

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 399 636**

51 Int. Cl.:

G06T 7/00

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **29.07.2009 E 09777521 (7)**

97 Fecha y número de publicación de la concesión europea: **12.12.2012 EP 2382599**

54 Título: **Método para determinar la postura de una cámara con respecto a por lo menos un objeto real**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
02.04.2013

73 Titular/es:

**METAIO GMBH (100.0%)
Infanteriestrasse 19 Haus 3, 2. OG
80797 München, DE**

72 Inventor/es:

BEN HIMANE, MOHAMED SELIM

74 Agente/Representante:

UNGRÍA LÓPEZ, Javier

ES 2 399 636 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

Método para determinar la postura de una cámara con respecto a por lo menos un objeto real

5 La presente invención se refiere a un método para determinar la postura de una cámara con respecto a por lo menos un objeto real. Una cámara se acciona para capturar una imagen bidimensional que incluye por lo menos una parte de un objeto real y, en el proceso de determinación, la postura de la cámara con respecto al objeto real se determina usando correspondencias entre puntos tridimensionales asociados con el objeto real y unos puntos bidimensionales correspondientes del objeto real en la imagen bidimensional.

10 **Antecedentes de la invención**

15 Los sistemas de realidad aumentada permiten la superposición de información virtual generada por ordenador con impresiones visuales del entorno real. Con este fin, las impresiones visuales del mundo real se mezclan con información virtual, por ejemplo por medio de vidrios de datos semi-transmisores o por medio de un visualizador montado en la cabeza que se lleva en la cabeza de un usuario. La combinación de objetos o información virtual puede efectuarse de una forma dependiente del contexto, es decir, correspondiente con y deducida del entorno respectivo observado. Como información virtual, es posible usar básicamente cualquier tipo de datos, tal como textos, imágenes etc. El entorno real se detecta por ejemplo con la ayuda de una cámara portada en la cabeza del usuario.

20 Cuando la persona que usa un sistema de realidad aumentada gira su cabeza, el seguimiento de los objetos virtuales es necesario con respecto al campo de visión cambiante. El entorno real puede ser un aparato complejo, y el objeto detectado puede ser una parte significativa del aparato. Durante un así denominado proceso de seguimiento, el objeto real (que puede ser un objeto que va a observarse tal como un aparato, un objeto dotado de un marcador que va a observarse, o un marcador colocado en el mundo real para fines de seguimiento) detectado durante la inicialización sirve como referencia para calcular la posición en la que la información virtual ha de visualizarse o combinarse en una imagen o fotografía tomada por la cámara. Para este fin, es necesario determinar la postura de la cámara con respecto al objeto real. Debido al hecho de que el usuario (y, en consecuencia, la cámara cuando esta se porta por el usuario) puede cambiar su posición y orientación, el objeto real ha de someterse a un seguimiento continuo con el fin de visualizar la información virtual en la posición correcta en el dispositivo visualizador también en el caso de una postura alterada (posición y orientación) de la cámara. El efecto conseguido de ese modo es que la información, con independencia de la posición y/o orientación del usuario, se visualiza en el dispositivo visualizador de una forma correcta con el contexto con respecto a la realidad.

35 Uno de los problemas en el campo de la realidad aumentada es la determinación de la posición de la cabeza y la orientación de la cabeza del usuario por medio de una cámara que está asociada de algún modo con la cabeza del usuario. Otro problema puede ser determinar la posición y orientación de la cámara en el interior de un teléfono móvil con el fin de superponer información sobre la imagen de cámara y de mostrar la combinación de ambas en el visualizador del teléfono. Con este fin, en algunas aplicaciones la postura de la cámara con respecto a por lo menos un objeto real del entorno real capturado se estima usando el flujo de vídeo o el flujo de imágenes de la cámara como fuente de información.

40 La estimación de postura es una de las tareas más básicas y más importantes en la visión por ordenador y en la realidad aumentada. En la mayoría de las aplicaciones en tiempo real, es necesario resolver esta en tiempo real. No obstante, debido a que la misma implica un problema de minimización no lineal, requiere una gran cantidad de tiempo de cálculo.

50 **Sumario de la invención**

Por lo tanto, un objeto de la presente invención es la provisión de un método para determinar la postura de una cámara con respecto a por lo menos un objeto real que puede calcularse mediante un medio de procesamiento de datos con un tiempo de cálculo bastante bajo.

55 La invención se dirige a un método para determinar la postura de una cámara con respecto a por lo menos un objeto real de acuerdo con las características de la reivindicación 1. Además, la invención se dirige a un producto de programa informático tal como se reivindica en la reivindicación 12.

60 De acuerdo con un aspecto, la invención se refiere a un método para determinar la postura de una cámara con respecto a por lo menos un objeto real, comprendiendo el método las etapas de:

- accionar la cámara para capturar una imagen bidimensional (o tridimensional) que incluye por lo menos una parte de un objeto real,
- proporcionar correspondencias entre puntos tridimensionales asociados con el objeto real y unos puntos bidimensionales (o tridimensionales) del objeto real en la imagen bidimensional (o tridimensional, respectivamente); por ejemplo, las coordenadas de los puntos en 3D pueden medirse sobre el objeto real o

