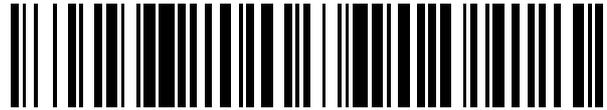


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 559 411**

51 Int. Cl.:

**G01N 15/14** (2006.01)  
**G01N 21/05** (2006.01)  
**G01N 21/47** (2006.01)  
**G02B 21/26** (2006.01)  
**G02B 21/36** (2006.01)  
**G06T 11/00** (2006.01)  
**G01N 15/10** (2006.01)  
**G01N 21/03** (2006.01)  
**G01N 21/17** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **06.06.2005 E 05757464 (2)**

97 Fecha y número de publicación de la concesión europea: **18.11.2015 EP 1769423**

54 Título: **Procedimiento de corrección de movimiento relativo de un detector de objetos entre vistas sucesivas**

30 Prioridad:

**24.06.2004 US 876328**

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

**12.02.2016**

73 Titular/es:

**VISIONGATE, INC. (100.0%)  
445 North 5th Street  
Phoenix, Arizona 85004, US**

72 Inventor/es:

**RAHN, JOHN RICHARD y  
NELSON, ALAN C.**

74 Agente/Representante:

**CARPINTERO LÓPEZ, Mario**

ES 2 559 411 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

Procedimiento de corrección de movimiento relativo de un detector de objetos entre vistas sucesivas

### Campo de la invención

5 La presente invención se refiere a la formación de imágenes y, más en particular, a la detección y corrección del movimiento relativo de un detector de objetos en un sistema de formación de imágenes donde, normalmente, se adquieren vistas sucesivas desde diferentes posiciones, representando cada vista una proyección o pseudoproyección en dos dimensiones del objeto en tres dimensiones.

### Antecedentes de la invención

10 Un sistema de microscopía tomográfica de proyección óptica (OPTM) es adecuado para la formación de imágenes en alta resolución de un objeto microscópico, tal como una célula biológica y su núcleo, que se incrustan en un medio fluido y se contienen dentro de un tubo microcapilar que tiene diámetros interiores y exteriores de 40 micrómetros y 150 micrómetros, respectivamente. Una OPTM emplea una pluralidad de vistas, adquirida cada una mediante la rotación del objeto y su recipiente de contención alrededor de un eje perpendicular al eje óptico y paralelo al eje del tubo microcapilar. Una cámara, que tiene un sensor de imágenes CCD compuesto de una matriz de píxeles MxN, captura la luz después de que pase a través del objeto y la óptica de formación de imágenes, lo que produce una imagen ampliada del campo de visión (FOV) en el CCD. Ya que cada vista se toma desde una perspectiva diferente, el contenido de cada vista se diferenciará de las otras.

15 Debido a los tamaños extremadamente pequeños de los componentes, puede ser bastante difícil colocar el eje de rotación (que coincide normalmente con el eje central del tubo microcapilar) en el centro del FOV del detector. Además, es muy difícil mantener el tubo microcapilar estacionario mientras rota. Adicionalmente, la propia célula puede moverse a lo largo del eje del tubo entre diferentes vistas. Como resultado, cada vista, que ya está alterada debido a la rotación del tubo, puede además someterse a traslaciones tanto axiales (paralelas al eje microcapilar) como laterales (perpendiculares al eje óptico y al eje del tubo). Estas traslaciones laterales se suman a las que ya están presentes para los objetos que no están en el eje de rotación.

20 Para obtener una reconstrucción precisa en 3D, ya sea a través de proyección por transparencia filtrada u otro medio, es por tanto necesario corregir el movimiento axial y esa porción del movimiento lateral que no se debe a la perspectiva cambiante desde una vista a otra. Es además necesario determinar dónde se ubica el eje de rotación en el FOV del detector.

25 La patente de Estados Unidos 4.858.128, de Nowak describe un procedimiento donde unas escenas consecutivas se relacionan unas con otras, primero en un eje y después, independientemente, en el otro eje. La ubicación del valor máximo para las dos correlaciones determina la desviación requerida para los dos ejes. El procedimiento describe fallos al proporcionar medios para distinguir la traslación lateral "natural", debido al cambio en perspectiva, de la traslación lateral "errónea", debido a la traslación del tubo microcapilar. La patente de Nowak enseña que, "puede ser útil estimar tal componente de fondo de la señal y sustraer la estimación de los datos de imagen".

30 William H. Press y col., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press; 2ª edición (1 de enero de 1993) describe medios para implementar, por medio de un programa informático, las técnicas de correlación mutua entre dos matrices de datos usando transformadas rápidas de Fourier (FFT). En resumen, la correlación mutua de dos matrices de datos (tales como datos de imagen) puede obtenerse aplicando una FFT a cada matriz, multiplicando una de las matrices resultantes por el conjugado complejo de la otra, y aplicando una FFT inversa al resultado.

35 Para superar los inconvenientes actuales en el estado de la técnica, es un objetivo de la presente invención proporcionar un procedimiento para encontrar la ubicación del eje central del tubo microcapilar para cada vista en un sistema de formación de imágenes de múltiples vistas. Es además un objetivo de la invención proporcionar un procedimiento para detectar el movimiento relativo del detector de objetos entre vistas sucesivas en un sistema de formación de imágenes de múltiples vistas. Es además un objetivo de la invención proporcionar un procedimiento para corregir datos de imágenes para retirar errores debido a un movimiento del objeto durante la recogida de datos de imagen. Es otro objetivo adicional de la invención proporcionar un sistema de formación de imágenes de un tipo que produce una polaridad de matrices de datos X-Y que representan listas de proyección o pseudoproyección de un objeto para una reconstrucción tomográfica posterior de cortes axiales del objeto. El movimiento detectado puede retirarse desplazando adecuadamente los datos posteriores para que se alineen con los datos anteriores, o viceversa.

### Sumario de la invención

40 La presente invención proporciona un aparato y un procedimiento para la corrección de movimiento para la formación de imágenes tomográficas y ópticas en tres dimensiones. Un objeto de interés se ilumina para producir una imagen. Un valor de corrección de desviación lateral se determina para la imagen. Un valor de corrección de desviación axial se determina para la imagen. El valor de corrección de desviación lateral y el valor de corrección de

desviación axial se aplican a la imagen para producir una imagen de archivo corregida.

### **Breve descripción de los dibujos**

Aunque las características nuevas de la invención se exponen con particularidad en las reivindicaciones adjuntas, la invención, tanto en organización como en contenido, se entenderá y apreciará mejor, junto con otros objetos y características de la misma, a partir de la siguiente descripción detallada tomada junto con los dibujos descritos a continuación.

La Figura 1 es un diagrama de bloques funcional de una realización de ejemplo de un procedimiento para la corrección del movimiento relativo de un detector de objetos entre vistas sucesivas construido de acuerdo con las enseñanzas de la presente invención.

La Figura 2 es un diagrama de bloques funcional de una porción de corrección lateral de un sistema de formación de imágenes que emplea la realización de ejemplo descrita en la Figura 1.

La Figura 3 es un diagrama de bloques funcional de una porción de corrección axial de un sistema de formación de imágenes que emplea la realización de ejemplo descrita en la Figura 1.

La Figura 4A representa una imagen de una célula antes de las operaciones de formación umbral que se emplean en un ejemplo del procedimiento de la presente invención.

La Figura 4B representa el resultado de aplicar operaciones de formación umbral que se emplean en un ejemplo del procedimiento de la presente invención en la imagen mostrada en la Figura 4A.

La Figura 4C ilustra un histograma que muestra las distribuciones de brillo de las imágenes mostradas en las Figuras 4A-4B.

La Figura 5 representa esquemáticamente un sistema de microscopía tomográfica de proyección óptica (OPTM) empleado en una realización de la invención.

La Figura 6A y la Figura 6B muestran esquemáticamente una realización de un sistema de tomografía óptica que incorpora un objetivo de microscopio montado en un dispositivo de traslación piezoeléctrica.

La Figura 7 muestra un diagrama de flujo de ejemplo que ilustra un procedimiento para adquirir imágenes usadas en la reconstrucción de imágenes en tres dimensiones (3D) tal como se contempla mediante una realización de la presente invención.

La Figura 8 muestra esquemáticamente un ejemplo de desviaciones de corrección de movimiento para su uso en una reconstrucción de imagen en tres dimensiones (3D) tal como se contempla mediante una realización de la presente invención.

### **Descripción detallada de las realizaciones preferentes**

La invención se describe en el presente documento con respecto a ejemplos específicos en relación con células biológicas, sin embargo, se entenderá que estos ejemplos tienen el fin de ilustrar los principios de la invención, y que la invención no queda limitada de esta manera. Aunque la presente invención puede emplearse en otros tipos de sistemas de formación de imágenes, tales como, por ejemplo, formación de imágenes de tomografía computada (CT) de rayos X, para concretar la descripción, la siguiente divulgación va dirigida a la invención en el entorno de un sistema de microscopía tomográfica de proyección óptica (OPTM).

En el análisis a continuación, se usan las siguientes suposiciones al proporcionar ejemplos numéricos:

1. Cada imagen consiste en una matriz, de 640 píxeles de ancho por 480 píxeles de alto;
2. Cada píxel contiene un único valor de brillo de 8 bits (nivel de grises de 0 a 255);
3. En referencia a una OPTM que usa un tubo microcapilar, el eje del tubo es paralelo al eje más corto (480 píxeles);
4. En referencia a una OPTM que usa un tubo microcapilar, la separación de la pared del tubo es 530 píxeles;
5. El número de barras usadas para encontrar la desviación lateral (B1) es 20;
6. El número de barras usadas para encontrar la desviación axial (B2) es 2;
7. La matriz se rellena con ceros a 1024 por 1024 píxeles.

Debe entenderse que estos valores numéricos tienen fines ilustrativos únicamente; pueden emplearse otros valores numéricos sin apartarse de la naturaleza de la invención.

En referencia ahora a la Figura 1, se muestra un diagrama de bloques funcional de una realización de ejemplo de un procedimiento para corrección del movimiento relativo de un detector de objetos entre vistas sucesivas construido de acuerdo con las enseñanzas de la presente invención. En la realización de ejemplo, se genera una copia alterada de cada imagen, en la que los píxeles más luminosos se reasignan a un nivel de brillo cero, mientras que todos los otros píxeles retienen el mismo brillo que en la imagen inicial. Entonces, se multiplica una FFT en dos dimensiones (2D) de esta imagen con formación de umbral, píxel por píxel, por el conjugado complejo de la FFT en 2D de una imagen de referencia. El brillo de la matriz resultante se suma entonces a lo largo de cada línea paralela al eje del tubo microcapilar (que se llamará de aquí en adelante eje Y) para desarrollar una matriz en una dimensión (1D) que contiene información sobre el patrón de brillo en la dirección (que se denominará de aquí en adelante como eje X) perpendicular al eje óptico y al eje microcapilar. Entonces, se aplica una FFT en 1D, y se determina la ubicación del máximo. La ubicación determina la cantidad de desviación que se aplicará a lo largo del eje X para llevar la imagen

del eje central del tubo microcapilar a su posición deseada en la imagen.

La imagen de referencia tiene la ventaja de la constancia de la separación entre las paredes del tubo microcapilar, y consiste en dos líneas brillantes al máximo separadas por la cantidad conocida encontrada en las imágenes adquiridas; el resto de la imagen de referencia tiene un brillo cero. Las paredes del tubo aparecen solo débilmente en las imágenes de pseudoproyección, ya que los índices de refracción de las paredes del tubo coinciden con los materiales dentro del tubo y entre el tubo y el conjunto de platina/cubreobjetos. El efecto de la operación de histograma es mejorar el contraste entre las paredes del tubo y el resto de la imagen. Usando la separación predeterminada de la pared del tubo, en conjunto con el número conocido de píxeles a lo largo del eje X de la imagen, se hace posible distinguir el movimiento del propio tubo del movimiento de los objetos dentro del tubo, debido a la rotación del tubo y al cambio consecuente de perspectiva. Al correlacionar mutuamente las dos imágenes basándose en una característica constante, nuestro procedimiento minimiza la posibilidad de rastrear los movimientos de características cambiantes dentro de la célula.

Se usa un procedimiento de correlación mutua para determinar la cantidad de la desviación axial con respecto al eje Y. Para hacer esto, se forma de nuevo el umbral de una copia de la imagen original, pero usando criterios diferentes para determinar qué píxeles se restablecen en un brillo cero. Se aplica una FFT en 2D en esta imagen, y se multiplica, píxel por píxel, por el conjugado complejo de la FFT en 2D de la imagen con formación de umbral derivada de la vista inmediatamente anterior. Una FFT en 2D se aplica al resultado, y la desviación del eje X se determina como el máximo en la función de correlación mutua a lo largo de la línea que se corresponde con la diferencia en la corrección lateral de la imagen actual con la de la imagen anterior. Esto es una distinción de los procedimientos anteriores, ya que la desviación del eje X se limita mediante la desviación del eje Y; y no se encuentra independientemente de la desviación del eje Y.

A diferencia de la corrección lateral, la corrección axial es un procedimiento iterativo y de esta manera está sometido a errores acumulativos. La correlación mutua axial funciona eficazmente, sin embargo, siempre y cuando el cambio en perspectiva entre imágenes consecutivas no sea demasiado grande; esto se corresponde con pequeños incrementos angulares en la rotación. Al mantener pequeño el incremento angular, el contenido espacial no varía mucho, permitiendo que la correlación mutua rastree características similares en cada imagen. Ya que el incremento angular también determina la resolución lateral de la reconstrucción tomográfica en 3D, no es oneroso el requerimiento de que el incremento angular se mantenga pequeño para permitir que la correlación mutua funcione bien.

Tal como se ha mencionado brevemente, esta realización de la presente invención elimina los efectos del movimiento axial y lateral mediante el tubo microcapilar desplazando adecuadamente las imágenes posteriores para alinearlas con imágenes anteriores, o viceversa. Los procedimientos de correlación mutua se usan para encontrar la desviación en el eje lateral, y después en el eje del tubo, con la restricción de que la correlación máxima para el movimiento axial debe llegar tras la determinación del movimiento lateral.

