más conjuntos de datos o el caso de un procesamiento posterior de celdas que ocupan la misma ubicación espacial. En la etapa 132, se determina si las celdas AND están seleccionadas, es decir, si van a visualizarse o no en un color distintivo cuando se presente el VI resultante. Si estas celdas están seleccionadas, la etapa 134 representa una rutina o un procedimiento adecuado de aislamiento de celdas presentes solo en el primer volumen. Por ejemplo, se aíslan celdas si va a realizarse una operación booleana AND entre dos VI coexistentes, pero es deseable renderizar solo las celdas comunes presentes únicamente en el primer VI. En la etapa 136 se determina si el primer VI (de los dos que van a someterse a la operación de unión) está activado, es decir, si el VI en su conjunto va a estar visible o no en el VI resultante cuando este se visualice. Si el VI está activado, la etapa 138 representa una rutina o un procedimiento adecuado para renderizar solo las celdas marcadas presentes en el segundo VI. En la etapa 140, se determina si el segundo VI (el otro de los dos que van a someterse a la operación de unión) está activado, es decir, si el VI en su conjunto va a estar visible o no en el VI resultante cuando este se visualice. Si está activado, la etapa 142 representa una rutina o un procedimiento adecuado semejante al representado por la etapa 130, aunque las celdas no se marcan de forma exclusiva con un color seleccionado durante el renderizado. Por último, en la etapa 144, se realizan algunas tareas de mantenimiento, tales como la supresión de información de la tabla, la reintroducción de información de la tabla y la supresión del volumen booleano temporal, antes de que el procesamiento vuelva a la etapa 96 y continúe por la etapa 98 descrita anteriormente.

En las figuras 3D a 3F, se ilustra la principal función recursiva activada en la etapa 90. Como se ha indicado anteriormente, esta función extrae o identifica el resto de celdas que pertenecen al VI al cual pertenece la celda semilla. Si en la etapa 146 se determina que el número de celdas contadas hasta ese momento en el VI actual, es decir el VI que se está sometiendo a procesamiento en ese momento, no se halla por debajo de cierto valor máximo predeterminado, la activación de la función retrocede en la etapa 148, puesto que esto significa que se ha producido un error o problema. Si en la etapa 150 se determina que la celda actual, es decir la celda que se está sometiendo a procesamiento en ese momento, no se halla dentro de la extensión o subvolumen seleccionado, la activación de la función retrocede en la etapa 152, puesto que esto significa que se ha producido un error o problema. Si en la etapa 154 se determina que la celda actual ya se ha procesado (mediante una activación recursiva anterior), entonces la activación de la función retrocede en la etapa 156. Como se ha mencionado anteriormente, el bit de estado indica si

se ha procesado la celda. Si ninguna de estas condiciones tiene como resultado un retroceso anterior de la activación de la función, el procesamiento continúa por la etapa 158, en la que se activa el bit de estado en la correspondiente posición de celda del volumen de bit de estado.

5 Si en la etapa 160 se determina que la celda actual es una celda semilla, debido a que esta es la primera celda del VI que se procesa, entonces el recuento total de volúmenes, que es el número total de celdas internas y externas, se pone a cero en la tabla del VI. La tabla del VI se utiliza para mantener información y estadísticas de objetos (pero no información que describe las celdas individuales del VI). Las coordenadas o la ubicación de la celda semilla se almacenan en la tabla. El atributo semilla se almacena en la estructura de datos de la celda, que contiene toda la información relacionada con cada celda, tal como información de posición, normal, caras, etcétera. El almacenamiento de información sobre la celda semilla, incluida su ubicación, es beneficioso porque permite extraer un VI sin necesidad de buscar la celda semilla.

15 En las etapas 164, 166, 168, 170, 172 y 174, se pregunta, respectivamente, si existe alguna celda a la derecha, izquierda, debajo, encima, detrás y delante de la celda actual. Análogamente, en la etapa 176 se pregunta si hay alguna de las 20 celdas vecinas restantes de una celda, por ejemplo, en la parte superior delantera derecha, en la parte superior delantera izquierda, etc. Debe tenerse en cuenta que una celda cúbica tiene un total de 26 celdas vecinas. Si cualquiera de dichas celdas vecinas está presente, el procesamiento avanza hasta la etapa 178, en la que se asigna el valor adecuado al índice de celda de borde, que es una marca que indica que la celda está situada en la superficie o cara externa del VI. A continuación, si en la etapa 180 se determina que la celda actual es una celda de borde, en la etapa 182 se establece la marca de borde en el valor True y se asigna el siguiente desplazamiento. Un desplazamiento es un valor que indica a qué distancia se halla la celda en una matriz lineal que representa la matriz de bits o la matriz de atributos. Es decir, todos los volúmenes de bits y matrices de datos presentan el mismo desplazamiento para una celda de una posición espacial específica. El almacenamiento de información que indica si una celda es una celda de borde permite renderizar la superficie con rapidez cuando el usuario desea ver solo la superficie del vóxel de un VI.

30 En la etapa 184, se determina si la celda vecina ya se ha procesado. La celda vecina es la que está situada a la derecha, izquierda, debajo, encima, detrás o delante de la celda actual, dependiendo de cuál de las etapas 154 a 174 genera un resultado verdadero. Si la celda vecina ya se ha procesado, el procesamiento continúa una vez que una de las etapas 154 a 174 ha generado un resultado verdadero. Si la celda vecina todavía no se ha procesado, entonces la función indicada en la etapa 90 e ilustrada en las figuras 3D a 3F se activa otra vez. La función, pues, se autoactiva de una manera recursiva hasta que todas las celdas que pertenecen al VI actual se han identificado y almacenado.

35 Cada vez que la función recursiva retrocede, el procesamiento continúa por la etapa 186, en la que se incrementa el recuento total de celdas del VI actual. A continuación, en la etapa 188 se determina si el usuario ha elegido procesar todo el VI o solo sus celdas de borde. Si el usuario elige procesar solo las celdas de borde, entonces en la etapa 190 se coloca una marca de límite y la función retrocede otra vez. Una marca de límite es un campo de datos que mantiene un recuento del número de celdas halladas. Cuando el recuento alcanza un valor máximo predeterminado, entonces se coloca la marca de límite, no se procesan más celdas y el procesamiento se detiene. Si se procesan todas las celdas, entonces en la etapa 192, la información que describe la celda se almacena en una estructura de datos de la celda. Existe una de dichas estructuras de datos para cada celda de cada VI. En particular, la información almacenada puede comprender, entre otras cosas, las coordenadas o la ubicación de la celda actual, un identificador del cuerpo de la celda que identifica de forma exclusiva la celda, un valor de atributo de la celda (por ejemplo, la amplitud) e información de vecindad. El valor del atributo es útil porque un VI puede utilizarse como plantilla para extraer con eficacia los valores de atributo de otros VI. La información de vecindad es una cadena de bits codificada que indica donde se hallan las celdas vecinas de la celda actual. Por ejemplo, un bit se pone a "1" si la celda actual tiene una vecina a su izquierda, otro bit se pone "1" si la celda actual tiene una vecina encima, etcétera. La información binaria de visibilidad de las caras del poliedro es coincidente y se deduce a partir de la información anterior. Por ejemplo, si se utiliza un hexaedro (cubo) y si no hay ninguna celda vecina encima de este, entonces significa que la cara superior del cubo de vóxel será visible, puesto que está orientada hacia el exterior del VI. Si hay un celda vecina enfrente de la parte inferior de la celda, entonces la visibilidad de la cara estará "desactivada" para la cara inferior del cubo, puesto que está orientada hacia el interior del VI.