- provenir del dibujo técnico en 2D del objeto real o se determinan gracias a un modelo de CAD en 3D del objeto, mientras que las coordenadas puntuales en 2D puede determinarse en general gracias a algunas técnicas de procesamiento de imágenes convencionales tal como la extracción de puntos característicos (puntos de esquina, puntos de arista, puntos invariantes de escala, etc.),
- 5 - determinar una estimación inicial de la matriz de transformación (que se usará posteriormente para determinar la postura) como base inicial para un proceso de minimización iterativa usado para depurar de forma iterativa la matriz de transformación,
- determinar una matriz jacobiana que incluye información con respecto a la estimación inicial de la matriz de transformación y valores de referencia de las coordenadas puntuales tridimensionales asociadas con el objeto real; por ejemplo, la matriz jacobiana puede estimarse por medios numéricos usando, por ejemplo, el método de diferencias finitas o por medios analíticos suponiendo un cierto modelo de proyección de cámara (en el presente caso se determina la misma por medios analíticos suponiendo el modelo de cámara estenopeica),
- 10 - en el proceso de minimización iterativa, determinar en cada uno de múltiples bucles de iteración una versión actualizada respectiva de la matriz de transformación basándose en una versión previa respectiva de la matriz de transformación y basándose en la matriz jacobiana, en el que la matriz jacobiana no se actualiza durante el proceso de minimización iterativa, y
- 15 - determinar la postura de la cámara con respecto al objeto real usando la matriz de transformación determinada al final del proceso de minimización iterativa.
- 20 En una realización de la invención, el proceso de minimización iterativa implica el algoritmo de Gauss-Newton o el algoritmo de Levenberg-Marquardt.
- En una realización adicional de la invención, el proceso de minimización iterativa implica un proceso de actualización compuesta en el que la versión actualizada respectiva de la matriz de transformación se calcula a partir de una multiplicación entre una matriz construida con parámetros de actualización de la matriz de transformación previa respectiva y la matriz de transformación previa respectiva.
- 25 En una realización adicional, se usa una parametrización de álgebra de Lie en el proceso de minimización iterativa.
- 30 Realizaciones adicionales de la invención se exponen en las reivindicaciones dependientes.

Breve descripción de los dibujos

- Algunos aspectos de la invención se analizarán con más detalle en lo siguiente por medio de las figuras que ilustran realizaciones de la invención, en las que
- 35 la figura 1 muestra una visualización esquemática de una escena que comprende un objeto real observado por una cámara y una imagen de cámara que visualiza una representación bidimensional del objeto real,
- 40 la figura 2 muestra un proceso de acuerdo con el algoritmo de Gauss-Newton adaptado para determinar una matriz de transformación T que es apropiada para la determinación de la postura de la cámara de acuerdo con la figura 1,
- 45 la figura 3 muestra una posible implementación de cálculo del proceso de acuerdo con la figura 2,
- la figura 4 muestra un proceso de acuerdo con el algoritmo de Levenberg-Marquardt adaptado para determinar una matriz de transformación T que es apropiada para la determinación de la postura de la cámara de acuerdo con la figura 1,
- 50 la figura 5 muestra una posible implementación de cálculo del proceso de acuerdo con la figura 4,
- la figura 6 muestra una posible implementación del algoritmo de Gauss-Newton para una estimación de postura no lineal en la que la matriz jacobiana se calcula previamente y la actualización se realiza de una forma compuesta,
- 55 la figura 7 muestra una posible implementación adicional del algoritmo de Gauss-Newton para una estimación de postura no lineal en la que la matriz inversa generalizada se calcula previamente asimismo,
- 60 la figura 8 muestra una posible implementación del algoritmo de Levenberg-Marquardt para una estimación de postura no lineal en la que la matriz jacobiana J y la matriz $J^T J$ se calculan previamente y la actualización se realiza de una forma compuesta,
- 65 la figura 9 muestra una posible implementación adicional del algoritmo de Levenberg-Marquardt para una estimación de postura no lineal en la que la jacobiana J y la matriz $J^T J$ se calculan previamente y la actualización se calcula usando la inversa generalizada,

la figura 10 muestra una posible implementación adicional del algoritmo de Levenberg-Marquardt para una estimación de postura no lineal en la que se han realizado algunos cálculos previos más,

5 la figura 11 muestra un conjunto de ecuaciones $m_i \times L \text{ Pi}^* = 0$, tal como se hace referencia al mismo con más detalle a continuación.

Descripción detallada de realizaciones

10 La figura 1 muestra una visualización esquemática de una escena que comprende cualquier tipo de un objeto real 3 (en el presente ejemplo, en forma de cubo) que comprende unos puntos en 3D Pi^* que se observan por una cámara 1. La cámara 1 está acoplada con un medio de procesamiento de datos 2, que puede ser cualquier tipo de procesador de cálculo que sea apropiado para procesar las imágenes digitales proporcionadas por la cámara 1. Por ejemplo, la cámara 1 y el procesador 2 pueden estar incluidos en el mismo dispositivo, tal como un teléfono móvil o cualquier otro dispositivo móvil, o pueden estar incorporados en unos dispositivos distribuidos respectivos acoplados por cable o de forma inalámbrica. La cámara genera una o más imágenes de cámara 4 que visualizan una representación bidimensional 5 del objeto real 3 que tiene unos puntos en 2D p_i . La representación del objeto en la imagen no está restringida a la apariencia del objeto como una respuesta fotométrica a la luz del espectro visible reflejado a partir del objeto (que se corresponde con la imagen convencional que adquieren las cámaras comerciales convencionales). La representación del objeto puede ser también su respuesta a otras porciones de la luz electromagnética. Esta puede también ser por ejemplo una imagen obtenida con una cámara infrarroja, una cámara térmica o incluso una cámara de Tiempo De Vuelo (Time Of Flight, TOF). En el caso de la cámara de Tiempo De Vuelo, la profundidad de los puntos de imagen puede medirse directamente: estas cámaras emiten una luz infrarroja modulada y entregan una imagen de amplitud (que es la amplitud de la señal reflejada en cada píxel) y una imagen de distancia (que es el desplazamiento de fase entre la señal emitida y la reflejada). Otro sistema de cámaras puede combinarse 2 cámaras para adquirir una imagen que incluye información de profundidad en 3D. También pueden imaginarse otros sistemas, que proporcionan información de imágenes combinada con 3D o datos de profundidad (por ejemplo, una combinación de una cámara TOF y una cámara a color o una cámara a color con una unidad de exploración láser).

30 De esta forma, una cámara apropiada puede capturar una imagen tridimensional y puede proporcionarse una matriz de transformación que incluye información con respecto a una correspondencia entre puntos tridimensionales asociados con el objeto real y unos puntos tridimensionales correspondientes del objeto real tal como se incluyen en la imagen tridimensional.