La primera etapa 111 es generar la imagen de plantilla. Dos líneas blancas que tienen, por ejemplo, un nivel de escala de grises de 65.535, se crean en sus posiciones ideales. Cada línea tiene una longitud de 480 píxeles, que se desarrollan en paralelo a la dimensión corta de la imagen. Las ubicaciones de las dos líneas se determinan mediante la dimensión larga de la imagen (640 píxeles) y la separación de la pared del tubo, determinada empíricamente como 530 píxeles. La primera línea se ubica en la línea 0 y la segunda línea se ubica en la línea 530. En esta realización, el tamaño de la imagen de plantilla puede expandirse desde 640x480 a 1024x1024 para proporcionar un relleno con ceros en ambas dimensiones; sin embargo, esta acción no es esencial para la invención.

Una FFT en 2D se realiza en la imagen de plantilla para que los componentes reales e imaginarios se guarden en índices alternativos de la matriz resultante. De esta manera, para una matriz rellena con ceros, el tamaño de matriz es 2048x1024. La imagen de plantilla está ahora en un formato listo para su uso.

En la etapa 114 se encuentra la desviación lateral. En la etapa 114, se realiza una formación de umbral de la imagen para oscurecer los píxeles de fondo, y después se realiza una correlación mutua con la imagen binaria de las dos líneas brillantes. Las imágenes de interés se someten a la determinación de desviación lateral 114. Para ayudar en la corrección axial,  $D_{LAT}$  se guarda para cada imagen.

En referencia ahora a la Figura 2, se muestra un diagrama de bloques funcional de una porción de corrección lateral de un sistema de formación de imágenes que emplea la realización de ejemplo descrita en la Figura 1. Las etapas implicadas para encontrar la desviación lateral 114 incluyen construir un histograma de escala de grises de la imagen, donde el número de barras ( $B_1$ ) puede establecerse en cualquier valor entero desde 2 a 255. Para el presente ejemplo, se asume que  $B_1=20$ . Se encuentra la barra con el número mayor de píxeles (excepto por la primera barra, que se corresponde con los píxeles más oscuros), y todos los píxeles en la imagen original que tienen el valor de escala de grises de esa barra o mayor se establecen igual a cero en una copia de la imagen original. El efecto de este procedimiento 21 es retirar los píxeles de fondo de consideraciones posteriores para producir una imagen con formación de umbral.

Como un ejemplo, se supone que la imagen de entrada tiene valores de escala de grises mínimos y máximos de 31

y 190, respectivamente, por lo que cada barra tiene una anchura de ocho niveles de grises  $[(1+190+31)/20=8]$ . Ahora se supone además que el máximo en el histograma ocurre en la barra #16 (es decir, niveles de gris de 151 a 158). Después, la imagen con formación de umbral será similar a la imagen original, excepto que todos los píxeles con un nivel de gris inicial mayor de 150 tienen ahora un nivel de gris de cero.

5 Las Figuras 4A-4C ilustran el efecto de aplicar estas etapas 21 a una imagen. Una comparación visual de una imagen original con una imagen segmentada puede realizarse en referencia a la Figura 4A, que muestra un ejemplo de una imagen de célula antes de la segmentación y la formación de umbrales, y después en referencia a la Figura 4B que muestra un ejemplo de una imagen de célula segmentada y con formación de umbral que se corresponde con la imagen original de la Figura 4A. La Figura 4C es un histograma de una imagen de ejemplo que muestra una  
10 comparación de los niveles de gris de la imagen original y la imagen tras la formación de umbrales.

Se aplica una FFT 2D en la imagen 22 con formación de umbral, y su transformada de Fourier se multiplica 23 por el conjugado complejo de la transformada de Fourier de la imagen de plantilla. La matriz resultante se suma 24 a lo largo de cada una de las 640 filas para computar una nueva matriz, que es la transformada de Fourier (en 1D) 25 para encontrar la correlación mutua de las filas de la imagen con formación de umbral y la imagen de referencia. El valor máximo de la matriz en 1D se ubica 26 y se evalúa 28. La posición del máximo se designa como  $D_{LAT}$  y su magnitud se designa como  $C_{MAX}$ .

La desviación necesaria se determina mediante la diferencia entre  $D_{LAT}$  y su posición ideal de  $55[(640-530)/2=55]$ . De esta manera, por ejemplo, si  $D_{LAT}=63$ , entonces es necesario un desplazamiento ascendente de 8 píxeles ( $63-55=8$ ), mientras que si  $D_{LAT}=41$ , entonces es necesario un desplazamiento descendente de 14 píxeles ( $55-41=14$ ).

20 El procedimiento 114 se repite para todas las imágenes en el conjunto de datos. Debe apreciarse que cada imagen hace referencia a la misma plantilla, por lo que no existe error acumulativo. Para ayudar en la corrección axial,  $D_{LAT}$  se guarda 29 para cada imagen.

En referencia ahora a la Figura 3, se muestra un diagrama de bloques funcional de una porción de corrección axial de un sistema de formación de imágenes que emplea la realización de ejemplo descrita en la Figura 1. La corrección axial 115 se realiza en todas las imágenes excepto en la primera. Se forma el umbral de una copia de la imagen de entrada en la etapa 31, y después se realiza una correlación mutua con una copia con formación de umbral de la imagen anterior. La desviación se determina como el máximo en la función de correlación mutua a lo largo de la línea que se corresponde con la diferencia en la corrección lateral para la perspectiva actual  $[D_{LAT}(N)]$  y la corrección lateral para la perspectiva inmediatamente anterior  $[D_{LAT}(N-1)]$ . A diferencia de la corrección lateral 114, por tanto, la corrección axial 115 es un procedimiento iterativo y de esta manera está sometido a errores acumulativos.

Se realiza una formación de umbral 31 de una copia de una imagen de entrada de la misma manera que para la corrección lateral, pero en este caso el número de barras en el histograma es  $B3$ . En el ejemplo presente,  $B3=2$ . De esta manera, todos los píxeles con un nivel de grises mayor que el nivel de grises de intervalo medio se establecen en cero, mientras que aquellos con niveles de grises menores retienen sus valores iniciales. Por ejemplo, una imagen de entrada con valores mínimos y máximos de 31 y 190, respectivamente, tendrá como resultado una imagen con formación de umbral idéntica a la inicial, excepto que todos los píxeles que eran inicialmente más brillantes que 110 ahora son cero.

Habiendo oscurecido de esta manera los píxeles brillantes, la imagen con formación de umbral sufre una transformada de Fourier en 2D 32. Después se filtra 33 para eliminar las características más pequeñas, que pueden producir picos espurios en la correlación mutua. Solo se multiplican las frecuencias espaciales de hasta 102 ciclos/píxel, que se corresponden con tamaños especiales de diez píxeles o menos, y los píxeles a mayores frecuencias espaciales se establecen en cero. La matriz resultante se guarda 34 como  $S_N$  y se multiplica 35 por el conjugado complejo de  $S_{N-1}$ , obtenido a partir de la copia con formación de umbral de la imagen anterior. Una FFT en 2D se aplica después a la matriz resultante para encontrar la correlación mutua de las dos imágenes consecutivas, con formación de umbral y con filtro de paso bajo. La diferencia en la desviación lateral entre las dos imágenes consecutivas  $[D_{LAT}(N) - D_{LAT}(N-1)]$  encontrada a partir de la etapa 114 de corrección lateral es ahora necesaria, ya que es incorrecto encontrar el máximo global de la matriz de correlación. En su lugar, debe encontrarse un máximo local,  $F_{MAX}$ , en la fila que se corresponde con  $[D_{LAT}(N) - D_{LAT}(N-1)]$ . La columna que contiene  $F_{MAX}$  se designa como  $G_{MAX}$ . Si  $G_{MAX}$  es mayor que la mitad de la dimensión rellenada de la imagen (1024, en este ejemplo), entonces su valor significa un desplazamiento negativo, en relación con la imagen anterior, teniendo una magnitud igual que la dimensión rellenada con ceros menos el valor de  $G_{MAX}$ . Si  $G_{MAX}$  es menor que la mitad de la dimensión rellenada con ceros, entonces el desplazamiento requerido, en relación con la imagen anterior, es positivo e igual a  $G_{MAX}$ .

Como un ejemplo, se supone que  $D_{LAT}(N-1)=45$ , mientras que  $D_{LAT}(N)=63$ . Entonces,  $F_{MAX}$  se encontrará en la fila 18 de la matriz de correlación ( $63-45=18$ ). Si  $F_{MAX}$ , el valor máximo de la fila 18, ocurre en la quinta columna, entonces  $G_{MAX}=5$  y la imagen debe desplazarse cinco píxeles a la izquierda de la imagen anterior. Si el máximo ocurre en la fila 1019 ( $G_{MAX}=1019$ ), entonces la imagen debe desplazarse cinco píxeles a la derecha ( $1024-1019=5$ ) de la imagen anterior, ya que 1019 es mayor que 512.

Después de encontrar  $G_{MAX}$ , el valor del desplazamiento se añade a la suma de todas las desviaciones axiales previas para determinar  $D_{AXIAL}$ , la diferencia acumulativa desde la primera imagen adquirida hasta la imagen actual. El desplazamiento puede ser positivo o negativo; por tanto, algunas imágenes pueden no necesitar ningún desplazamiento axial. Para cada imagen, se escriben cuatro valores en un archivo de texto:

- 5        1. La posición de la pared del tubo superior,  $D_{LAT}$ ;
2.  $C_{MAX}$ , el valor máximo de la correlación mutua entre la imagen actual y la imagen de referencia;
3.  $G_{MAX}$ , la ubicación de  $F_{MAX}$  en la fila apropiada de la correlación mutua entre la imagen actual y la imagen anterior (para la primera imagen,  $G_{MAX} = 0$ );
4.  $F_{MAX}$  (para la primera imagen,  $F_{MAX} = 0$ ).

- 10      El archivo corregido se genera recortando el número apropiado de píxeles desde uno o dos bordes y desplazando los píxeles restantes mediante el número recortado. Para mantener las dimensiones originales de la imagen (640x480), los espacios en los bordes opuestos desde los bordes recortados se sustituyen por píxeles establecidos en el máximo nivel de grises de la imagen original.

- 15      Por ejemplo, se supone que para una de las imágenes, el nivel de grises máximo es 229,  $D_{LAT}=63$ ,  $D_{AXIAL}= 29$ , y  $G_{MAX} = 1022$ . Entonces, los píxeles en las ocho filas superiores (63-55=8) y las 27 columnas izquierdas (29-1024+1022 = 27) se eliminan de la imagen. De esta manera, la novena fila de la columna 28 ocupa la esquina superior izquierda. Ocho filas se añaden a la parte inferior de la imagen, y 28 columnas se añaden a la derecha de la imagen; estos píxeles tienen niveles de grises de 229. Cuando estos procedimientos se completan, la región de 632x453 píxeles en la izquierda superior de la imagen corregida es idéntica a la región de 632x453 píxeles en la derecha inferior de la imagen original. Ambas imágenes tienen dimensiones de 640x480.
- 20

Otra realización de ejemplo incorpora solamente la corrección axial 115 y la escritura de la imagen corregida 116. Esta realización es útil cuando las paredes del tubo microcapilar no son visibles y se conoce que el movimiento lateral del tubo es insignificamente pequeño.

- 25      En otra realización adicional, la separación de la pared del tubo se calcula automáticamente a partir de la primera vista ( $N=0$ ). De lo contrario, es idéntica a la realización descrita anteriormente en referencia a las Figuras 1-3. En otra realización de la invención, la separación de las paredes del tubo se determina basándose en un cálculo de su separación en una o más de las imágenes. Esto se logra usando como referencia una imagen derivada de una única línea brillante, tal como mediante una FFT en 2D y una conjugación compleja. Las filas se suman, como en la primera realización, y la ubicación del máximo se toma como la ubicación de una pared del tubo en relación con su ubicación en la imagen a partir de la que se derivó la imagen de referencia. La ubicación del siguiente valor de correlación más alto proporciona la ubicación de la otra pared del tubo, en relación con la primera. En caso deseado, la búsqueda de este máximo secundario puede limitarse a un intervalo cuya ubicación central, en relación con la primera pared del tubo, se encuentra en las proximidades de la anchura presumida del tubo. Esta realización también abarca la posibilidad de usar la referencia de línea única para los puntos de vista adquiridos. Tal disposición puede ser útil cuando la separación de la pared del tubo no es conocida, o cuando las paredes interiores del tubo no forman un círculo, como cuando el perfil interior del tubo es cuadrado o elíptico.
- 30
- 35

- En otra realización de la invención, las características de la etapa de formación de umbral pueden variar basándose en información de la correlación. Tales enfoques iterativos pueden emplearse en la primera etapa de formación de umbral para la corrección lateral, en la segunda etapa de formación de umbral para la corrección axial, o en ambas. Una característica que puede variar es el número de divisiones o barras usadas en el histograma. Otra característica que puede variar es el número de niveles de gris contenidos dentro de cada barra del histograma. Por ejemplo, el histograma puede basarse en la raíz cuadrada del nivel de brillo.
- 40

- De acuerdo con una característica de la invención, la salida del procedimiento es una copia recortada del archivo de entrada, con las porciones no cortadas desplazadas verticalmente y/u horizontalmente, y con píxeles vacíos adicionales insertados en uno o dos de los límites para retener el tamaño de la imagen de entrada.
- 45

- De acuerdo con una característica adicional de la invención, los resultados del procedimiento empleado se guardan en un archivo digital, que puede alterarse y editarse usando aplicaciones informáticas de procesamiento de texto. El archivo de texto alterado puede usarse entonces para generar las desviaciones en los dos ejes, evitando de esta manera muchos de los cálculos antes descritos. En esta realización, el procedimiento de corrección lateral de las etapas 114 a 116 se repite para encontrar el máximo de  $C_{MAX}$ . Si  $C_{MAX}$  tiene una magnitud menor que un valor crítico  $C_{CRIT}$ , entonces todo el procedimiento se repite, comenzando con la formación de umbrales 27, pero con el número de barras en el histograma cambiado de B1 a B2. De nuevo,  $C_{MAX}$  se ubica 26 y se evalúa 28.
- 50

- Ahora, en referencia a la Figura 5, se muestra esquemáticamente una ilustración de ejemplo de células empaquetadas en un tubo capilar tal como se contempla mediante una realización de la presente invención. En esta realización de ejemplo, una sección del tubo capilar 3 se llena con objetos de interés 1, tal como células, que se empaquetan rígidamente en el tubo. Cada una de las células puede incluir un núcleo 2. El tubo capilar 3 tiene un eje central 4 orientado en referencia a un sistema 6 de coordenadas que tiene coordenadas en las direcciones x, y, y z. en algunos casos, al menos una sonda molecular 153 puede unirse dentro de la célula. Un ordenador 7 se acopla
- 55

para proporcionar señales de control a un motor rotativo 5 y un motor 8 de traslación. Se reconocerá que también pueden emplearse disposiciones equivalentes de uno o más motores, engranajes o fluidica y otros medios de generación de movimiento para lograr el movimiento de traslación y rotativo necesario del tubo capilar u otro sustrato. En algunos casos, uno o más de los motores pueden sustituirse por dispositivos o engranajes de colocación manual o por otro medio de generación de movimiento tal como dispositivos hidráulicos o piezoeléctricos. El eje de traslación es el eje z, y la rotación es alrededor del eje z. El motor 9 de colocación se acopla para mover la célula en un plano definido por los ejes x, y, sustancialmente perpendicular al eje central con el fin de conseguir el centrado, según sea necesario.