55 Si en la etapa 194 se determina que se está realizando una operación booleana en el VI actual, entonces las celdas de uno o más VI se comparan en la etapa 196, y sus campos de estructura de datos se actualizan para un correcto renderizado. Por ejemplo, si se desean todas las celdas comunes, entonces el campo m_AND de todas las celdas AND se pondrán a "1". La activación de la función retrocede en la etapa 190.

60 Debe observarse que en el momento concreto de retroceso de la primera activación de la función recursiva, el procesamiento continúa desde la etapa 90. Como se ha descrito anteriormente, el procesamiento que sigue a la etapa 90 puede comprender no sólo la visualización de los VI que se han identificado y almacenado, sino también operaciones de unión, operaciones de proyección y cualquier otro tipo de operación con la cual estén familiarizados los expertos en la materia, ya que toda la información necesaria para realizar dichas operaciones de una manera sencilla se ha almacenado en las estructuras de datos. Aunque dichas operaciones convencionales pueden

realizarse mediante cualquier algoritmo o procedimiento adecuado, debe tenerse en cuenta que la operación de renderizado de una imagen puede realizarse mediante el nuevo procedimiento de renderizado descrito anteriormente con referencia a la etapa 38 (figura 2).

5 Una característica exclusiva del algoritmo descrito anteriormente es que solo es necesario calcular una vez toda la geometría de renderizado. Los algoritmos basados en vóxeles o pseudovóxeles conocidos dentro del ámbito de la técnica, tal como los sistemas de renderizado basados en raycasting, splatting y texturas, necesitan recalcularse su geometría cada vez que se renderiza un marco y/o cuando la dirección de visión del renderizado cambia. Además, la nueva primitiva de vóxel de algoritmo descrita anteriormente facilita de una manera económica la provisión de una primitiva de vóxel de hardware 3D verdadera, mediante el hardware 3D convencional disponible en el mercado.

10 También debe observarse que partiendo todo el volumen en componentes espacialmente atómicos (volúmenes irregulares), los datos pueden dividirse fácilmente en grupos para equilibrar la ejecución del sistema y la eficacia del renderizado en los sistemas de varios canales y varios subprocesos. En comparación, otros algoritmos deben ejecutar complicados sistemas de gestión de datos y realizar diversos tipos de cálculos cada vez que se renderiza un marco y/o cada vez que la dirección de visión de renderizado cambia.

15 Como resultará evidente para los expertos en la materia, es posible realizar diversas modificaciones y variantes de la presente invención sin apartarse del alcance de la misma. Otras formas de realización de la presente invención resultarán evidentes para los expertos en la materia tras la consulta de la presente memoria y la puesta en práctica de la presente invención dada a conocer en la misma. Debe observarse que la memoria y los ejemplos solo son ilustrativos, y que el alcance de la presente invención se indica en las reivindicaciones adjuntas.

20

REIVINDICACIONES

- 5 1. Procedimiento de codificación en vóxeles de un conjunto de datos de volumen que comprende una multiplicidad de celdas, que comprende las etapas siguientes:
- definir una pluralidad de vóxeles poliédricos, correspondiendo cada vóxel poliédrico a una celda;
- 10 almacenar una representación de cada vóxel poliédrico, incluyendo la representación una cadena de caras que presenta una pluralidad de bits, y una cadena de posición que presenta una pluralidad de bits, correspondiendo cada bit de la cadena de caras a una cara del vóxel poliédrico y presentando un valor que indica si la cara va a visualizarse o no; presentando la cadena de posición un valor que indica una posición espacial tridimensional del vóxel poliédrico; y
- 15 renderizar la representación de cada vóxel poliédrico, presentando la cadena de caras de por lo menos una representación de vóxel poliédrico un bit con un valor que indica que por lo menos se va a visualizar una cara para una celda interna.
- 20 2. Procedimiento según la reivindicación 1, en el que la etapa de renderizado de la representación de cada vóxel poliédrico comprende la transformación de cada cara o vóxel que se va a visualizar en una primitiva seleccionada de entre las tres siguientes: un punto, una línea y un polígono.
3. Procedimiento según la reivindicación 1, en el que la representación de cada vóxel poliédrico almacenada en la memoria comprende además un bit que indica si se ha seleccionado el vóxel.
- 25 4. Procedimiento según la reivindicación 1, en el que la representación de cada vóxel poliédrico almacenada en la memoria comprende además una cadena normal que indica el sombreado de cada cara que se va a visualizar.
- 30 5. Procedimiento según la reivindicación 1, en el que:
- cada vóxel poliédrico se representa en un cubo que presenta seis caras; y
- cada bit de la cadena de caras presenta un valor que indica si la cara se va a visualizar.
- 35 6. Procedimiento según la reivindicación 5, que comprende además:
- la etapa de transformación de cada cara que debe ser visualizada en una primitiva, dividiendo cada cara en dos triángulos.
- 40 7. Producto de programa informático de codificación en vóxeles de un conjunto de datos de volumen que comprende una multiplicidad de celdas, que comprende un soporte utilizable por un ordenador que contiene:
- unos medios para definir una pluralidad de vóxeles poliédricos, correspondiendo cada vóxel poliédrico a una celda;
- 45 unos medios para almacenar una representación de cada vóxel poliédrico, comprendiendo la representación una cadena de caras que presenta una pluralidad de bits y una cadena de posición que presenta una pluralidad de bits, correspondiendo cada bit de la cadena de caras a una cara del vóxel poliédrico y presentando un valor que indica si la cara se va a visualizar o no; presentando la cadena de posición un valor que indica una posición espacial tridimensional del vóxel poliédrico; y
- 50 unos medios para renderizar la representación de cada vóxel poliédrico, presentando la cadena de caras para por lo menos una representación de vóxel poliédrico un bit con un valor que indica que por lo menos se va a visualizar una cara para una celda interna.
- 55 8. Producto de programa informático según la reivindicación 7, en el que los medios para renderizar la representación de cada vóxel poliédrico comprenden unos medios para transformar cada cara que se va a visualizar en una primitiva seleccionada de entre las tres siguientes: un punto, una línea y un polígono.
- 60 9. Producto de programa informático según la reivindicación 7, en el que la representación de cada vóxel poliédrico almacenada en la memoria del ordenador incluye además un bit que indica si se ha seleccionado el vóxel.
10. Producto de programa informático según la reivindicación 7, en el que la representación de cada vóxel poliédrico almacenada en la memoria del ordenador comprende además una cadena normal que presenta un valor que indica el sombreado de cada cara que se va a visualizar.
- 65 11. Producto de programa informático según la reivindicación 7, en el que cada vóxel poliédrico se representa como

un cubo que presenta seis caras; y

cada bit de la cadena de caras presenta un valor que indica si la cara se va a visualizar.