35 En un proceso de seguimiento, el objeto es determinar la postura de la cámara 1 con respecto al objeto real 3, calculándose la postura usando la matriz de transformación T entre el sistema de coordenadas 11 de la cámara y el sistema de coordenadas 13 del mundo asociado con el objeto real 3. La información obtenida de este modo sirve como referencia para calcular la posición en la que la información virtual ha de visualizarse o combinarse en una imagen 4 tomada por la cámara. Un seguimiento continuo del objeto real 3 se realiza cuando cambia la posición y/o orientación relativas entre el usuario o la cámara 1 y el objeto real 3.

45 Dadas las coordenadas del conjunto de puntos en 3D Pi^* (expresadas en el sistema de coordenadas 13 del mundo, en aras de simplicidad, en la figura 1 el sistema de coordenadas del mundo se elige superpuesta con el sistema de coordenadas del objeto real) y las coordenadas de sus proyecciones en 2D p_i en la imagen (expresadas en el sistema de coordenadas 14 de la imagen), el objetivo es determinar la postura de cámara en el sistema de coordenadas 13 del mundo: la posición y la orientación de la cámara 1 en el sistema de coordenadas 13 del mundo. (Observación: con una cámara de Tiempo De Vuelo o una cámara estereoscópica, es posible conseguir la profundidad de las proyecciones en 2D, teniéndose entonces acceso a las coordenadas en 3D completas de los puntos considerados en el sistema de coordenadas unido a la cámara).

50 Las coordenadas de los puntos en 3D Pi^* puede, por ejemplo, medirse manualmente en el objeto real o provenir del dibujo técnico en 2D del objeto real o determinarse gracias un modelo de CAD en 3D del objeto. Mientras que las coordenadas puntuales en 2D p_i se determinan en general gracias a algunas técnicas de procesamiento de imágenes convencionales tal como la extracción de características (puntos de esquina, puntos de arista, puntos invariantes de escala, etc.). Las correspondencias entre los Pi^* y p_i se obtienen gracias a un método de visión por ordenador convencional en general basándose en las propiedades geométricas de los puntos en 3D y el uso de una distribución espacial geométrica especial de los puntos y sus proximidades. Esto puede también basarse en algún conocimiento de la apariencia de los puntos en 3D en la imagen basándose en un conjunto de imágenes etiquetado manualmente denominado fotogramas clave (teniéndose entonces una apariencia posible diferente del objeto y encontrándose los puntos en 3D sobre la misma en imágenes adquiridas desde diferentes puntos de vista). En ese caso, los enfoques que usan, por ejemplo, descriptores (por ejemplo, David G. Lowe: "Distinctive Image Features from Scale-Invariant Keypoints In: International Journal of Computer Vision", Vol. 60, n.º 2, páginas 91-110, 2004) o clasificadores (por ejemplo, V. Lepetit y P. Fua, Keypoint Recognition using Randomized Trees. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, n.º 9, páginas 1465-1479, 2006) podrían usarse para establecer las correspondencias. Por ejemplo, las correspondencias entre los puntos en 3D y los puntos en 2D se establecen entre el objeto real y los fotogramas clave (2D↔3D). Y con el fin de establecer las correspondencias

entre el objeto real y la imagen en cuestión (la imagen 4), sólo se necesitan entonces unas correspondencias 2D↔2D. En el caso de que se encuentre disponible un modelo de CAD en 3D con textura del objeto, este puede usarse durante el proceso de determinación de las correspondencias por ejemplo mediante la generación sintética de las imágenes de fotogramas clave en lugar de adquirir las mismas con una cámara.

5 La posición de la cámara se denomina también traslación de la cámara y puede representarse usando un vector t (3x1) tal como conoce el experto. La orientación de la cámara se denomina también rotación de la cámara y puede representarse con una matriz de rotación R (3x3) tal como también conoce el experto. La matriz de rotación R es una matriz que pertenece al así denominado grupo Ortogonal Especial $SO(3)$ que es conocida por el experto en la materia. El grupo $SO(3)$ es el grupo de matrices (3x3) que verifica: $R^T R = Id$ y $\det(R) = 1$, Id es la matriz identidad (que es una matriz (3x3) con unos en la diagonal principal y ceros en todas las otras posiciones y \det representa el determinante de la matriz).

10 La postura de cámara en el sistema de coordenadas 13 del mundo puede escribirse usando una matriz (4x4)

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

15 Esta matriz T se denomina la matriz de transformación y está encuadrada en el grupo Euclidiano Especial $SE(3)$. Esta tiene 6 grados de libertad: 3 grados de libertad para la traslación y 3 grados de libertad para la rotación.

20 Lo anterior se conoce básicamente por el experto y se describe en, por ejemplo, "Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision", Cambridge University press, segunda edición, 2003 (véase, por ejemplo, el capítulo 3, páginas 65-86).

25 La proyección de la imagen puede resumirse tal como sigue:

30 A una cierta postura T de la cámara, un punto en 3D $P_i^* = [X_i^*, Y_i^*, Z_i^*, 1]$ expresado en el sistema de coordenadas 13 del mundo (en el que X_i^* es la coordenada a lo largo del eje x , Y_i^* es la coordenada a lo largo del eje y y Z_i^* es la coordenada a lo largo del eje z) se proyecta en primer lugar sobre el plano normalizado como un punto "normalizado" $m_i = [x_i, y_i, 1] = [X_i/Z_i, Y_i/Z_i, 1]$ en el que $[X_i, Y_i, Z_i, 1] = T[X_i^*, Y_i^*, Z_i^*, 1]$, y en el que x_i, y_i son las coordenadas del punto en 2D en el plano normalizado. El plano normalizado es el plano ortogonal al eje z del sistema de coordenadas de la cámara (que se corresponde con el eje óptico de la cámara) y la distancia entre este plano y el centro óptico de la cámara es una distancia de 1 unidad. Esto es sólo un concepto matemático que permite la expresión de las coordenadas puntuales como una proyección simple de los puntos en 3D sin la necesidad de los parámetros intrínsecos de la cámara (véase a continuación). (Observación: con una cámara de Tiempo De Vuelo o una cámara estereoscópica, es posible conseguir las coordenadas en 3D de los puntos transformados en el sistema de coordenadas unido a la cámara, es decir, es posible medir directamente $[X_i, Y_i, Z_i, 1]$).