Se reconocerá que la superficie curvada del tubo capilar actuará como una lente cilíndrica y que este efecto de enfoque puede no ser aconsejable en un sistema de proyección. Los expertos en la materia apreciarán que la flexión de fotones mediante el tubo puede eliminarse si los espacios entre la fuente de punto y el tubo y entre el tubo y las superficies detectoras se llenan con un material 10 cuyo índice de refracción coincide con el del tubo capilar, y que el tubo puede acoplarse ópticamente (con aceite o un gel, por ejemplo) al material que rellena el espacio.

Considérese el presente ejemplo de células empaquetadas en un tubo capilar. Las células pueden empaquetarse preferentemente en un único archivo para que no se superpongan. La densidad de empaquetar células completas de aproximadamente 100 micrómetros de diámetro en un tubo capilar con un diámetro menor de 100 micrómetros puede ser aproximadamente 100 células por centímetro de longitud del tubo. Para núcleos básicos de aproximadamente 20 micrómetros de diámetro, el empaquetado puede ser aproximadamente 500 núcleos por centímetro de longitud del tubo, donde el diámetro del tubo es proporcional al tamaño del objeto, alrededor de 20 micrómetros en este caso. De esta manera, dentro de varios centímetros de longitud del tubo capilar, pueden empaquetarse varios miles de núcleos básicos no superpuestos. Al trasladar el tubo a lo largo de su eje central 4, puede lograrse el movimiento en la dirección z. Mover el tubo en las direcciones x e y permite que los objetos dentro del tubo se centren, según sea necesario, en el cilindro de reconstrucción del sistema de tomografía óptica. Al rotar el tubo alrededor de su eje central 4, pueden producirse una multiplicidad de vistas de proyección radial. Mover el tubo en la dirección z con una velocidad constante y sin rotación estimula el caso especial de tomografía óptica de flujo.

Una ventaja de mover un tubo lleno de células, que de lo contrario son estacionarias dentro del tubo, es que los objetos de interés pueden detenerse, y luego rotarse, a velocidades que permiten una exposición casi óptima para la tomografía óptica basándose en las células una a una. Es decir, la relación de la señal con respecto al ruido de las imágenes de proyección puede mejorarse para producir mejores imágenes de las que pueden producirse normalmente a velocidades y dirección constantes típicas de los sistemas de flujo. Los objetos que no son de interés pueden moverse fuera del sistema de formación de imágenes rápidamente, para obtener una velocidad general al analizar células de interés en una mezcla que consiste en una multitud de células. Adicionalmente, la capacidad de detenerse en un objeto de interés, y después rotar según sea necesario para múltiples proyecciones, elimina prácticamente los artefactos en movimiento. Adicionalmente, el sistema de movimiento puede guiarse en movimientos de submicrómetros y puede aplicarse ventajosamente de una manera que permite un muestreo de la célula a una resolución más nítida de lo que permite el tamaño de píxel del detector. Más en particular, el factor de muestreo Nyquist de 2 podría gestionarse mediante el sistema de movimiento moviéndose en incrementos que rellenan la mitad de la anchura de un píxel, por ejemplo. De manera similar, el sistema de movimiento puede compensar el factor de forma imperfecto del detector.

Ahora, en referencia a la Figura 6A, se muestra una vista cercana de un único espécimen, tal como por ejemplo una única célula, sumergida dentro de un medio de material de indexación óptica. El único espécimen se muestra dentro de un tubo microcapilar 3 (por ejemplo, uno de tales tubos se fabrica por parte de Polymicro Technologies, LLC., AZ, Estados Unidos) que puede rotarse para adoptar múltiples proyecciones, y un objetivo 40 que puede escanearse axialmente se muestra esquemáticamente. Una fuente de iluminación incluye una fuente 50 de luz que proyecta luz a través de una abertura 51, un retén 52, y a través de una lente condensadora 53 que se coloca delante de una platina 54 de microscopio. Un tubo microcapilar 3 contiene una célula 1 entre la platina y un cubreobjetos 55 delgado. Un objetivo 40, preferentemente una lente de inmersión en aceite, está dispuesto para recibir luz que pasa a través del tubo microcapilar 3. El objetivo se traslada a lo largo del eje óptico mediante un accionador 57 tal como un elemento piezoeléctrico. El cubreobjetos 55 debe ser lo suficientemente fino para que la distancia entre el centro del tubo microcapilar y la superficie exterior del cubreobjetos sea más pequeña que la distancia de trabajo del objetivo. La lente condensadora 53 está dentro del índice de refracción  $n_1$  (por ejemplo, aire). La platina 54 y el cubreobjetos 55 tienen un índice de refracción  $n_2$ . Una región 58 alrededor del tubo microcapilar 3 contiene un medio 15 de coincidencia de índice tal como gel óptico o aceite de inmersión, que tiene un índice de refracción  $n_3$ . El propio tubo microcapilar 3 tiene un índice de refracción  $n_4$ . La región 59 alrededor de la célula 1 dentro del tubo microcapilar contiene un medio 10 que posee un índice de refracción  $n_5$ . Una región 60 dentro de la célula puede rellenarse con el mismo medio 10, o puede diferenciarse en su índice de refracción  $n_6$ . Se prefiere que  $n_3 = n_4 = n_5 = n_6$  (las diferencias deben minimizarse) entre las dos superficies paralelas y planas formadas mediante la platina 54 y el cubreobjetos 55 para evitar una distorsión de la lente cilíndrica. La imagen se proyecta sobre una cámara 43.

En referencia ahora a la Figura 6A y la Figura 6B, se muestra esquemáticamente una realización de un sistema de tomografía óptica empleado en la presente invención, que incorpora un objetivo de microscopio montado en un dispositivo de traslación piezoeléctrica. El transductor 57 piezoeléctrico se usa para mover un objetivo 60 una

5 distancia axial de aproximadamente 40 micrómetros o más. En una realización útil, un sistema de colocación de microobjetivo proporciona un accionador 57 adecuado, que se desplaza hacia arriba y hacia abajo a lo largo del eje z del sistema 6 de coordenadas del tubo. En esta realización, puede usarse con un gran objetivo numérico de abertura, montado en un microscopio 64 de transmisión estándar con una videocámara 43 unida y un conjunto 61 controlado por ordenador de fuente de luz y lente condensadora. El condensador y la fuente de luz 50 controlados por ordenador pueden ser ventajosamente una fuente de luz que incluye una o más bombillas incandescentes, una lámpara de arco, un láser o un diodo emisor de luz. Unas señales 70 de control de ordenador se vinculan con el condensador y la fuente de luz 50 controlados por ordenador para controlar la modulación de la luz.

10 La salida desde la cámara 43 se almacena en una memoria informática 72. Un tubo microcapilar 3 que contiene el espécimen puede trasladarse a lo largo de los ejes x o y del sistema 6 de coordenadas del tubo. Además, el tubo microcapilar 3 puede rotar alrededor de su eje "θ" 49, por medio de un motor rotativo 5 que puede controlarse por ordenador. Tal como se usa en el presente documento, el tubo microcapilar se define como un tubo capilar que tiene un diámetro donde el campo de visión para la formación de imágenes microscópicas es comparable con el diámetro del tubo capilar. En una realización de ejemplo, el motor rotativo 5 se controla mediante señales 71 de control proporcionadas mediante el ordenador 7. Para aplicaciones de alta velocidad pueden añadirse otros controles para reducir las vibraciones durante un escaneo axial. La imagen adquirida puede mostrarse en el monitor 73.

15 Ahora, en referencia a la Figura 7, se muestra un diagrama de flujo de ejemplo que ilustra un proceso para adquirir imágenes usadas en una reconstrucción de imágenes en tres dimensiones (3D) tal como se contempla mediante una realización de la presente invención. Tal como se contempla mediante un ejemplo de la presente invención, un proceso de reconstrucción de imagen en 3D incluye las etapas de cargar el tubo empaquetado con células en la etapa 81, trasladar el tubo hasta que la primera célula de interés se ha ubicado en la etapa 82, centrar la célula de interés, según sea necesario, en la etapa 83, generar un conjunto de proyecciones en cada ángulo diferente de rotación en la etapa 84, determinar cuándo el conjunto de datos está completo en la etapa 85, y repetir el proceso desde las etapas 82 a 85 hasta que todas las células de interés se han escaneado. En la etapa 86, se realizan correcciones de movimiento. El proceso puede implementarse en un programa de software informático ejecutado mediante un ordenador personal tal como el ordenador 7, por ejemplo.

20 Ahora, en referencia a la Figura 8, se muestra esquemáticamente un ejemplo de desviaciones de corrección de movimiento para su uso en una reconstrucción de imagen en tres dimensiones (3D) tal como se contempla mediante una realización de la presente invención. La corrección de movimiento se aplica para encontrar la posición lateral de un objeto 1 de interés, tal como una célula o núcleo 2, contenida en un tubo capilar 3 que tiene una pared 62 del tubo capilar. La desviación lateral es el error a lo largo de la dimensión más larga de la imagen (640 píxeles), perpendicular al eje Z del tubo. La desviación axial es el error a lo largo de la dimensión más corta de la imagen, (480 píxeles), en paralelo al eje Z del tubo. El objeto 1 de interés tiene una posición lateral LP y una posición axial AP. Ya que las imágenes se adquieren desde diversos puntos de vista, la corrección de movimiento se aplica para permitir la reconstrucción del objeto de interés con características idénticas mantenidas en el mismo plano en las diversas vistas.

#### Ejemplo de descripción del código de software

##### Código fuente

30 A continuación aparece el texto de un ejemplo de código fuente para implementar una realización del procedimiento de la invención para corrección de movimiento. El archivo ejecutable (regcorr.exe) se construye a partir de dos archivos de cabecera (*rrahn.h* y *regcorr.h*) y cuatro archivos C++ (*fileinfo.cpp*, *nr-fft.cpp*, *multiplycomplexvalues.cpp*, y *regcorr2.cpp*), unidos mediante el proyecto *regcorr.ide*. El proyecto se compiló por parte de Borland C++ 5.01, configurado para el modo "Win32 Console", y usando la infraestructura de biblioteca de clase estática. El tamaño del archivo ejecutable es aproximadamente 60 kB. (Los programas, "regcorr.exe" y "regcorr\_no\_lateral.exe," solo se diferencian en si proporcionan la opción de calcular la desviación lateral y escribirla en un archivo de texto. Ambos programas pueden saltarse los cálculos y leer en su lugar las desviaciones desde un archivo de texto.)

##### A.1 *rrahn.h* (38 líneas)

```

50 #include <math.h>
#include <malloc.h>
#include <fstream.h>
#include <dos.h>
#include <iostream.h>
#include <stdio.h>
55 #include <conio.h>
#include <complex.h>
#pragma hdrstop
#pragma package(smart_init)

#define pi_const 3.14159265358979323846

```

```

#define unsigned int

template<class T>
inline const T SQR(const T a) {return a*a;}
5 inline float pow(float x, double y) {return pow(double(x), y);} inline float pow(double x, float y) {return
pow(x, double(y));} inline float atan2(float x, double
y) {return atan2(double(x), y);}
10 inline float atan2(double x, float y) {return atan2(x,
double(y));}

template <class T>
inline void SWAP(T &a, T &b)
15 {T dum=a; a=b; b=dum;}

#define SwapFFT SWAP

#define wait while(!kbhit()); return;

20 unsigned *Ulivector(unsigned, unsigned);

```

*A.2 regcorr.h (40 líneas)*

```

#include "rrahn.h"
#pragma hdrstop
25 #pragma package(smart_init)

const unsigned ImageDimX = 640;
const unsigned ImageDimZ = 480;
const unsigned BigDimX = 1024;
30 const unsigned BigDimZ = 1024;
const unsigned NumHistoBins = 20;
const unsigned WallSpacingMin = 505, WallSpacingMax = 565;

35 extern unsigned FileYear, FileMonth, FileDay, FileSet, NumPerspectives, CurrentPerspective, Bin-
Hist[NumHistoBins];
extern int SliceNumber;
extern char filenameIn[64];
extern char ProcessedFilenameOut[64];
40 extern float RawImage[ImageDimX*ImageDimZ];
extern double ShiftedImage[BigDimX*BigDimZ*2], ShiftedImageNew[BigDimX*BigDimZ*2];
extern double TemplateIm-
age[BigDimX*BigDimZ*2], TemplateImageNew[BigDimX*BigDimZ*2];
extern int ShiftX;
45 extern int ShiftZ;
extern double MaxVal, MinVal;
extern float BinVal[ImageDimX*ImageDimZ];
extern unsigned MaxBin, MaxHisto;
extern int BigDims[2];
50 extern char* DirectoryName;
extern unsigned WallEdge1, WallEdge2, WallSpacing;
extern double MinValDev1, MinValDev2, MaxVal- Corr, MaxValCorr2;
extern unsigned MinDevIndex, MaxCorrIndex, MaxCorrIndex2;
extern unsigned MinDevIndexOld;
55 extern double ShiftedMag[2*BigDimX*BigDimZ];
extern unsigned ppmax;
extern float CriticalValue;

void MultiDimFFT(double[],int[],int, int);
60 void FindDeviation(); void GetFileInfo();
void WriteCorrected Image(double[]);
void MultiplyComplexValues(double[], double[],double[], unsigned, unsigned,unsigned, unsigned);
void BasicFFT(float[], unsigned long, int);
void MakeFileName(char*, unsigned, unsigned);