- 5 12. Producto de programa informático según la reivindicación 11, que comprende además unos medios para transformar cada cara que se va a visualizar en una primitiva, dividiendo cada cara en dos triángulos.

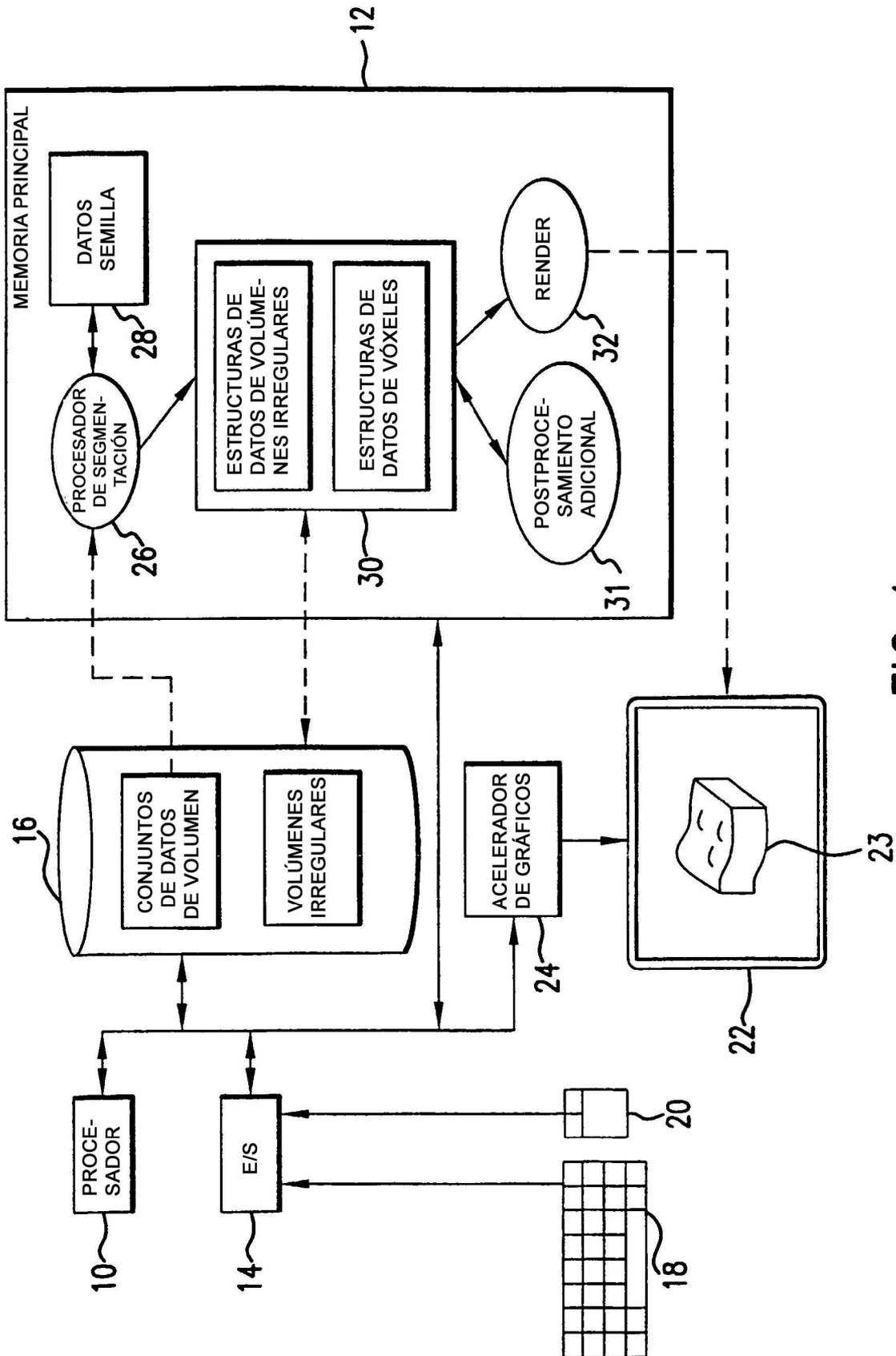


FIG. 1