35 A continuación, m_i se proyecta en la imagen de cámara como $p_i = [u_i, v_i, 1] = K m_i$ en el que la matriz K es la matriz de parámetros intrínsecos de la cámara (3x3)

$$K = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

45 Los parámetros f_x y f_y son respectivamente la longitud focal horizontal sobre el eje x y la longitud focal vertical sobre el eje y , s es la inclinación (Este parámetro es diferente de 0 en el caso de que los elementos de pixel de la disposición de sensores de la cámara no tenga unos ejes perpendiculares), u_0 y v_0 son las coordenadas del punto principal en píxeles en la imagen (el punto principal es la intersección del eje óptico con el plano de la imagen). Para más detalles, véase "Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision", Cambridge University press, segunda edición, 2003 (véase, por ejemplo, el capítulo 6, páginas 153-177).

Se da también el fenómeno de la distorsión de la cámara. Pero en el presente caso se considerará que la cámara está calibrada y que los parámetros intrínsecos son conocidos. Las imágenes también se consideran sin distorsión.

55 Por lo tanto, cuando se tiene un punto de imagen $p_i = [u_i, v_i, 1]$ con las coordenadas expresadas en una imagen sin distorsión, es posible conseguir su punto "normalizado" correspondiente m_i tal como sigue: $m_i = K^{-1} p_i$. Una ilustración de la proyección puede verse en la figura 1.

En adelante, en lugar de considerar unos puntos correspondientes P_i^* p_i , se considerarán unos puntos correspondientes: $P_i^* \leftrightarrow m_i$ debido a que tener p_i permite conseguir directamente m_i usando la fórmula anterior.

5 Tener un conjunto de puntos en 3D-2D correspondientes, la postura de la cámara puede resolverse de forma no lineal usando, por ejemplo, el algoritmo de Gauss-Newton o usando el algoritmo de Levenberg-Marquardt, los cuales se conocen básicamente por el experto y se describen, por ejemplo, en P. Gill, W. Murray y M. Wright: "Practical Optimization", Academic Press, páginas 134 - 137, 1981.

10 Estimación inicial de la matriz de transformación T

10 Ambos algoritmos (Gauss-Newton o Levenberg-Marquardt) requieren una estimación inicial T1 de la matriz de transformación T (véase la etapa inicial respectiva en los procesos tal como se muestra en las figuras 2-5). La estimación inicial T1 se usa como base inicial para el proceso de minimización iterativa no lineal subsiguiente para depurar de forma iterativa la matriz de transformación inicial T1 en una matriz de transformación final T que se usa para determinar la postura de cámara.

15 Lo siguiente es una realización de cómo se obtiene la estimación inicial T1 de la matriz de transformación T:

$$\text{Resolver de forma lineal la matriz } L = \begin{bmatrix} a1 & a2 & a3 & b1 \\ a4 & a5 & a6 & b2 \\ a7 & a8 & a9 & b3 \end{bmatrix}$$

20 Cada correspondencia $P_i^* \leftrightarrow m_i$ proporciona un conjunto de ecuaciones $m_i \times L P_i^* = 0$ (x es el operador para el producto vectorial), tal como se muestra en la figura 11.

25 Entonces, mediante la acumulación de todas las matrices C_i es posible escribir el sistema como un problema lineal $C1=0$ en el que lo desconocido es el vector 1 (véase la figura 11).

30 Con el fin de resolver este problema, se puede realizar la descomposición en valores singulares de la matriz como $C = U_c S_c V_c^T$, en la que U_c y V_c son dos matrices ortogonales y S_c es una matriz diagonal en la que las entradas de la diagonal se corresponden con los valores singulares de la matriz C. La última columna de la matriz V_c se corresponde con la solución de mínimos cuadrados del problema lineal.

35 Con el fin de tener un mejor acondicionamiento de la matriz C, se pueden normalizar en primer lugar los puntos en 3D de tal modo que su media es [0 0 0 1] y su desviación estándar es raíz cuadrada de (3) y se deberían normalizar los puntos en 2D de tal modo que su media [0 0 1] y su desviación estándar sea raíz cuadrada de (2). Después de la resolución del sistema, a continuación, se debería desnormalizar la solución obtenida.

40 Lo anterior se conoce básicamente por el experto y se describe en, por ejemplo: Richard Hartley and Andrew Zisserman, "Multiple View Geometry in Computer Vision", Cambridge University press, segunda edición, 2003 (véase, por ejemplo, el capítulo 7, páginas 178-194).

40 Cuando se tiene 1 es posible construir la matriz L. No obstante, no se garantiza que la matriz L se encuentre en SE(3) debido a que no se garantiza que la matriz

$$A = \begin{bmatrix} a1 & a2 & a3 \\ a4 & a5 & a6 \\ a7 & a8 & a9 \end{bmatrix}$$

45 Sea una matriz de rotación.

Se puede aproximar la matriz A por una matriz de rotación R_a tal como sigue:

50 - Realizar la descomposición en valores singulares de $A = U_a S_a V_a^T$ (en la que U_a y V_a son de nuevo matrices ortogonales y S_a es una matriz diagonal en la que las entradas de la diagonal se corresponden con los valores singulares de la matriz A)

$$D_a = \begin{bmatrix} 1 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & \det(U_a V_a^T) & \end{bmatrix}$$

- $R_a = U_a D_a V_a^T$ en la que

La matriz R_a es una matriz de rotación debido a que la misma verifica $R_a R_a^T = I_d$ y $\det(R_a) = 1$.