```

**A.3 fileinfo.cpp (50 líneas)**

```

#include "rrahn.h"
unsigned NumPerspectives;
unsigned FileYear, FileMonth, FileDay, FileSet;
5 void GetFileInfo()
{
  cout << "Year (YY, 3 or 4): ";
  cin >> FileYear;
  while(FileYear != 3 && FileYear != 4)
10 {
  cout << "Year (YY, 3 or 4): ";
  cin >> FileYear;
}
  cout << "Month (MM):";
15 cin >> FileMonth; while(FileMonth>12)
  {
  cout << "Month (MM):";
  cin >> FileMonth;
}
  cout << endl << "Day (DD):";
20 cin >> FileDay;
  while(FileDay>31)
  {
  cout << "Day (DD):";
25 cin >> FileDay;
}
  cout << endl << "Set (1-99): ";
  cin >> FileSet;
  while(FileSet>99)
30 {
  cout << "Set (1-99): ";
  cin >> FileSet;
}
  cout << endl;
35 cout << "Number of Perspectives (1-255):";
  cin >> NumPerspectives;
  while(NumPerspectives>255 ||
  NumPerspectives<1)
  {
40   cout << "Number of Perspectives (1-255):";
   cin >> NumPerspectives;
  }
  return;
}
45

```

**A.4 nr-fft.cpp (221 líneas)**

```

#include "rrahn.h"
void BasicFFT(float FFTdata[], unsigned long FFTnn, int FFTisign)
50 {
  unsigned long FFTm, FFTistep;
  unsigned long FFTmmax = 2;
  double FFTwtemp, FFTwr, FFTwpr, FFTwpi, FFTwi, FFTtheta;
  float FFTtemp, FFTtempi;
55 unsigned long FFTn = FFTnn << 1;
  unsigned long FFTj = 1;
  for(unsigned long jj = 1; jj < FFTn; jj += 2)
  if (FFTj > jj)
  {
60   SwapFFT(FFTdata[FFTj], FFTdata[jj]);
   SwapFFT(FFTdata[FFTj+1], FFTdata[jj+1]);
  }
  FFTm = FFTn >> 1;.
  while(FFTm >= 2 && FFTj > FFTm)
65 {

```

```

FFTj -= FFTm;
FFTm = FFTm >> 1;
}
FFTj += FFTm;
5 }
while(FFTN > FFTmmax) {
FFTistep = FFTmmax << 1;
FFTtheta = FFTisign*(2.0*pi_const/FFTmmax);
FFTwtemp = sin(0.5*FFTtheta);
10 FFTwpr = -2.0*FFTwtemp*FFTwtemp;
FFTwpi = sin(FFTtheta);
FFTwr = 1.0;
FFTw = 0.0;
for(FFTm = 1; FFTm < FFTmmax; FFTm += 2)
15 {
for(unsigned long jj = FFTm; jj <= FFTN; jj +=
FFTistep)
{
FFTj = jj + FFTmmax;
20 FFTtempr = FFTwr*FFTdata[FFTj] - FFTwi*FFTdata[FFTj+1];
FFTtempi = FFTwr*FFTdata[FFTj+1] + FFTwi*FFTdata[FFTj];
FFTdata[FFTj] = FFTdata[jj] - FFTtempr;
FFTdata[FFTj + 1] = FFTdata[jj+1] - FFTtempi;
25 FFTdata[jj] += FFTtempr;
FFTdata[jj+1] += FFTtempi;
}
FFTwtemp = FFTwr;
FFTwr = FFTwtemp*FFTwpr - FFTwi*FFTwpi + FFTwr;
30 FFTwi = FFTwi*FFTwpr + FFTwtemp*FFTwpi + FFTwi;
}
FFTmmax = FFTistep;
}
return;
}
35 void ApplyOneDimFFT(float FFTdata[], unsigned long FFTN, int FFTisign)
{
unsigned long FFTi, FFTi1, FFTi2, FFTi3, FFTi4, FFTnp3;
float FFTc1=0.5, FFTc2, FFTh1r, FFTh1i, FFTh2r, FFTh2i;
double FFTwr, FFTwi, FFTwpr, FFTwpi, FFTwtemp, FFTtheta;
40 FFTtheta = pi_const/(double)(FFTN>>1);
void BasicFFT(float FFTdata[], unsigned long FFTn, int FFTisign);
if (FFTisign == 1)
{
45 FFTc2 = -0.5;
BasicFFT(FFTdata, FFTN>>1, 1);}
else {
FFTc2 = 0.5;
FFTtheta = -FFTtheta;
}
50 FFTwtemp = sin(0.5*FFTtheta);
FFTwpr = -2.0*FFTwtemp*FFTwtemp;
FFTwpi = sin(FFTtheta);
FFTwr = 1.0 + FFTwpr;
FFTw = FFTwpi;
55 FFTnp3 = FFTN + 3;
for (FFTi=2; FFTi<=(FFTN>>2); FFTi++)
{
FFTi1 = FFTi + FFTi - 1;
FFTi2 = 1 + FFTi1;
60 FFTi3 = FFTnp3 - FFTi2;
FFTi4 = 1 + FFTi3;
FFTh1r = FFTc1*(FFTdata[FFTi1] + FFTdata[FFTi3]);
FFTh1i = FFTc1*(FFTdata[FFTi2] - FFTdata[FFTi4]);
FFTh2r = -FFTc2*(FFTdata[FFTi2] + FFTdata[FFTi4]);
FFTh2i = FFTc2*(FFTdata[FFTi1] - FFTdata[FFTi3]);
65 FFTdata[FFTi1] = FFTh1r + FFTwr*FFTh2r - FFTwi*FFTh2i;
FFTdata[FFTi2] = FFTh1i + FFTwr*FFTh2i + FFTwi*FFTh2r;

```

## ES 2 559 411 T3

```

FFTdata[FFTi3] = FFThlr - FFTwr*FFTh2r + FFTwi*FFTh2i;
FFTdata[FFTi4] = -FFThli + FFTwr*FFTh2i + FFTwi*FFTh2r;
FFTwtemp = FFTwr;
5 FFTwr = FFTwtemp*FFTwpr - FFTwi*FFTwpi + FFTwr; //The recurrence
FFTwi = FFTwi *FFTwpr + FFTwtemp*FFTwpi + FFTwi;
}
if(FFTisign == 1)
{
10 FFThlr = FFTdata[1];
FFTdata[1] = FFThlr + FFTdata[2];
FFTdata[2] = FFThlr - FFTdata[2];
}
else
{
15 FFThlr = FFTdata[1];
FFTdata[1] = FFTc1*(FFThlr + FFTdata[2]);
FFTdata[2] = FFTc1*(FFThlr - FFTdata[2]);
BasicFFT(FFTdata, FFTn>>1, -1);
for(unsigned jj=0; jj<FFTN; jj++)
20 {
FFTdata[jj+1] = 2.0*FFTdata[jj+1]/FFTN;
}
}
25 void MultiDimFFT(double data[], int nn[], int ndim, int isign)
{
int i1, i2, i3, i2rev, i3rev, ip1, ip2, ip3, ifp1, ifp2;
int ibit, idim, k1, k2, n, nprev, nrem, ntot;
double tempi, tempr;
30 double theta, wi, wpi, wpr, wr, wtemp;
ntot = 1;
for(idim=1; idim<=ndim; idim++) ntot *= nn[idim];
nprev = 1;
for(idim=ndim; idim>=1; idim--) { n=nn[idim];
35 nrem = ntot/(n*nprev);
ip1 = nprev << 1;
ip2 = ip1*n;
ip3 = ip2*nrem;
i2rev = 1;
40 for(i2=1; i2<=ip2; i2+=ip1)
{
if(i2<i2rev)
{
45 for (i1=i2; i1<=(i2+ip1-2); i1+=2)
{
for(i3=i1; i3<=ip3; i3+=ip2)
{
i3rev = i2rev+i3-i2;
SwapFFT(data[i3], data[i3rev]);
50 SwapFFT(data[i3+1], data[i3rev+1]);
}
}
}
ibit = ip2 >> 1;
55 while( (ibit>=ip1) && (i2rev>ibit) )
{
i2rev -= ibit;
ibit >>= 1;
}
60 i2rev += ibit;
}
ifp1 = ip1; while(ifp1<ip2)
{
ifp2 = ifp1 << 1;
65 theta = isign*2*pi const/(ifp2/ip1);
wtemp = sin(0.5*theta);
wpr = -2.0*SQR(wtemp);

```

```

wpi = sin(theta);
wr = 1.0;
wi = 0.0;
5   for(i3=1; i3<=ifp1; i3+=ip1)
    {
      for(i1=i3; i1<=(i3+ip1-2); i1+=2)
        {
          for(i2=i1; i2<=ip3; i2+=ifp2)
            {
10          k1 = i2;
           k2 = k1+ifp1;
           tempr = wr*data[k2] - wi*data[k2+1];
           tempi = wr*data[k2+1] + wi*data[k2];
           data[k2] = data[k1] - tempr;
15          data[k2+1] = data[k1+1] - tempi;
           data[k1] += tempr;
           data[k1+1] += tempi;
            }
          }
20          wtemp = wr;
           wr = wtemp*wpr - wi*wpi + wr;
           wi = wi*wpr + wtemp*wpi + wi;
            }
           ifp1 = ifp2;
25          }
           nprev *= n;
           }
           return;
           }
30

```

#### A.5 multiplycomplexvalues.cpp (30 líneas)

```

#include "rrahn.h"
void MultiplyComplexValues(double MCV1[], double MCV2[], double MCV3[], unsigned
35 MCVDim1, unsigned MCVDim2, unsigned MaxFrequency1, unsigned MaxFrequency2)
    {
      for(unsigned mm=0; mm<MCVDim2; mm++)
        {
          for(unsigned nn=0; nn<MCVDim1; nn++)
            {
30          if( (mm>=MaxFrequency2 && mm<MCVDim2- MaxFrequency2)
                || (nn>=MaxFrequency1 &&
                    nn<MCVDim1-MaxFrequency1) )
              {
                MCV3[2*nn*MCVDim2+2*mm] = (double) 0.0;
45                MCV3[2*nn*MCVDim2+2*mm+1] = (double) 0.0;
              }
            else
              {
                MCV3[2*nn*MCVDim2+2*mm] = (double)
50                (MCV1[2*nn*MCVDim2+2*mm]
                 *MCV2[2*nn*MCVDim2+2*mm]
                 + MCV1[2*nn*MCVDim2+2*mm+1]
                 *MCV2[2*nn*MCVDim2+2*mm+1]);
                MCV3[2*nn*MCVDim2+2*mm+1] =
55                (double) (MCV2[2*nn*MCVDim2+2*mm]
                          *MCV1[2*nn*MCVDim2+2*mm+1]
                          + MCV2[2*nn*MCVDim2+2*mm+1]
                          *MCV1[2*nn*MCVDim2+2*mm]);
              }
            }
60          }
        }
      return;
    }

```

#### 65 A.6 regcorr2.cpp (374 líneas)

```

#include <strstream.h>
#include "regcorr.h"

NumPerspectives;
5 unsigned CurrentPerspective;
unsigned BinHist[NumHistoBins];
int SliceNumber; char filenameIn[64];
char ProcessedFilenameOut[64], IndexFilenameOut[64];
10 float RawImage[ImageDimX*ImageDimZ];
double ShiftedImage[BigDimX*BigDimZ*2];
double TemplateImage[BigDimX*BigDimZ*2], Tem-
plateImageNew[BigDimX*BigDimZ*2];
double OneLine[BigDimX*BigDimZ*2];
15 float LateralSum[2*BigDimX];
double MaxVal, MinVal;
float BinVal[ImageDimX*ImageDimZ];
unsigned MaxBin, MaxHisto, MinDevIndexOld;
int BigDims[2], SmallDims[2];
20 unsigned MaxCorrIndex, WallEdge1, WallEdge2;
double MaxValCorr;
double ShiftedMag[2*BigDimX*BigDimZ];
char *DirectoryName = "E:\\VisionGate\\Projection
Images\\";
25 unsigned nnc, WallOld1, WallOld2;
float LateralMax1 = 0, LateralMax2 = 0;
float MaxValOld, MinValOld;
unsigned SecondSeg = 2; unsigned FirstSeg = 20;
unsigned CorrectedFile[4*360];
30 float RedoLateralCorrection = 2.2e17;
int CumulativeShift = 0;

void ReadUncorrectedImage(unsigned RUI1)
{
35 MaxValOld = MaxVal;
MinValOld = MinVal;
MaxVal = 0;
MinVal = 1e22;
MakeFileName(filenameIn, RUI1, 2);
fstream RawFile(filenameIn, ios::in);
40 cout << filenameIn << endl;
for(unsigned mm=0; mm<ImageDimZ; mm++)
{
for(unsigned nn=0; nn<ImageDimX; nn++)
{
45 RawFile >> (float)RawImage[nn*ImageDimZ+mm];
if(RawImage[nn*ImageDimZ+mm]<MinVal) MinVal =
RawImage[nn*ImageDimZ+mm];
if(RawImage[nn*ImageDimZ+mm]>MaxVal) MaxVal =
50 RawImage[nn*ImageDimZ+mm];
}
}
RawFile.close();
return;
55 void CopyRawImage(double CRI1[])
{
for(unsigned mm=0; mm<BigDimZ; mm++)
for(unsigned nn=0; nn<BigDimX; nn++)
{
60 CRI1[2*nn*BigDimZ + 2*mm+1] = 0;
CRI1[2*nn*BigDimZ + 2*mm] = 0;
if(mm<ImageDimZ && nn<ImageDimX)
(double) CRI1[2*nn*BigDimZ + 2*mm] =
65 RawImage[nn*ImageDimZ+mm];
}
return;