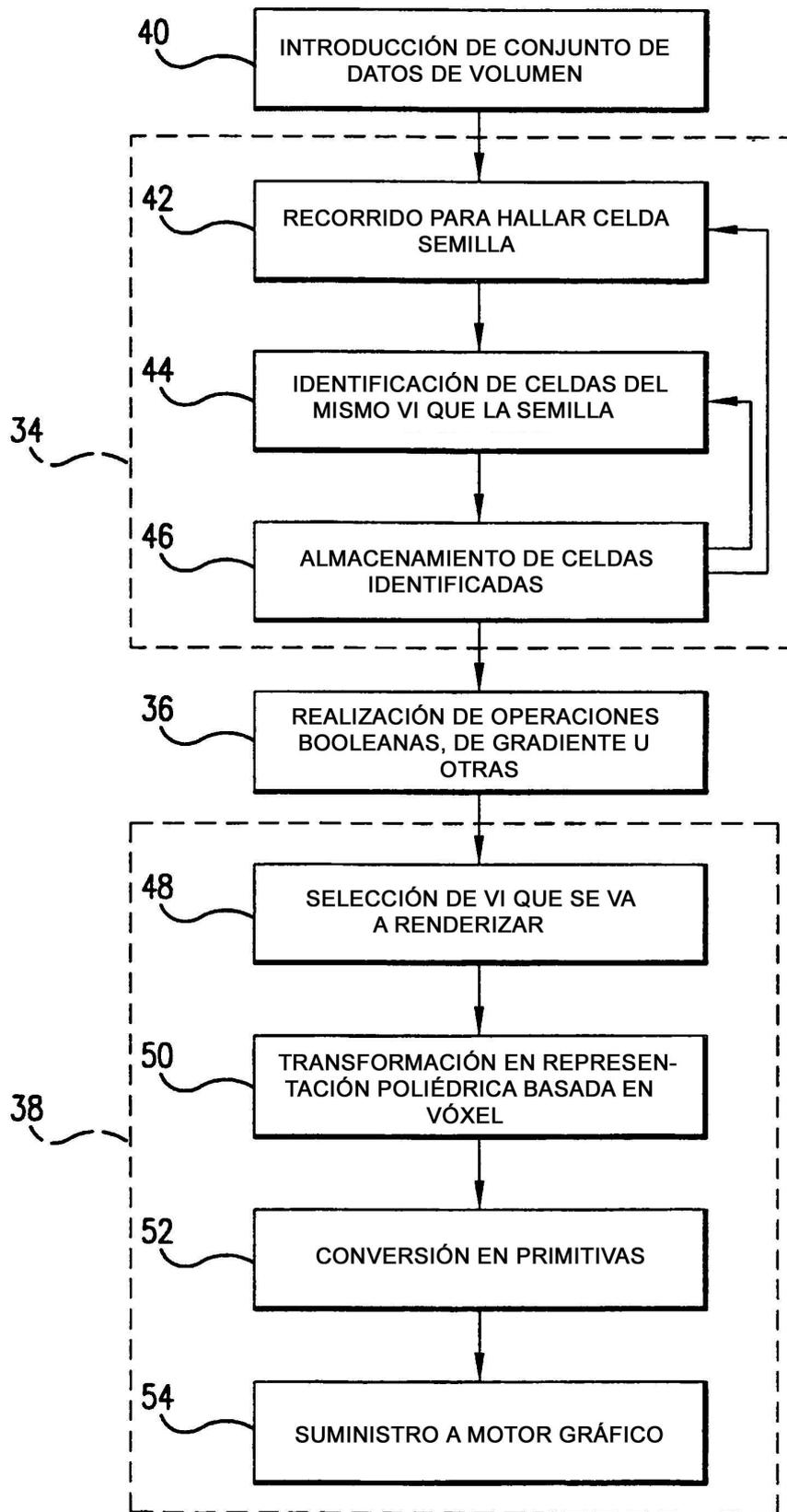


FIG. 2

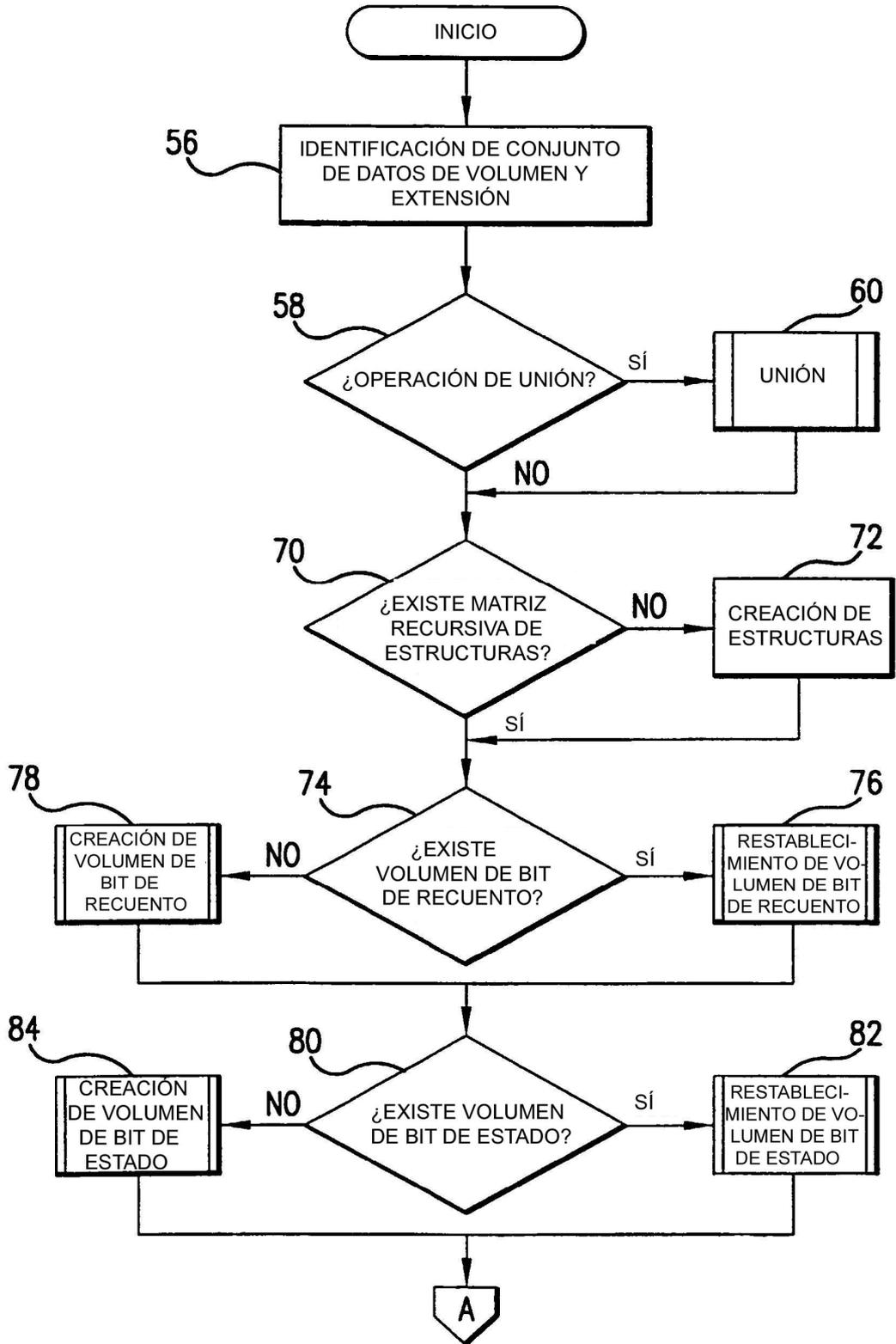


FIG. 3A

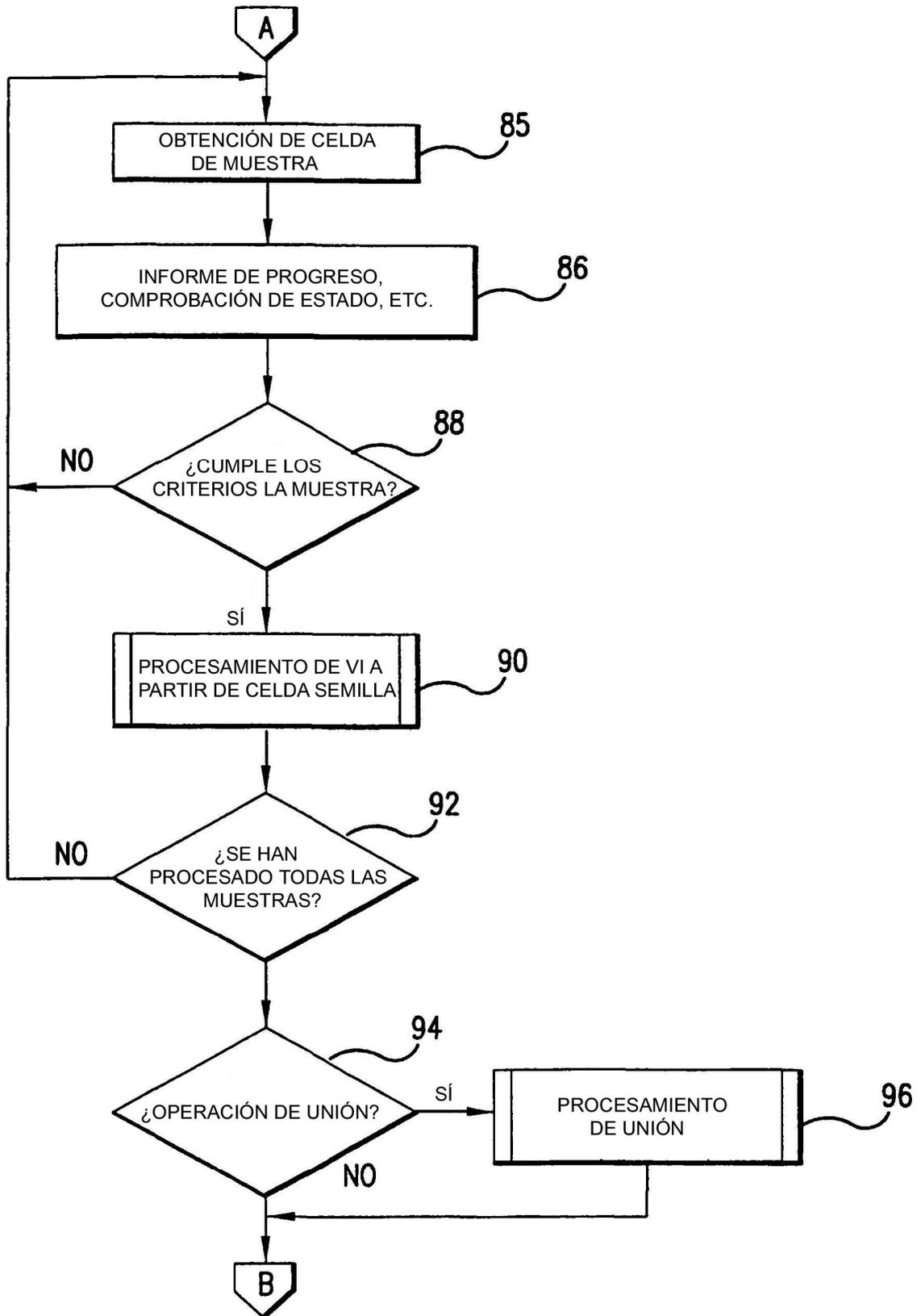