5 La matriz $T_1 = \begin{bmatrix} R_a & b \\ 0 & 1 \end{bmatrix}$ es entonces una solución posible para la postura T_1 .

Esta postura se obtuvo en Reimer lugar a través de un sistema lineal y forzando a continuación la matriz (3x3) superior izquierda para que fuera una matriz de rotación.

10 Proceso de minimización iterativa no lineal

En este momento, se tiene una estimación inicial de la matriz de transformación T y se está preparado para una estimación no lineal. Pueden usarse tanto el algoritmo de Gauss-Newton como el de Levenberg-Marquardt. Véanse los diagramas de flujo de las figuras 2-4 para una descripción general de estos algoritmos.

15 La implementación de Gauss-Newton convencional puede verse en la figura 2. En la figura 2, la tarea “Resolver el sistema: $J d = e$ ” puede hacerse de varias formas. Existen unos algoritmos muy eficientes para encontrar el vector d conociendo la matriz jacobiana J y el vector de error e . En la figura 2, la tarea “Calcular la nueva matriz T ” depende de la parametrización y la actualización (tal como se describe con más detalle a continuación). Por lo tanto, esto
20 puede hacerse también de varias formas. Una posible implementación se muestra en la figura 3. En el presente caso, el vector que contiene el parámetro de actualización se resuelve usando la inversa generalizada J_p de la matriz jacobiana J : $J_p = (J^T J)^{-1} J^T$ y la actualización se realiza usando un enfoque compuesto, es decir, construyendo una matriz de actualización $\Delta T(d)$ que depende del parámetro de actualización d y multiplicando entonces la matriz antigua T con la matriz de actualización $\Delta T(d)$.

25 De manera similar, una implementación de Levenberg-Marquardt convencional puede verse en la figura 4, y una posible implementación está representada en la figura 5 en la que la actualización se resuelve usando la inversa generalizada y usando un enfoque compuesto.

30 Ambos algoritmos actualizan (depuran) de forma iterativa la estimación de postura: $T \leftarrow \text{actualizar}(T, d)$, en la que d es un vector (6x1) que contiene los 6 parámetros de actualización de la traslación y la rotación y la función de actualización depende de la parametrización elegida para d .

35 La parametrización El método dado a conocer por Bleser y col.: “Online camera pose estimation in partially known and dynamic scenes”, en IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR’06), 22-25 de octubre, páginas 56-65, 2006, CA, Santa Barbara, CA, EE.UU., para una estimación de postura de cámara en un entorno dinámico cuando el modelo de CAD de un objeto en la escena está disponible usa un algoritmo de Levenberg-Marquardt para minimizar la función de error, en el que la postura del fotograma anterior se proporciona como suposición inicial. Adicionalmente, se añaden pesos a los diferentes puntos de acuerdo con la covarianza
40 conjunta de correspondencias 2D/3D. De hecho, dentro del método, la matriz jacobiana se actualiza durante el proceso de minimización iterativa, véase la sección 7.2. El documento GB 2 315 124 A da a conocer un sistema para la determinación con conexión de una postura de cámara a distancia basándose en la extracción sin conexión de características que es usando una caída de gradiente convencional (es decir, un proceso de minimización iterativa) en el que se calcula una matriz jacobiana para cada vector de error (para cada iteración, véase la figura 3 siendo J la
45 matriz jacobiana, véase la página 12).

En general, la traslación se parametriza usando una parametrización R^3 natural. Las 3 primeras entradas de d representan la traslación. No obstante, para la rotación puede seleccionarse una parametrización basándose en los ángulos de Euler o en la representación eje-ángulo o en la representación de cuaterniones. Respectivamente, las 3
50 últimas entradas de d son o bien los ángulos de Euler o bien la representación eje-ángulo de la rotación. El vector $d = [d_1, d_2, d_3, d_4, d_5, d_6]$ contiene 3 entradas para la traslación $[d_1, d_2, d_3]$ y 3 entradas para la rotación $[d_4, d_5, d_6]$.

Cuando la rotación se parametriza usando cuaterniones, el vector d tiene 7 entradas y los últimos 4 parámetros representan la representación de cuaterniones de la rotación. Por ejemplo, en el caso de que se usen ángulos de Euler para la parametrización de la rotación, se tendría la matriz T en la siguiente forma

T =

$$\begin{bmatrix} \cos(d6)*\cos(d5) & -\text{sén}(d6)*\cos(d4)+\cos(d6)*\sin(d5)*\sin(d4) & \text{sén}(d6)*\text{sén}(d4)+\cos(d6)*\text{sén}(d5)*\cos(d4) & d1 \\ \text{sén}(d6)*\cos(d5) & \cos(d6)*\cos(d4)+\text{sén}(d6)*\sin(d5)*\sin(d4) & -\cos(d6)*\text{sén}(d4)+\sin(d6)*\text{sén}(d5)*\cos(d4) & d2 \\ [-\text{sén}(d5) & \cos(d5)*\text{sén}(d4) & \cos(d5)*\cos(d4) & d3 \end{bmatrix}$$

Puede usarse una parametrización de álgebra de Lie. Esta parametrización se usa con frecuencia en la comunidad robótica. Los principios de la parametrización, tal como se describe a continuación, se conocen por el experto, y se describen en la bibliografía tal como sigue:

En "Tracking people with twists and exponential Maps", Bregler y Malik, en IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Santa Barbara, California, páginas 8-15, 1998, la parametrización que se describe anteriormente se introdujo para el seguimiento de personas.

En "Real-Time Visual Tracking of Complex Structures", Tom Drummond y Roberto Cipolla, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, n.º 7, páginas 932 - 946, julio de 2002 se usó para la parametrización de incrementos en el contexto de seguimiento basado en bordes.

Recientemente en "Motilal Agrawal, A Lie Algebraic Approach for Consistent Pose Registration for General Euclidean Motion", International Conference in Computer Vision, páginas 1891 - 1897, 2005 se usó para registrar estimaciones de postura relativa local y para producir una trayectoria consistente global de un robot.