```

```

}
void MakeOneLine()
{
for (unsigned jj=0; jj<ImageDimZ; jj++)
5   {
        OneLine[2*jj] = 65535;
        OneLine[2*jj+2*530*BigDimX] = 65535;
    }
MultiDimFFT(OneLine-1, BigDims-1, 2, 1);
10  return;
}
void MakeHistogram(double MHdata[], unsigned
MHBins)
{
15  if (MHBins>NumHistoBins)
MHBins = NumHistoBins;
MaxBin = 0;
MaxHisto = 0;
for (unsigned jj=0; jj<NumHistoBins; jj++)
20  BinHist [jj] = 0;
for (unsigned mm=0; mm<ImageDimZ; mm++)
{
    for (unsigned nn=0; nn<ImageDimX; nn++)
25  {
        BinVal [nn*ImageDimZ+mm] = (int)MHBins* (RawImage [nn*ImageDimZ+mm] -
        MinVal) / (MaxVal-MinVal);
        if (BinVal [nn*ImageDimZ+mm] >= MHBins) BinVal [nn*ImageDimZ+mm] = MHBins - 1u;
        if (BinVal [nn*ImageDimZ+mm] <= 0) BinVal [nn*ImageDimZ+mm] = 0u;
        BinHist [(int)BinVal [nn*ImageDimZ+mm]] += 1u;
30  }
    }
for (unsigned jj=0; jj<MHBins; jj++)
{
    if (BinHist [jj] > MaxHisto)
35  {
        MaxHisto = BinHist [jj];
        MaxBin = jj;
    }
}
40  for (unsigned jj=0; jj<2*BigDimX*BigDimZ; jj++)
    MHdata [jj] = 0;
for (unsigned mm=0; mm<ImageDimZ; mm++)
{
    for (unsigned nn=0; nn<ImageDimX; nn++)
45  {
        if (BinVal [nn*ImageDimZ+mm] < MaxBin) MHdata [2*nn*BigDimZ + 2*mm] =
        RawImage [nn*ImageDimZ+mm];
    }
}
50  return;
}
void LateralCorrection (unsigned LC1)
{
    MakeHistogram (ShiftedImage, LC1);
55  MultiDimFFT (ShiftedImage-1, BigDims-1, 2, 1);
MultiplyComplexValues (ShiftedImage, OneLine, TemplateImageNew,
BigDimX, BigDimZ,
                        BigDimX, BigDimZ);
for (unsigned mm=0; mm<BigDimX; mm++)
60  {
    LateralSum [2*mm] = 0;
    LateralSum [2*mm+1] = 0;
    for (unsigned nn=0; nn<BigDimZ; nn++)
{
65  LateralSum [2*mm] +=
        TemplateImageNew [2*mm*BigDimZ+2*nn];
        LateralSum [2*mm+1] += TemplateImageNew [2*mm*BigDimZ+2*nn+1];
}
}
}
}

```

```

    }
    }
    BasicFFT(LateralSum-1, BigDimX, -1);
    LateralMax1 = 0;
5   for(unsigned jj=0; jj<110; jj++)
    {
        if(LateralSum[2*jj] > LateralMax1)
        {
10          WallEdge1 = jj;
            LateralMax1 = LateralSum[2*jj];
        }
    }
    WallEdge2 = WallEdge1+530;
    return;
15   }

void AxialCorrection(unsigned AC1)
{
    MaxValCorr = 0.0;
    MaxCorrIndex = 0;
20   MakeHistogram(ShiftedImage,AC1);
    for(unsigned jj=0; jj<2*BigDimZ*BigDimX; jj++)
    {
        if (ShiftedImage [jj] >0)
25         ShiftedImage [jj] = MaxVal+MinVal- ShiftedImage[jj];
        if (TemplateImage[jj]>0)
            TemplateImage[jj] = MaxValOld+MinValOld- TemplateImage[jj];
    }
30   MultiDimFFT(TemplateImage-1, BigDims-1, 2, 1);
    MultiDimFFT(ShiftedImage-1, BigDims-1, 2, 1);
    MultiplyComplexValues(ShiftedImage, TemplateImage,
        TemplateImageNew, BigDimX, BigDimZ,
            BigDimX/10, BigDimZ/10);
    MultiDimFFT(TemplateImageNew-1, BigDims-1, 2, -1);
35   if(   MinDevIndexOld   >   (WallEdge1+WallEdge2)/2)   nnc   =   BigDimX   +
        (WallEdge1+WallEdge2)/2 - MinDevIndexOld;
    else
        nnc = (WallEdge1+WallEdge2)/2 - MinDevIndexOld;
    for(unsigned mm=0; mm<BigDimZ; mm++)
40   {
        {
            if( TemplateImageNew[2*nnc*BigDimZ+2*mm]
>
                MaxValCorr )
45   {
            MaxValCorr =
            TemplateImageNew[2*nnc*BigDimZ+2*mm];
            MaxCorrIndex = mm;
        }
50   }
    }
    return;
    }

55   void CrossCorrelation()
    {
        //include these two lines if no lateral correction is required
        // WallEdge1 = 55;
        // WallEdge2 = WallEdge1 + 530;
60   MinDevIndexOld = (WallEdge1+WallEdge2)/2;
        WallOld1 = WallEdge1;
        WallOld2 = WallEdge2;
        //remove these three lines if no lateralcorrection is required
        LateralCorrection(FirstSeg);
65   If (LateralMax1 < RedoLateralCorrection) LateralCorrection(SecondSeg);
        AxialCorrection(SecondSeg);
        MakeHistogram(TemplateImage,SecondSeg);
    }

```

## ES 2 559 411 T3

```

return;
}

void MakeFileName(char *MFN1, unsigned MFN2, unsigned MFN3)
5 {
  if (MFN3==0)
    sprintf (MFN1,
10 "%spp%2.2d%2.2d%2.2d%2.2d\\PP%2.2d%2.2d%2.2d%
    DirectoryName,FileYear, FileMonth, FileDay, FileSet, FileYear, FileMonth,
    FileDay,
    FileSet);
  if (MFN3==1)
15 sprintf (MFN1,
    "%spp%2.2d%2.2d%2.2d%2.2d\\PP%2.2d%2.2d%2.2d%
    2.2d%3.3dAVG.crw",
    DirectoryName,FileYear, FileMonth, FileDay, FileSet, FileYear, FileMonth,
    FileDay, FileSet, MFN2);
20 if (MFN3==2)
    sprintf (MFN1,
    "%spp%2.2d%2.2d%2.2d%2.2d\\PP%2.2d%2.2d%2.2d%
    2.2d%3.3dAVG.raw",
    DirectoryName,
25 FileYear, FileMonth,
    FileDay, FileSet, FileYear, FileMonth, FileDay, FileSet, MFN2);
    return;
  }

void WriteCorrectedImage(float WCIdata[])
30 {
  int XCenter = (WallEdge1+WallEdge2)/2;
  if(XCenter >= ImageDimX/2)
  {
35     if (CumulativeShift <= 0)
    {
      for(unsigned jj=0; jj<ImageDimX-XCenter+ImageDimX/2;
        jj++)
        {
40     for(int kk=ImageDimZ-1; kk>= -CumulativeShift; kk--)
        // for(unsigned kk= -CumulativeShift;
        kk<ImageDimZ;
        kk++)
          WCIdata[jj*ImageDimZ+kk]
45     = WCIdataC(jj+XCenter-
          ImageDimX/2)*ImageDimZ+kk+CumulativeShift];
      for(unsigned kk=0; kk<-CumulativeShift;
        kk++)
50     WCIdata[jj*ImageDimZ+kk] = MaxVal; // 0;
    }
    for(unsigned jj=(ImageDimX- XCenter+ImageDimX/2)*ImageDimZ;
      jj<ImageDimX*ImageDimZ; jj++)
      WCIdata[jj] = MaxVal;//0;
55 }
  if (CumulativeShift > 0)
  {
    for(unsigned jj=0; jj<ImageDimX-
      XCenter+ImageDimX/2; jj++)
60 {
      for(int kk=0; kk<ImageDimZ-CumulativeShift;
        kk++)
          WCIdata[jj*ImageDimZ+kk]
65     = WCIdata[(jj+XCenter-
          ImageDimX/2)*ImageDimZ+kk+CumulativeShift];

```

```

        for(unsigned kk=ImageDimZ-CumulativeShift;
            kk<ImageDimZ; kk++)
            WCIData[jj*ImageDimZ+kk] = MaxVal;//0;
5    }
        for(unsigned jj=(ImageDimX- XCenter+ImageDimX/2)*ImageDimZ;
            jj<ImageDimX*ImageDimZ; jj++)
            WCIData[jj] = MaxVal;//0;
    }
10   }
    if(XCenter < ImageDimX/2)
    {
        if(CumulativeShift <= 0)
        {
            for(int jj=ImageDimX-1;
15         jj>=(-XCenter+ImageDimX/2); jj-
                -)
                {
                    for(int kk=ImageDimZ-1; kk>= -CumulativeShift; kk--)
20                 // for(unsigned kk= -CumulativeShift;
                    kk<ImageDimZ;
                    kk++)
                        WCIData[jj*ImageDimZ+kk]
                            = WCIData[(jj+XCenter-
25
                    ImageDimX/2)*ImageDimZ+kk+CumulativeShift];
                    for(unsigned kk=0; kk<-CumulativeShift;
                    kk++)
                        WCIData[jj*ImageDimZ+kk] = MaxVal;//0;
30                }
                for(unsigned jj=0;
                jj<(-XCenter+ImageDimX/2)*ImageDimZ;
                jj++)
                    WCIData[jj] = MaxVal;//0;
35            }
            if(CumulativeShift > 0)
            {
                for(int jj=ImageDimX-1;
40             jj>=(-XCenter+ImageDimX/2); jj-
                    -)
                    {
                        for(unsigned kk=0; kk<ImageDimZ- CumulativeShift; kk++)
70                     // for(int kk=ImageDimZ-CumulativeShift-1;
                        kk>=0; kk--)
                            WCIData[jj*ImageDimZ+kk]
                                = WCIData[(jj+XCenter-
45
                                ImageDimX/2)*ImageDimZ+kk+CumulativeShift];
                            for(unsigned kk=ImageDimZ-CumulativeShi ft;
                            kk<ImageDimZ; kk++)
                                WCIData[jj*ImageDimZ+kk] = MaxVal;//0;
                            }
                            for(unsigned jj=0;
55             jj<(-XCenter+ImageDimX/2)*ImageDimZ;
                            jj++)
                                WCIData[jj] = MaxVal;//0;
                            }
60            }
            MakeFileName (ProcessedFilenameOut,CurrentPers
            pective,1);
            fstream ProcessedFile;
65            ProcessedFile.open(ProcessedFilenameOut,ios::
            binary|ios::out);
            for(unsigned jj=0; jj<ImageDimZ; jj++)

```

```

{
for(unsigned kk=0; kk<ImageDimX; kk++)
    ProcessedFile << WCIData[kk*ImageDimZ+jj]
5   << "\t";
}
ProcessedFile.close();
return;
}

10 void WriteToIndexFile()
{
MakeFileName(IndexFilenameOut,0,0);
fstream IndexFile;

15     IndexFile.open(IndexFilenameOut,ios::binary|i
os::app);
IndexFile << WallEdge1 << "\t" << LateralMax1 << "\t" <<
MaxCorrIndex << "\t"
20 << MaxValCorr << endl;
IndexFile.close();
return;
}

void CalculateOffsets()
25 {
MakeFileName(IndexFilenameOut,0,0);
fstream IndexFile;

IndexFile.open(IndexFilenameOut,ios::binary|i
30 os::out);
IndexFile.close();
BigDims[0] = BigDimX;
BigDims[1] = BigDimZ;
MakeOneLine();
35 for(CurrentPerspective=0;
CurrentPerspective<NumPerspectives;
CurrentPerspective++)
{

40 ReadUncorrectedImage(CurrentPerspective);
CopyRawImage(ShiftedImage);
CrossCorrelation();
if(MaxCorrIndex>BigDimZ/2)
CumulativeShift -= (BigDimZ-MaxCorrIndex);
45 else
CumulativeShift += MaxCorrIndex;
WriteCorrectedImage(RawImage);
WriteToIndexFile();
}
50 return;
}

void ReadOffsets()
{
55 double Junk;
MakeFileName(IndexFilenameOut,0,0);
fstream IndexFile;

IndexFile.open(IndexFilenameOut,ios::binary|i
60 os::in);
for(unsigned jj=0; jj<2*NumPerspectives; jj++)
{
IndexFile >> (unsigned) CorrectedFile[jj];
IndexFile >> Junk;
65 cout << CorrectedFile[jj] << "\n";
}
IndexFile.close();

```

```

for(CurrentPerspective=0;
CurrentPerspective<NumPerspectives;
CurrentPerspective++)
5 {
    ReadUncorrectedImage(CurrentPerspective);
    WallEdge1 =
CorrectedFile[2*CurrentPerspective];
    WallEdge2 = WallEdge1+530;
10    MaxCorrIndex =
CorrectedFile[2*CurrentPerspective+1];
    if(MaxCorrIndex>BigDimZ/2)
        CumulativeShift -= (BigDimZ- MaxCorrIndex);
    else
15        CumulativeShift += MaxCorrIndex;
    cout << CumulativeShift << endl;
    WriteCorrectedImage(RawImage);
    }
    return;
20 }

void main()
{
    int Calcoffs;
25 GetFileInfo();
    printf("Calculate offsets? (Y/N)\n");
    Calcoffs = getch ();
    if(Calcoffs == 'Y' || Calcoffs == 'y')
        CalculateOffsets();
30 if(Calcoffs == 'N' || Calcoffs == 'n')
        ReadOffsets ();
    cout << "Done!" << endl;
    wait
    }
35

```

La invención se ha descrito en el presente documento en considerable detalle para cumplir los estatutos de patente y para proporcionar a los expertos en la materia la información necesaria para aplicar los principios nuevos de la presente invención, y para construir y usar tales componentes ejemplares y especializados según sea necesario.