FIG. 3B

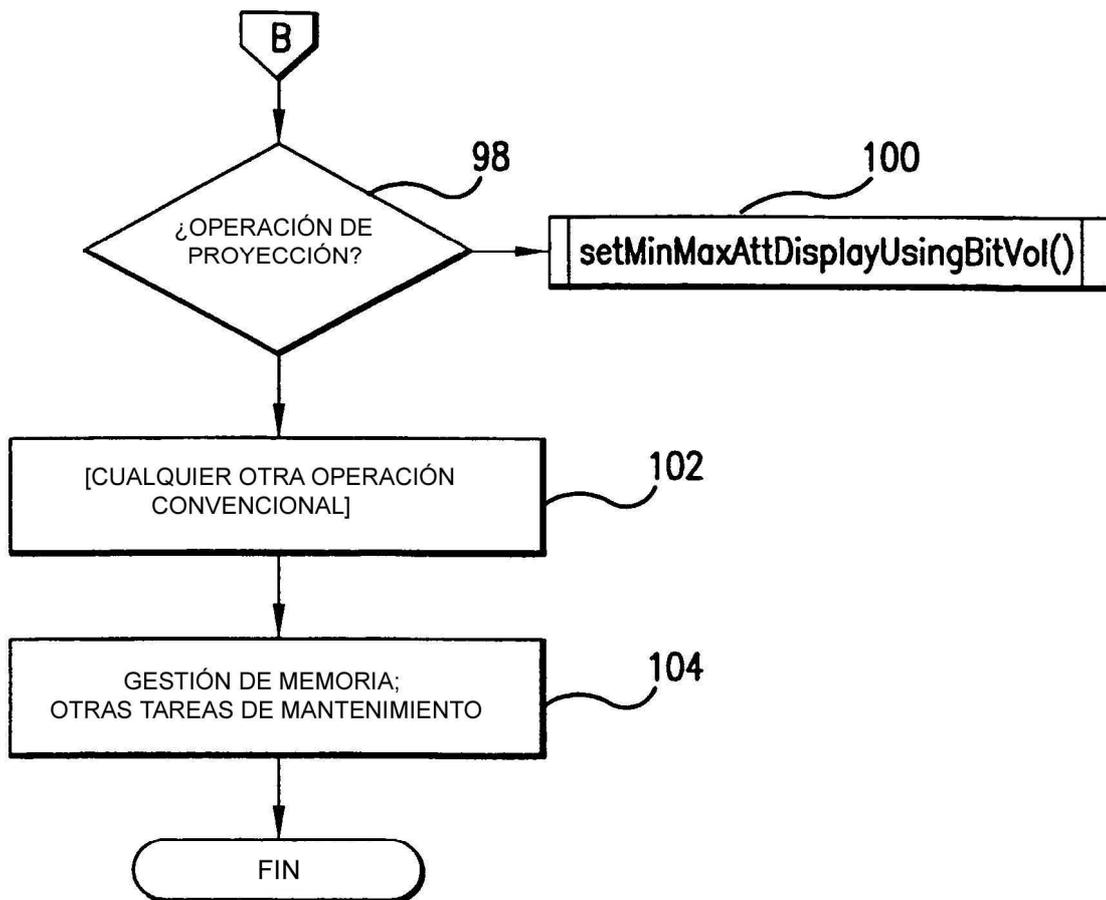


FIG. 3C

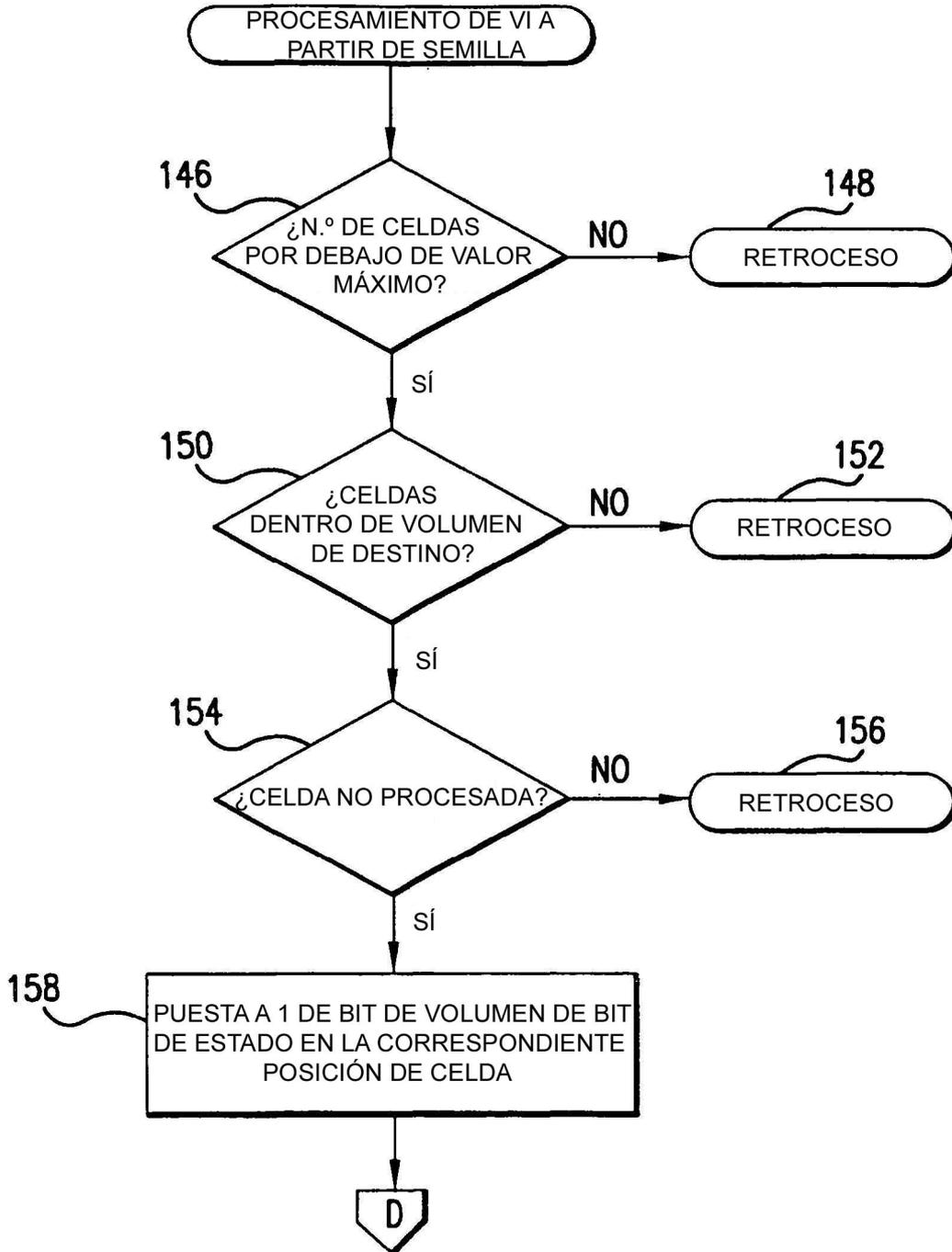


FIG. 3D