Cuando se usa esta parametrización, la matriz T tiene la siguiente forma:

$$T = \begin{bmatrix} R(d4,d5,d6) & t(d1,d2,d3,d4,d5,d6) \\ 0 & 1 \end{bmatrix}$$

en la que la matriz de rotación se escribe usando la fórmula de Rodriguez:

$$R(d4,d5,d6) = Id + \text{sén}(\theta) [u]_x + (1 - \cos(\theta)) [u]_x^2$$

θ es el ángulo de la rotación ($\theta = \|[d4,d5,d6]\|$) y $[u]_x$ es la matriz de inclinación del eje de rotación $u = [ux, uy, uz] = [d4,d5,d6]/\theta$

$$[u]_x = \begin{bmatrix} 0 & -uz & uy \\ uz & 0 & -ux \\ -uy & ux & 0 \end{bmatrix}$$

Y la matriz de traslación depende de todos los parámetros del vector de actualización

$$t(d1,d2,d3,d4,d5,d6) = (Id + (1-\cos(\theta)) [u]_x / \theta + (1-\text{sén}(\theta) / \theta)* [u]_x^2)*[d1, d2, d3]$$

Esta parametrización tiene muchas ventajas en comparación con la convencional (ángulos de Euler para la rotación y R^3 para la traslación). Una de las ventajas que se refiere a la presente invención es que la matriz jacobiana (la derivada de la matriz de actualización con respecto a los parámetros de actualización d) tiene una expresión muy sencilla y barata (en cuanto a costes computacionales).

La actualización de la matriz de transformación T

Para cada parametrización, puede elegirse o bien una actualización adicional o bien una actualización compuesta. La actualización adicional construye una matriz actualizada T añadiendo a los parámetros antiguos los nuevos parámetros. La actualización compuesta calcula la matriz actualizada T como la multiplicación entre una matriz

construida con los parámetros de actualización $\Delta T(d)$ y la matriz antigua T. La actualización puede escribirse como:

$$T \leftarrow \Delta T(d) T$$

5 Los diferentes métodos de actualización se conocen básicamente por el experto y se describen, por ejemplo, en: Lucas-Kanade 20 Years On: A Unifying Framework”, Simon Baker y Iain Matthews, International Journal of Computer Vision, vol. 56, n.º 3, páginas 221-255, 2004 en el que las diferentes actualizaciones disponibles se explican en el contexto de seguimiento sin marcadores.

10 El proceso de minimización iterativa

En cada iteración, se transforman los puntos en 3D de referencia P_i^* con la matriz de transformación estimada T (que en la primera iteración se ajusta a T_1 , como se describe anteriormente). Se obtiene la ecuación $P_i = T P_i^* = [X_i, Y_i, Z_i, 1]$.

15 Entonces, los puntos en 3D obtenidos P_i se proyectan en $m_{e_i} = [X_i/Z_i, Y_i/Z_i]$. Es importante observar que m_{e_i} depende de la estimación de T.

20 Para cada correspondencia, se calcula $[e_{1x}, e_{1y}] = [x_i, y_i] - m_{e_i}$, en la que $[x_i, y_i]$ son las dos primeras entradas del punto “normalizado” m_i que se explica anteriormente.

Acumular todas las correspondencias da como resultado el vector de error geométrico, es decir, el vector $e = [e_{1x}, e_{1y}, e_{2x}, e_{2y}, \dots, e_{Nx}, e_{Ny}]$ en el que N es el número total de puntos.

25 La estimación no lineal minimiza de forma iterativa la norma al cuadrado de este vector de error.

Por este motivo, se necesita también calcular la matriz jacobiana que es la primera derivada de m_{e_i} con respecto a los incrementos de rotación y traslación. La dimensión de la matriz jacobiana es entonces $(2N \times 6)$.

30 Las líneas $2 \cdot i$ y $(2 \cdot i + 1)$ de la matriz jacobiana pueden escribirse como: $\Delta(m_{e_i}) / \Delta d$

La matriz jacobiana puede calcularse por medios numéricos usando el método de diferencias finitas. Como alternativa, puede calcularse por medios analíticos. Para ambos enfoques, la parametrización de matriz de incremento desempeña un papel importante en la complejidad del cálculo de jacobiana. Por ejemplo, en el caso de una actualización de rotación parametrizada usando los ángulos de Euler, es necesario calcularse y volver a calcularse varias funciones trigonométricas (por ejemplo, cos y sen) en cada iteración. El modo de actualizar la estimación (adicional o compuesta) desempeña también un papel importante en la complejidad del algoritmo.

De acuerdo con realizaciones de la invención, se realizan las siguientes consideraciones:

40 En la figura 6, se muestra una posible implementación del algoritmo de Gauss-Newton para una estimación de postura no lineal en la que la matriz jacobiana se calcula previamente después de la etapa inicial de determinar la estimación inicial de la matriz de transformación T_1 y la actualización se realiza de una forma compuesta. Con respecto al proceso convencional tal como se muestra en la figura 3, de acuerdo con el presente aspecto de la invención, el proceso de iteración no vuelve al cálculo (que requiere mucho tiempo) de una nueva matriz jacobiana basándose en una nueva matriz de transformación T_{nueva} , sino que vuelve a la etapa de cálculo del vector de error e. De acuerdo con la invención, mantener fija la matriz jacobiana y no actualizarla durante el proceso de minimización iterativa proporciona aún muy buenos resultados. La matriz jacobiana calculada justo después de la solución lineal de la matriz de transformación T obteniendo la matriz T_1 es una buena aproximación que permite alcanzar el mínimo de la función de coste con una potencia de cálculo mucho menor.

45 En la figura 7, se muestra que en el caso de que se use el algoritmo de Gauss-Newton para una estimación de postura no lineal, si la actualización se resuelve usando la inversa generalizada J_p , la inversa generalizada J_p asimismo puede calcularse previamente. Por lo tanto, con respecto al proceso de la figura 6 la carga de cálculo en el bucle de iteración puede reducirse adicionalmente, en particular cuando se resuelve el sistema $J d = e$ de una manera eficiente con el uso de la ecuación $d = J_p^{-1} e$ y calculando previamente J_p .