## REIVINDICACIONES

1. Un procedimiento de corrección del movimiento relativo de un detector de objetos entre vistas sucesivas en un sistema de microscopía tomográfica de proyección óptica que incluye un tubo microcapilar (3) que tiene un eje del tubo, en el que las vistas sucesivas comprenden una vista actual y una vista anterior, en el que la vista actual y la vista anterior se toman cada una desde una perspectiva diferente, comprendiendo el procedimiento las etapas de:
- 5 iluminar un objeto (1) de interés contenido dentro del tubo microcapilar para producir imágenes adquiridas que incluyen una imagen actual (111) a partir de la vista actual y una imagen anterior a partir de la vista anterior; determinar un valor de corrección de desviación lateral para cada una de las imágenes (114) actuales y anteriores, donde la desviación lateral es una traslación a lo largo de la dimensión de la imagen perpendicular al eje del tubo, formando umbrales de cada imagen (21) y realizando correlaciones cruzadas de cada imagen con formación de umbral con una imagen (22, 23, 24, 25, 26, 27) de plantilla a lo largo de las filas de cada imagen con formación de umbral y cada imagen de plantilla, en el que la imagen de plantilla se crea (111) creando dos líneas brillantes separadas por una cantidad conocida encontrada entre las paredes del tubo microcapilar en una de las imágenes adquiridas para formar una imagen de plantilla preliminar, expandir la imagen de plantilla preliminar para proporcionar relleno con ceros en dos dimensiones, lo que tiene como resultado una imagen de plantilla expandida, y realizar una FFT (22) en dos dimensiones en la imagen de plantilla expandida para crear la imagen de plantilla final;
- 10 determinar un valor de corrección de desviación axial para la imagen actual (115) donde la desviación axial es una traslación a lo largo de la dimensión de la imagen en paralelo al eje del tubo, realizando una formación de umbral de una copia de la imagen actual para producir una versión con formación de umbral de la imagen actual, realizando una correlación cruzada de la versión con formación de umbral de la imagen actual con una versión con formación de umbral de la imagen anterior y determinar una desviación axial como un máximo en la función de correlación cruzada a lo largo de la línea que se corresponde con la diferencia en los valores de corrección de desviación lateral de las imágenes actuales y anteriores; y aplicar el valor de corrección de desviación lateral y el valor de corrección de desviación axial para la imagen actual en la imagen actual para producir una imagen (116) de archivo corregida.
- 15
- 20
- 25
2. El procedimiento de la reivindicación 1, en el que el tubo microcapilar (3) tiene una pared de tubo superior y en el que la etapa de realizar una formación de umbral de la imagen (21) comprende además las etapas de:
- 30 encontrar el histograma de escala de grises de la imagen que incluye una pluralidad de barras; identificar una barra con el mayor número de píxeles; configurar todos los píxeles en la imagen con el valor de escala de grises de esa barra o mayor igual a cero en una copia de la imagen;
- 35 aplicar una FFT (22) en dos dimensiones en la copia de la imagen para producir una transformada de Fourier; multiplicar la transformada de Fourier por un conjugado complejo de la transformada de Fourier de la imagen de plantilla para producir una nueva matriz (23) de imagen; sumar la nueva matriz (24) de imagen, a lo largo de cada una de la pluralidad de filas para computar una matriz de suma lateral;
- 40 computar una transformada de Fourier de una dimensión de la matriz (25) de suma lateral para encontrar la correlación cruzada de las filas de la copia de la imagen original y la imagen de plantilla; establecer una posición sin corregir de la pared de tubo superior, en la ubicación del valor máximo de la matriz (26) de suma lateral; y determinar la desviación lateral como la diferencia entre la posición sin corregir de la pared de tubo superior y una posición (28) predeterminada de borde de pared.
- 45
3. El procedimiento de la reivindicación 1, en el que la etapa de determinar una desviación axial comprende además las etapas de:
- encontrar el histograma de escala de grises de la imagen actual que incluye una pluralidad de barras; identificar una barra con el mayor número de píxeles; configurar todos los píxeles en la imagen original con el valor de escala de grises de esa barra o mayor igual a cero en una copia de la imagen original (31); aplicar un filtro de paso bajo a la imagen (33) con formación de umbral; computar una correlación cruzada de la imagen con formación de umbral y con filtro de paso bajo con una versión (35, 36) de la imagen anterior con formación de umbral y con filtro de paso bajo; encontrar el valor de corrección máximo en la fila de la correlación cruzada resultante que se corresponde con la diferencia en las desviaciones laterales (37) de las dos imágenes; y añadir el valor de corrección a la suma de todas las desviaciones axiales anteriores.
- 50
- 55
4. El procedimiento de la reivindicación 3, que comprende además las etapas de:
- escribir la posición de la pared de tubo superior en un archivo; escribir el valor máximo de la correlación cruzada entre la imagen actual y la imagen de plantilla en el archivo;

escribir en el archivo la ubicación del valor de corrección máximo en la fila apropiada de la correlación cruzada, donde la fila apropiada es la fila que se corresponde con la diferencia en las desviaciones laterales de las dos imágenes:

escribir el valor de corrección máximo en el archivo; y

5 generar una imagen corregida recortando el número apropiado de píxeles desde uno o dos bordes y desplazando los píxeles restantes mediante el número recortado.

5. El procedimiento de la reivindicación 1, en el que el objeto (1) de interés es una célula o el núcleo de una célula.

10 6. El procedimiento de la reivindicación 4, que comprende además la etapa de repetir las etapas con un número diferente de barras en el histograma, si el valor máximo de la correlación cruzada tiene una magnitud menor que un valor (28) predeterminado.

15 7. El procedimiento de la reivindicación 1, en el que una pluralidad de imágenes se producen para generar un archivo de imagen de entrada, y una pluralidad de imágenes de archivo corregidas son una copia recortada de la imagen de archivo de entrada, con porciones sin recortar desplazadas verticalmente y/u horizontalmente, y con píxeles vacíos adicionales insertados en uno o dos de los límites para retener el tamaño de la imagen de archivo de entrada.

8. El procedimiento de la reivindicación 1, en el que las imágenes de archivo corregidas comprenden valores de desviación calculados y guardados en un archivo digital.

20 9. El procedimiento de la reivindicación 8, en el que los valores de desviación calculados se procesan usando procesamiento de texto por ordenador para producir un archivo de texto alterado, y el archivo de texto alterado se usa para generar desviaciones en dos ejes, y en el que la etapa de determinar un valor de corrección de desviación lateral para la imagen se repite para encontrar un valor máximo.

10. El procedimiento de la reivindicación 9, en el que los valores de desviación calculados comprenden:

una posición de la pared de tubo superior;

un valor máximo de la correlación cruzada entre la imagen actual y la imagen de plantilla;

25 una ubicación del valor de correlación máxima en la fila correspondiente de una correlación cruzada entre la imagen actual y la imagen anterior; y  
un valor de correlación máxima.