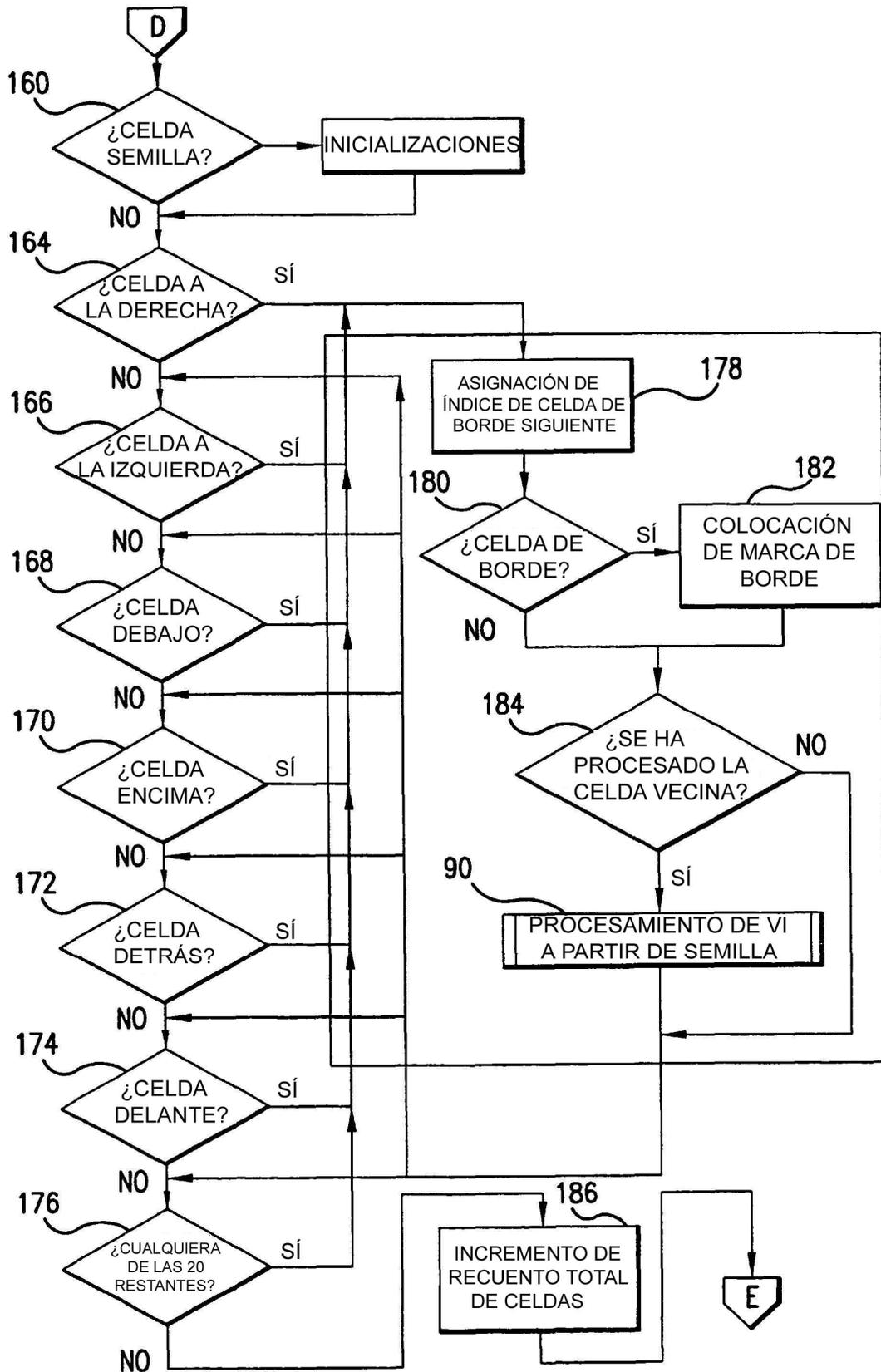


FIG. 3E

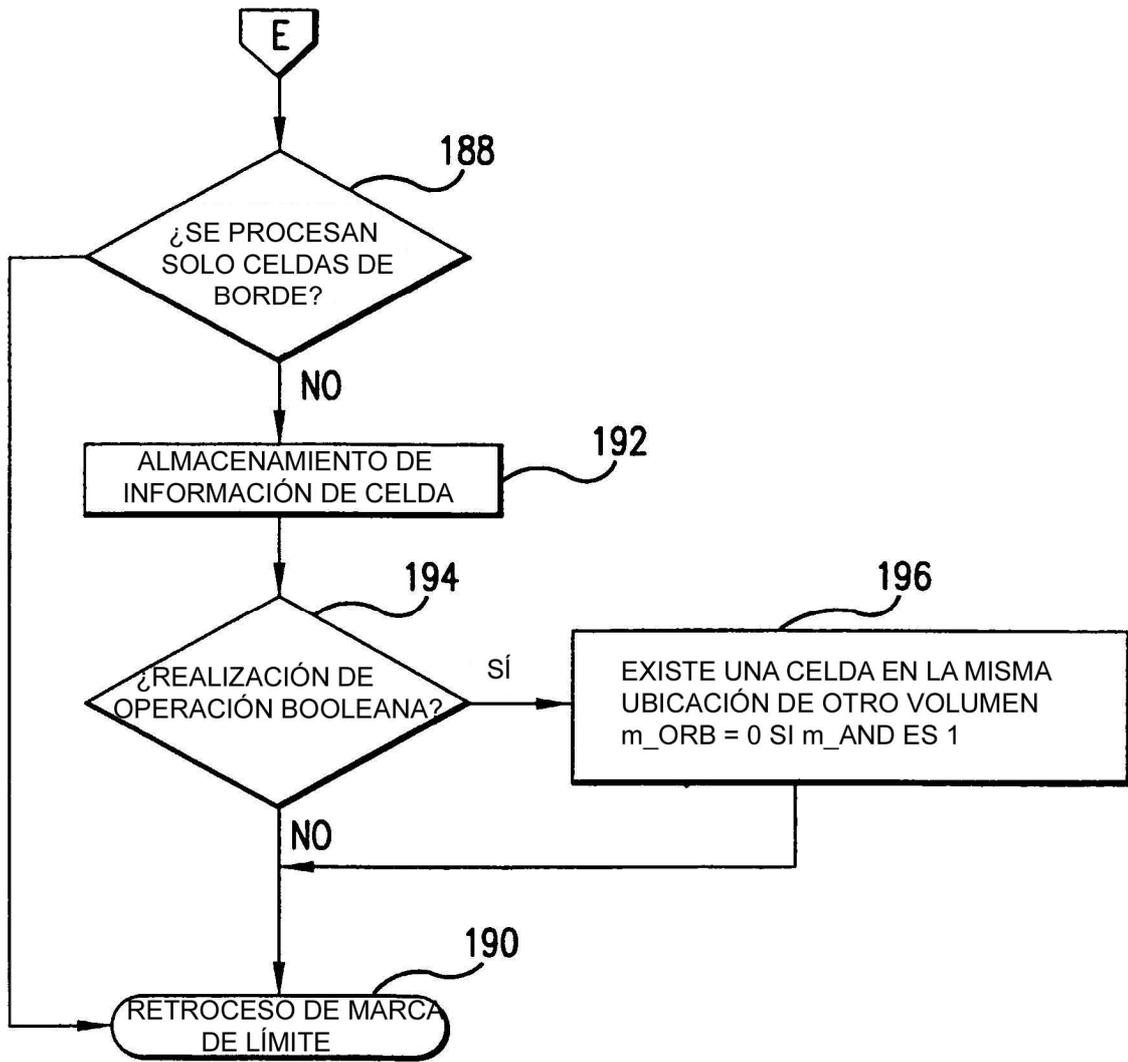


FIG. 3F

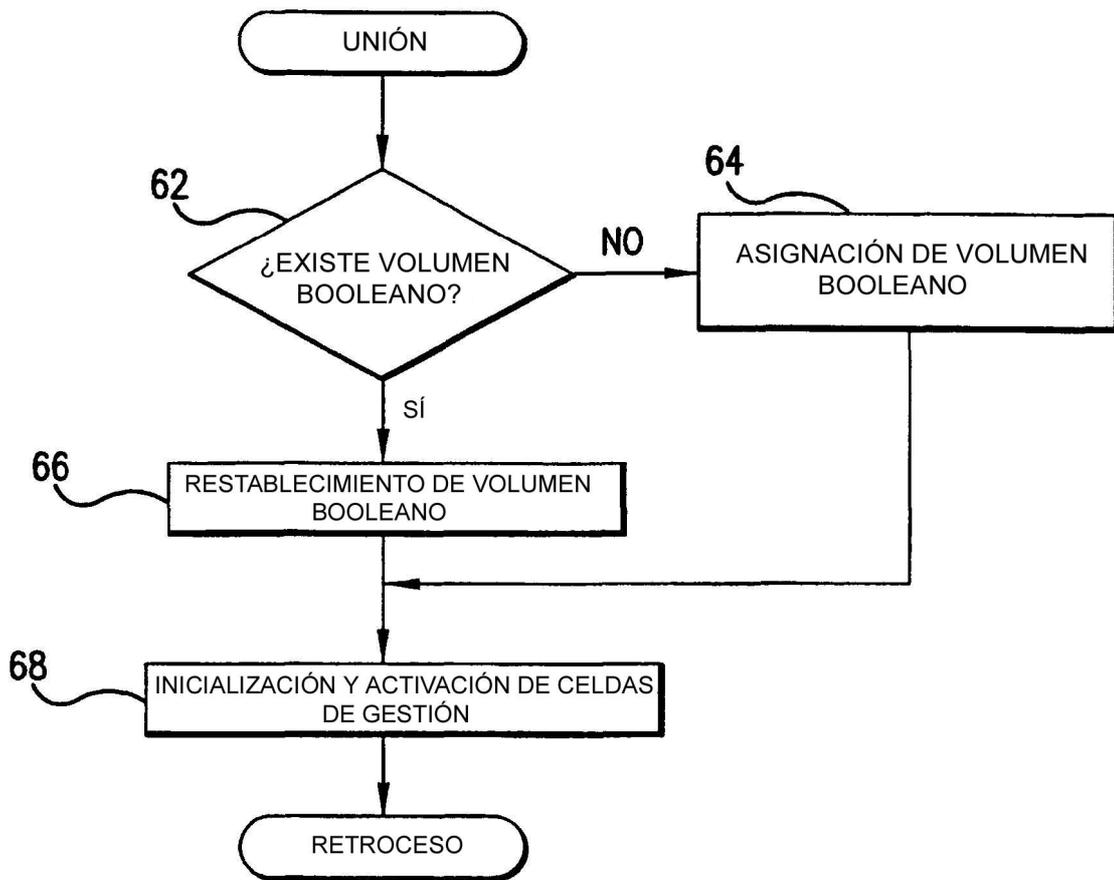