50 En la figura 8, se muestra una posible implementación del algoritmo de Levenberg-Marquardt para una estimación de postura no lineal en la que la matriz jacobiana J y la matriz $J^T J$ se calculan previamente y la actualización se realiza de una forma compuesta. Con respecto al proceso convencional tal como se muestra en la figura 5, de acuerdo con el presente aspecto de la invención el proceso de iteración no vuelve al cálculo (que requiere mucho tiempo) de una nueva matriz jacobiana basándose en una nueva matriz T_{nueva} , sino que vuelve a la etapa de cálculo del vector de error e o la etapa de resolución del sistema $(J^T J + \lambda Id) d = J^T e$, respectivamente. De acuerdo con la invención, mantener fija la matriz jacobiana y no actualizarla durante el

proceso de minimización iterativa proporciona aún muy buenos resultados.

5 En la figura 9, se muestra una posible implementación del algoritmo de Levenberg-Marquardt para una estimación de postura no lineal en la que, ambas, la matriz jacobiana J y la matriz $J^T J = Q$ se calculan previamente y la actualización se calcula usando la inversa generalizada. Por lo tanto, con respecto al proceso de la figura 5 puede aumentarse significativamente la carga de cálculo en el bucle de iteración.

10 En la figura 10, se muestra una posible implementación del algoritmo de Levenberg-Marquardt para una estimación de postura no lineal en la que la actualización se calcula usando la inversa generalizada, en la que la matriz jacobiana J y la matriz $J^T J$ se calculan previamente y la actualización se calcula usando la inversa generalizada, y en la que se han realizado algunos cálculos previos más (tal como la descomposición del valor propio de $J^T J$, que se conoce básicamente por el experto en la materia) con el fin de tener una matriz más sencilla de invertir en el bucle principal.

15 De acuerdo con algunos aspectos de la invención, puede elegirse una parametrización apropiada de la matriz de transformación T , en la que pueden elegirse múltiples tipos de parametrizaciones. De acuerdo con una realización, se decide parametrizar la matriz ΔT usando el álgebra de Lie $se(3)$ asociada con el grupo de Lie $SE(3)$ (véase anteriormente).

20 La complejidad de cálculo de jacobiana se reduce entonces considerablemente. Se obtiene una expresión sencilla y analítica de la matriz jacobiana. Además, cuando se usa el enfoque de actualización compuesta, pueden calcularse previamente partes de la matriz jacobiana obtenida. Esto reduce el procesamiento en tiempo de ejecución. Véase el efecto de tal parametrización en la simplificación de un cálculo de jacobiana en el contexto de seguimiento sin marcadores plano en: "Integration of Euclidean constraints in template based visual tracking of piecewise-planar scenes", Selim Benhimane y Ezio Malis, en Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 9 - 15 de octubre, páginas 1218 - 1223, 2006, Pekín, China. El error de la función de coste en este documento se basaba en el error fotométrico entre una plantilla de referencia y platilla actual. No se basaba en el error geométrico tal como se considera en la presente invención.

30 Además de lo anterior, la estimación no lineal sigue requiriendo mucho tiempo y ejecutarla en hardware con una potencia de cálculo muy limitada, tal como en dispositivos móviles, hace imposible tener la estimación de postura para realizarse en tiempo real especialmente con un número de puntos muy elevado. De acuerdo con la invención, manteniendo fija la matriz jacobiana y no actualizándola durante el proceso de minimización iterativa (o bien usando el método de Gauss-Newton o bien usando el método de Levenberg-Marquardt) proporciona aún muy buenos resultados. La matriz jacobiana calculada justo después de la solución lineal es una buena aproximación que permite alcanzar el mínimo de la función de coste con una potencia de cálculo mucho menor.

40 De acuerdo con una realización de la invención, los diferentes aspectos tal como se describen anteriormente pueden también combinarse en un método para determinar la postura de una cámara. Por ejemplo, usando la parametrización de álgebra de Lie durante una estimación no lineal basándose en una actualización compuesta se reduce el coste de cálculo del problema de la estimación de postura. Usando una matriz jacobiana fija, tal como se explica con respecto a las figuras 6-10, se reduce el coste de cálculo sin deteriorar el resultado.

45 El método puede implementarse en un programa informático adaptado para cargarse en la memoria interna de un ordenador. Tal ordenador puede incluir además, por ejemplo, un procesador convencional 2 tal como se muestra en la figura 1 y puede estar acoplado con una cámara, tal como la cámara 1 que se muestra en la figura 1, para tomar imágenes de un entorno real. El producto de programa informático que tiene el programa informático almacenado en el mismo comprende secciones de código de software por medio de las cuales se realizan las etapas de acuerdo con cualquiera de las realizaciones que se mencionan anteriormente del método de acuerdo con la invención
50 cuando dicho programa o producto se está ejecutando en dicho ordenador. En una realización adicional, el método puede también estar integrado o incrustado en un chip usado en conexión con el procesamiento de imágenes de cámara. Por ejemplo, tal chip incrustado puede estar acoplado con o incorporado en el procesador 2 o la cámara 1.