11. Un aparato configurado para realizar el procedimiento de cualquier reivindicación anterior.

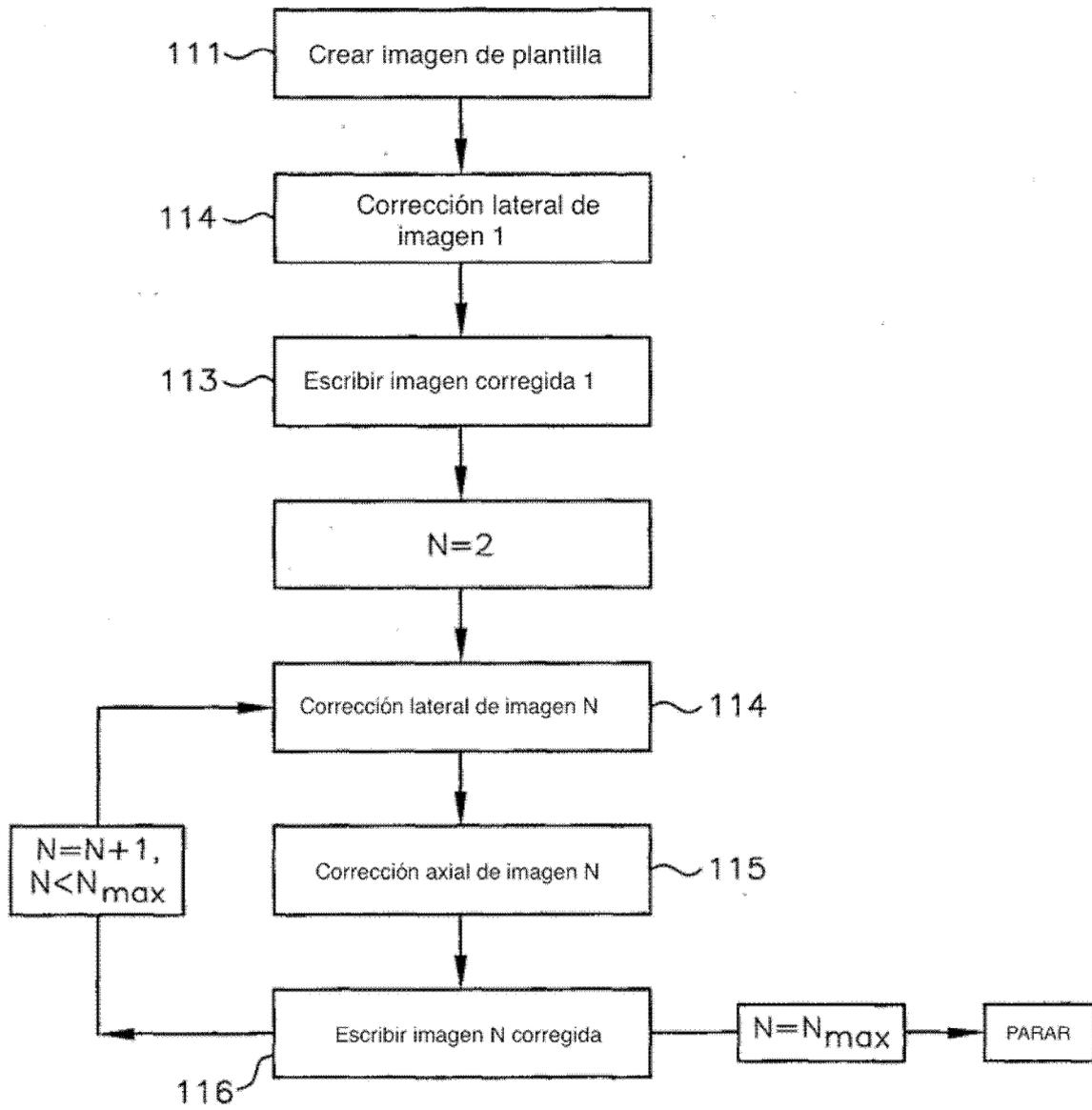


FIG. 1

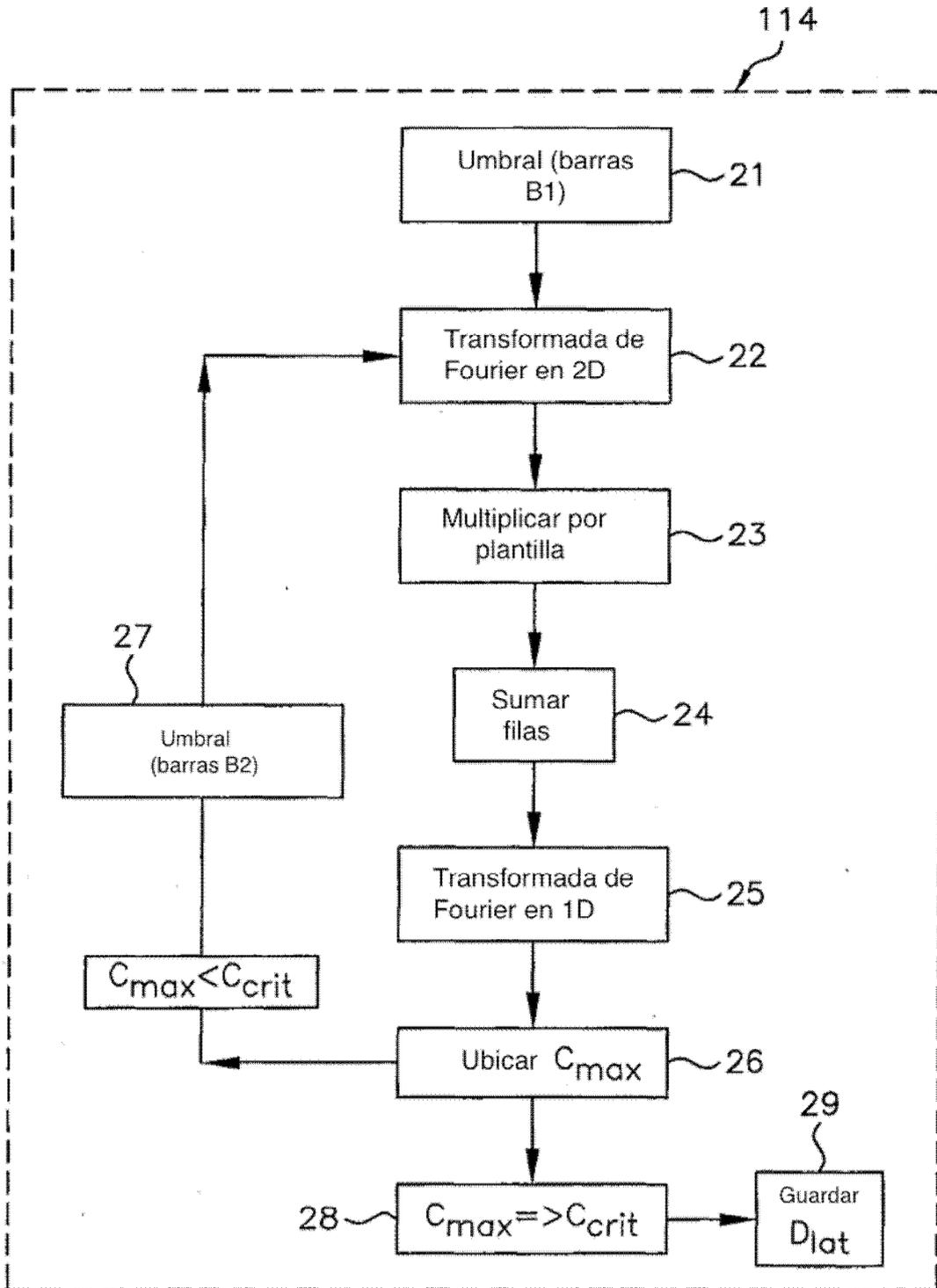
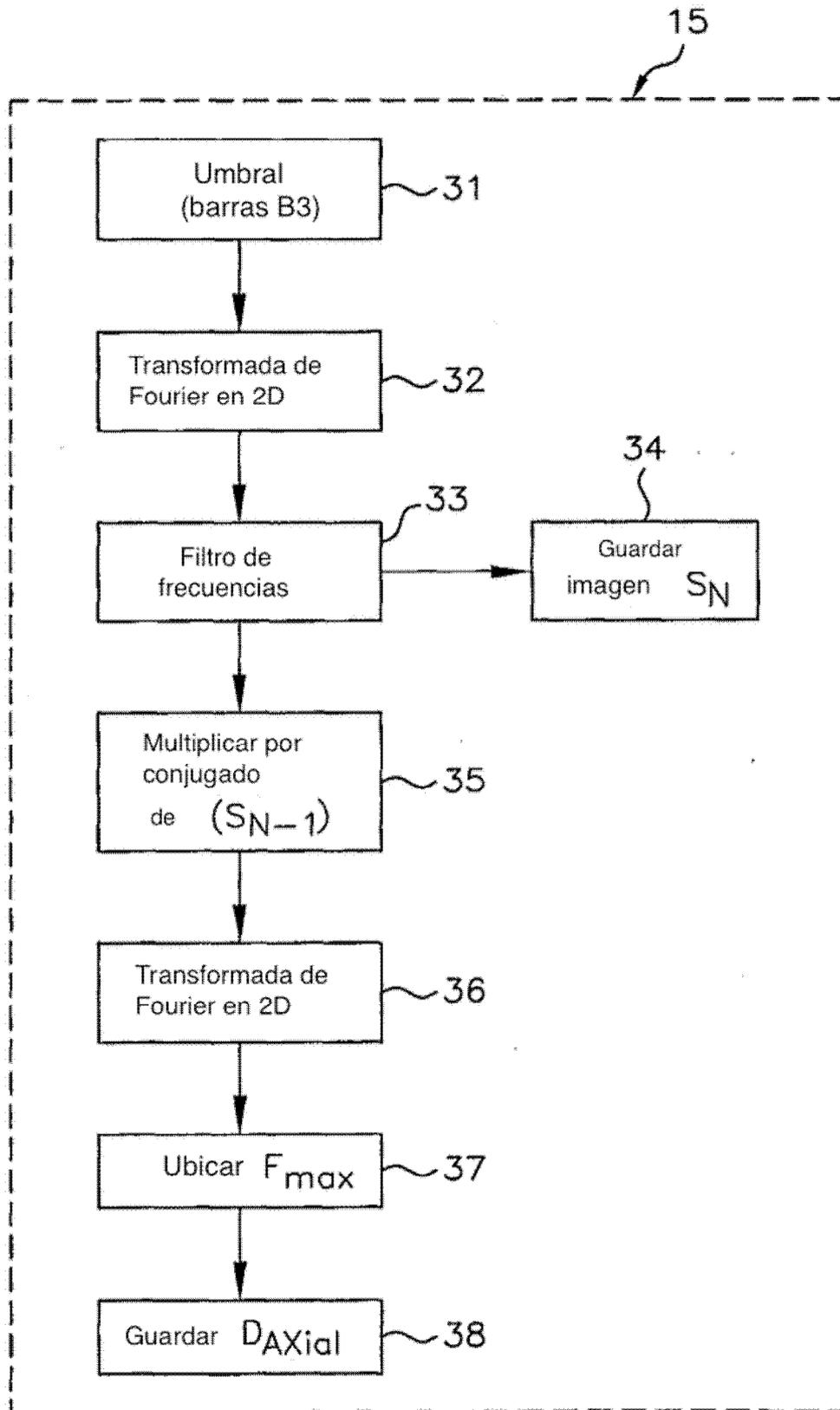


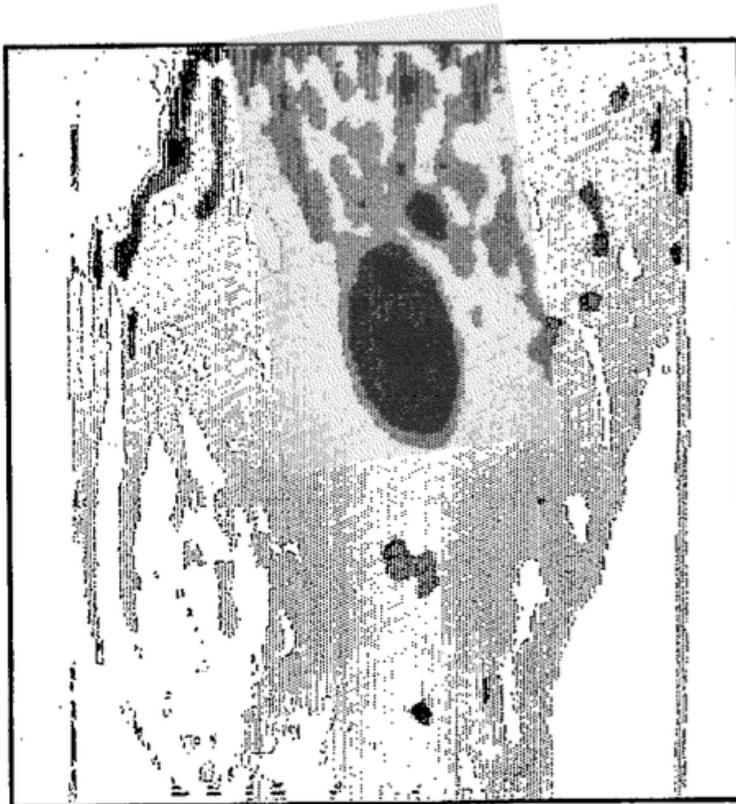
FIG. 2



**FIG. 3**



**FIG. 4B**



**FIG. 4A**

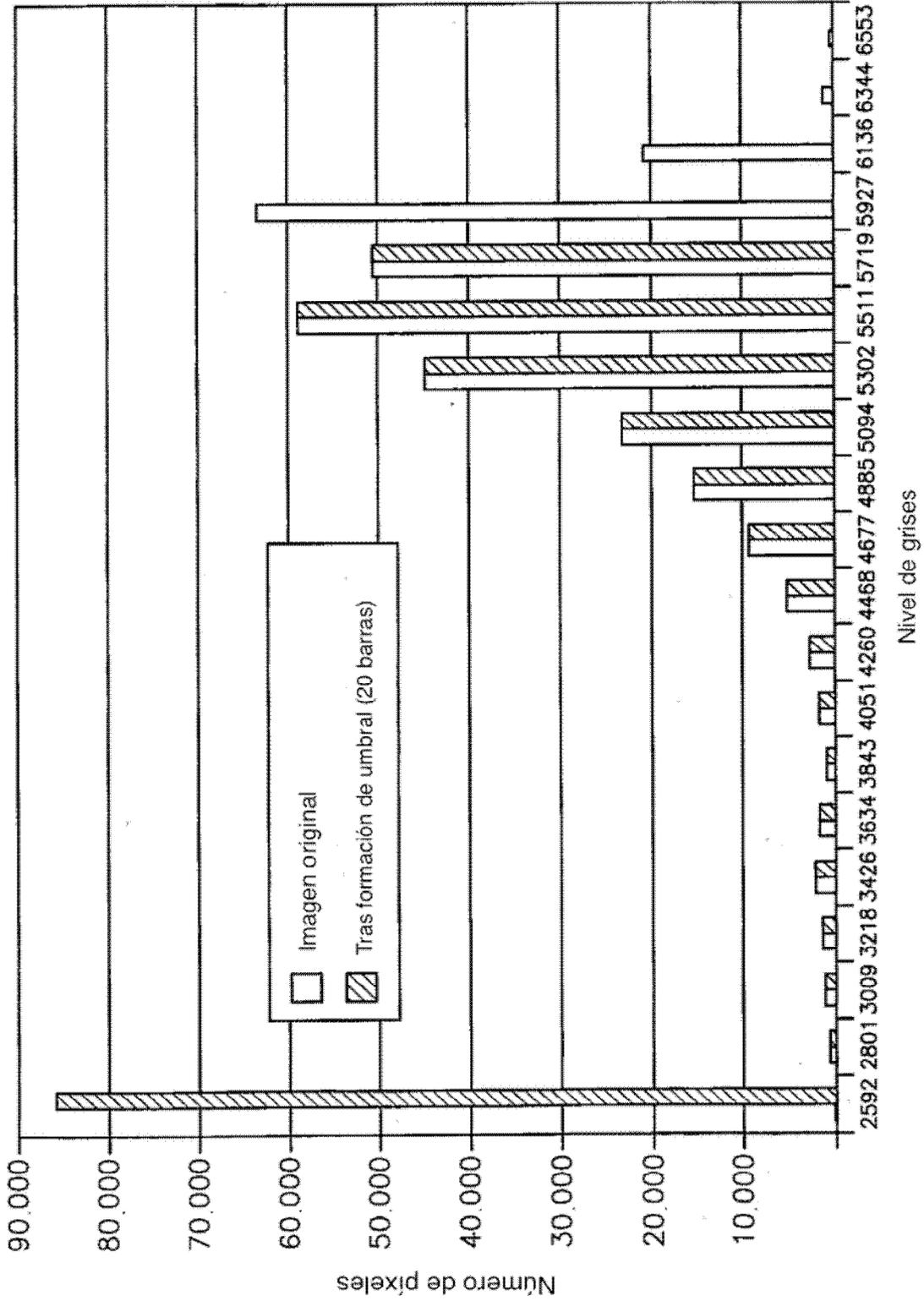
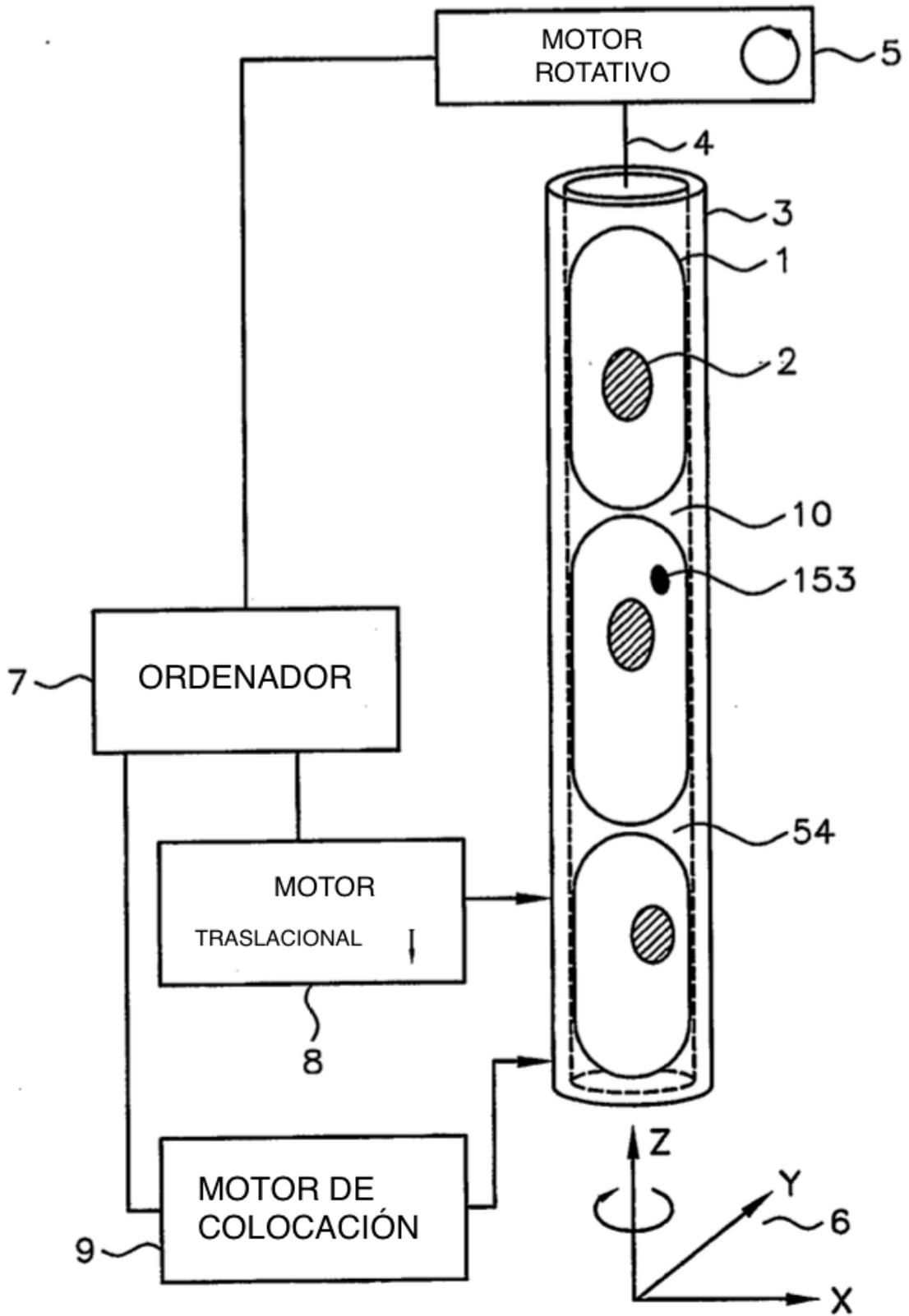
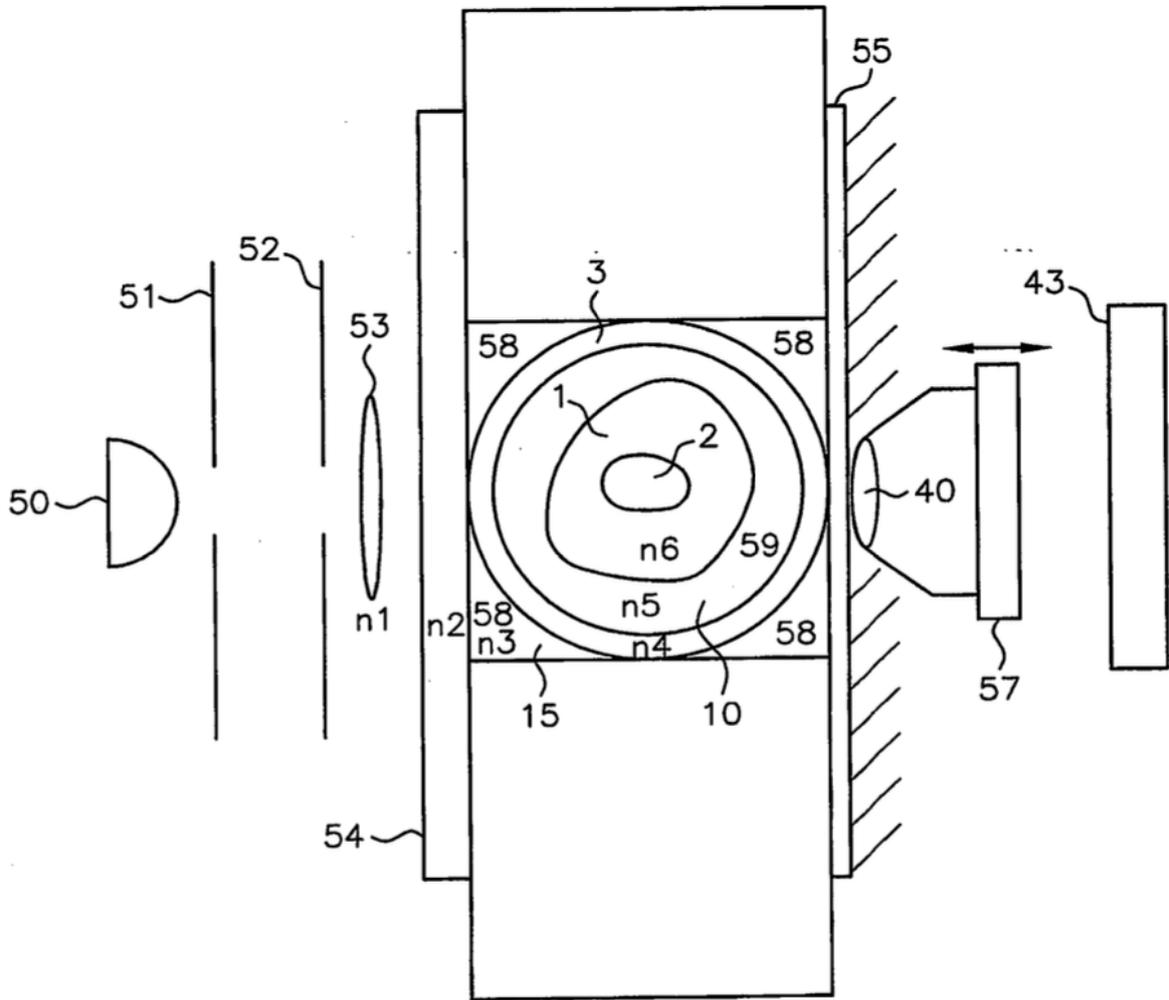