FIG. 3G

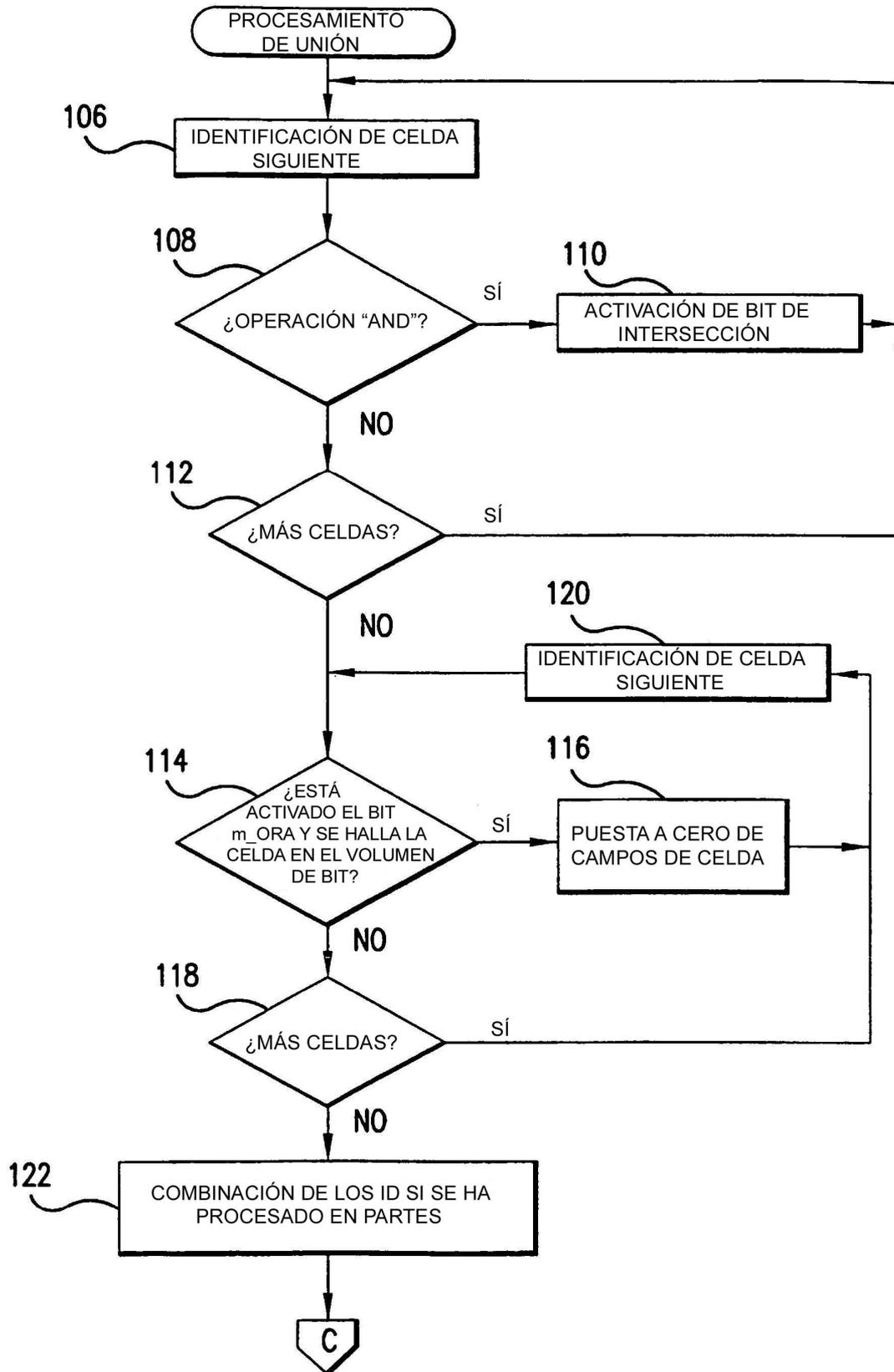


FIG. 3H

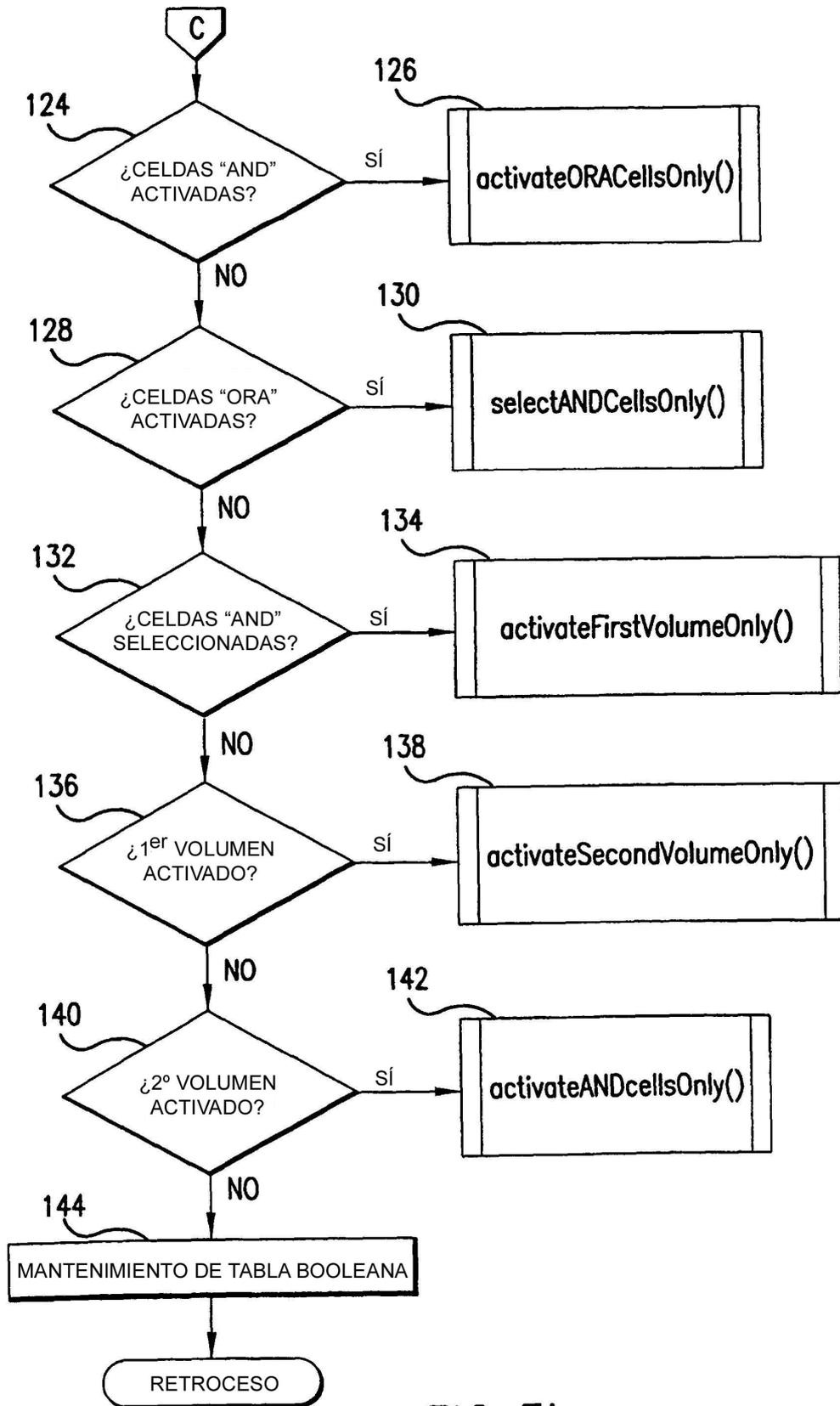


FIG. 31