FIG. 4C



**FIG. 5**



**FIG. 6A**

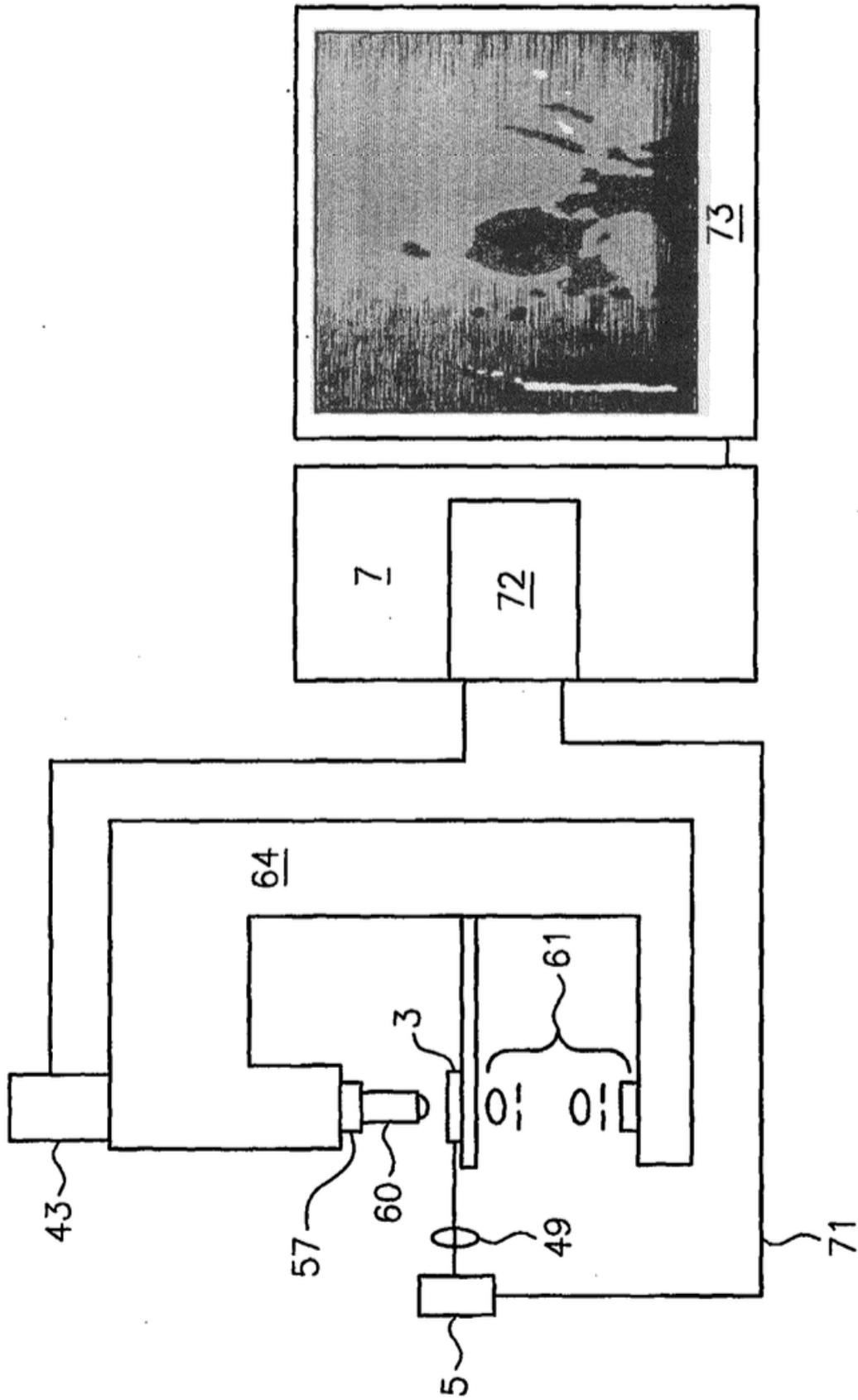
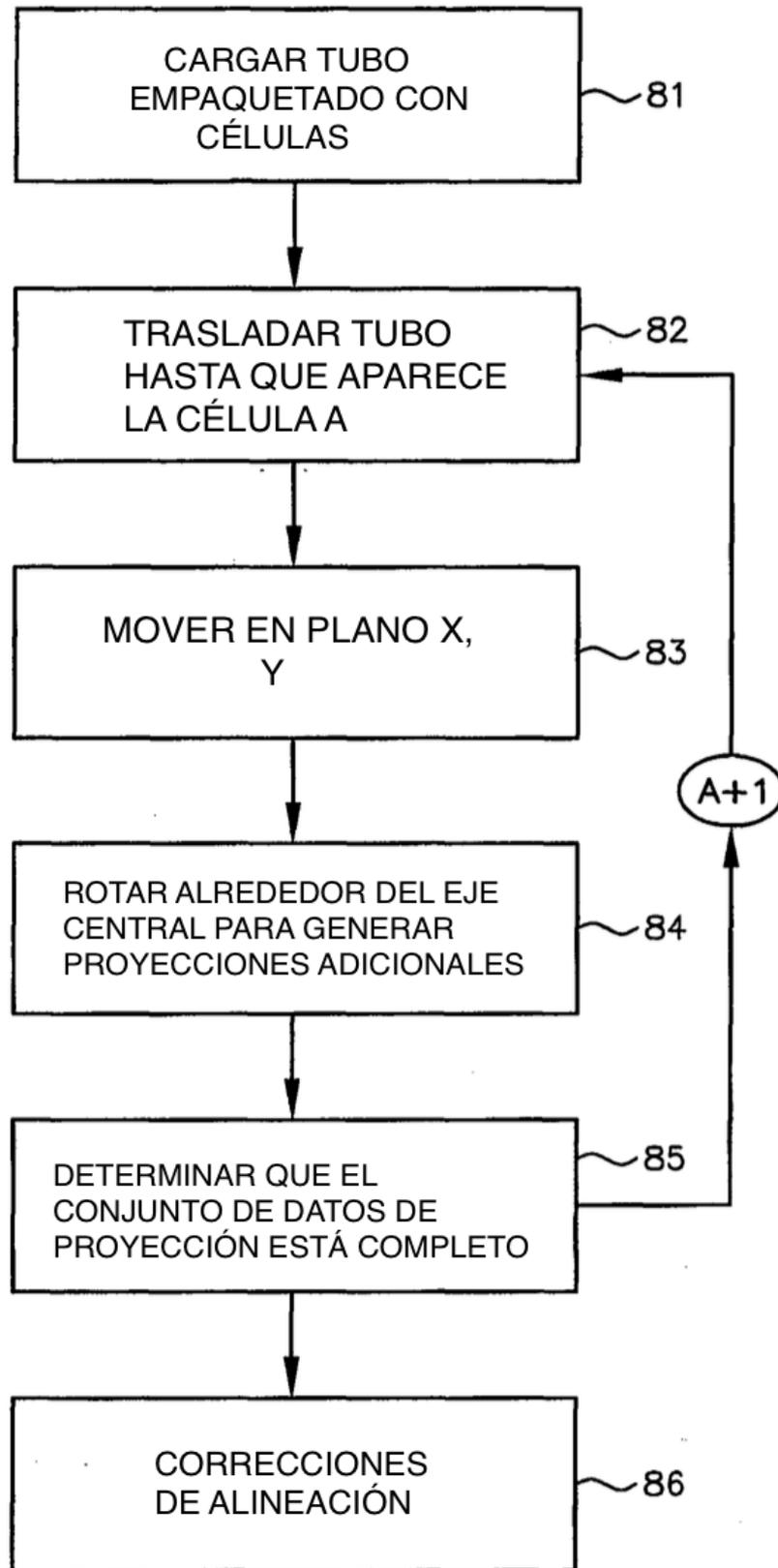
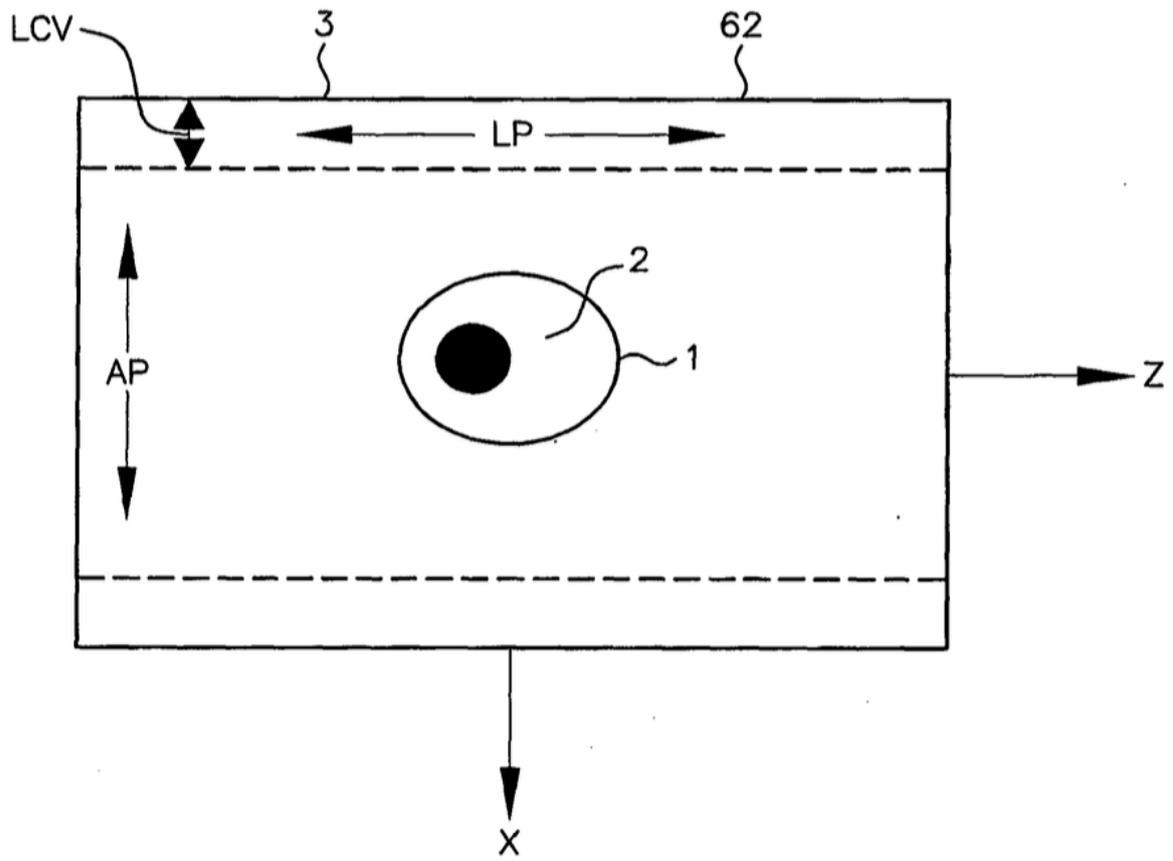


FIG. 6B



**FIG. 7**



**FIG. 8**