REIVINDICACIONES

1. Un método para determinar la postura de una cámara con respecto a por lo menos un objeto real, comprendiendo el método las etapas de:

5 - accionar la cámara (1) para capturar una imagen bidimensional (4) que incluye por lo menos una parte del objeto real (3), **caracterizado por:**

10 - proporcionar una matriz de transformación (T) que incluye información con respecto a una correspondencia entre puntos tridimensionales (P_i^*) asociados con el objeto real (3) y unos puntos bidimensionales correspondientes (p_i) del objeto real (5) tal como se incluyen en la imagen bidimensional (4),

15 - determinar una estimación inicial de la matriz de transformación (T1) como base inicial para un proceso de minimización iterativa usado para depurar de forma iterativa la matriz de transformación,

15 - determinar una matriz jacobiana (J) que incluye información con respecto a la estimación inicial de la matriz de transformación (T1) y valores de referencia de los puntos tridimensionales (P_i^*) asociados con el objeto real (3),

20 - en el proceso de minimización iterativa, determinar en cada uno de múltiples bucles de iteración una versión actualizada respectiva de la matriz de transformación (T) basándose en una versión previa respectiva de la matriz de transformación (T) y basándose en la matriz jacobiana (J), en el que la matriz jacobiana no se actualiza durante el proceso de minimización iterativa, y

20 - determinar la postura de la cámara (1) con respecto al objeto real (3) usando la matriz de transformación (T) determinada al final del proceso de minimización iterativa.

25 2. Un método para determinar la postura de una cámara con respecto a por lo menos un objeto real, comprendiendo el método las etapas de:

- accionar la cámara (1) para capturar una imagen tridimensional (4) que incluye por lo menos una parte del objeto real (3),

30 - proporcionar una matriz de transformación (T) que incluye información con respecto a una correspondencia entre puntos tridimensionales (P_i^*) asociados con el objeto real (3) y unos puntos tridimensionales correspondientes (p_i) del objeto real (5) tal como se incluyen en la imagen tridimensional (4),

- determinar una estimación inicial de la matriz de transformación (T1) como base inicial para un proceso de minimización iterativa usado para depurar de forma iterativa la matriz de transformación,

35 - determinar una matriz jacobiana (J) que incluye información con respecto a la estimación inicial de la matriz de transformación (T1) y valores de referencia de los puntos tridimensionales (P_i^*) asociados con el objeto real (3),

- en el proceso de minimización iterativa, determinar en cada uno de múltiples bucles de iteración una versión actualizada respectiva de la matriz de transformación (T) basándose en una versión previa respectiva de la matriz de transformación (T) y basándose en la matriz jacobiana (J), en el que la matriz jacobiana no se actualiza durante el proceso de minimización iterativa, y

40 - determinar la postura de la cámara (1) con respecto al objeto real (3) usando la matriz de transformación (T) determinada al final del proceso de minimización iterativa.

45 3. El método de acuerdo con la reivindicación 1 o 2, en el que el proceso de minimización iterativa implica el algoritmo de Gauss-Newton o el algoritmo de Levenberg-Marquardt.

4. El método de acuerdo con una de las reivindicaciones 1 a 3, en el que el proceso de minimización iterativa implica un proceso de actualización compuesta en el que la versión actualizada respectiva de la matriz de transformación (T) se calcula a partir de una multiplicación entre una matriz ($\Delta T(d)$) construida con parámetros de actualización de la matriz de transformación previa respectiva y la matriz de transformación previa respectiva (T).

5. El método de acuerdo con una de las reivindicaciones 1 a 4, en el que el proceso de minimización iterativa implica el algoritmo de Gauss-Newton, comprendiendo el proceso de minimización iterativa en cada bucle de iteración las etapas de:

- determinar un vector de error e,
- determinar un vector de actualización d mediante la resolución del sistema: $J d = e$, siendo J la matriz jacobiana,

60 - determinar una versión actualizada respectiva de la matriz de transformación basándose en una versión previa respectiva de la matriz de transformación al comienzo del bucle de iteración respectivo y basándose en el vector de actualización d,

- en el que, después de determinar la versión actualizada respectiva de la matriz de transformación, el proceso de minimización iterativa vuelve a determinar un nuevo vector de error e basándose en la versión actualizada respectiva de la matriz de transformación.

65

6. El método de acuerdo con la reivindicación 5, en el que el vector de actualización d se calcula a partir de: $d = (J^T J) - I J^T e = J_p e$, calculándose J_p al comienzo del proceso de minimización iterativa y manteniéndose fija en múltiples bucles de iteración durante el proceso de minimización iterativa.
- 5
7. El método de acuerdo con una de las reivindicaciones 1 a 4, en el que el proceso de minimización iterativa implica el algoritmo de Levenberg-Marquardt, comprendiendo el proceso de minimización iterativa en cada bucle de iteración las etapas de:
- 10
- determinar un vector de error e ,
 - determinar un vector de actualización d mediante la resolución del sistema: $(J^T J + \lambda I d) d = J^T e$, siendo J la matriz jacobiana,
 - determinar una versión actualizada respectiva de la matriz de transformación basándose en una versión previa respectiva de la matriz de transformación al comienzo del bucle de iteración respectivo y basándose en el vector de actualización d ,
- 15
- en el que, después de determinar la versión actualizada respectiva de la matriz de transformación, el proceso de minimización iterativa vuelve a determinar un nuevo vector de error e basándose en la versión actualizada respectiva de la matriz de transformación o determinar un nuevo vector de actualización d .
- 20
8. El método de acuerdo con la reivindicación 7, en el que el vector de actualización d se calcula a partir de: $d = (Q + \lambda I d)^{-1} J^T e$, calculándose $Q = J^T J$ al comienzo del proceso de minimización iterativa y manteniéndose fija en múltiples bucles de iteración durante el proceso de minimización iterativa.
- 25
9. El método de acuerdo con una de las reivindicaciones 1 a 8, en el que en el proceso de minimización iterativa se usa una parametrización de álgebra de Lie.
10. El método de acuerdo con una de las reivindicaciones 5 a 8, que además comprende
- 30
- determinar una matriz de actualización $\Delta T(d)$ usando el vector de actualización d ,
 - en el que la matriz de actualización $\Delta T(d)$ se parametriza usando álgebra de Lie.
- 35
11. El método de acuerdo con la reivindicación 10, en el que la matriz de actualización $\Delta T(d)$ se parametriza usando álgebra de Lie $se(3)$ asociada con el grupo de Lie $SE(3)$.
- 40
12. Un producto de programa informático adaptado para cargarse en la memoria interna de un ordenador, que está acoplado con una cámara para tomar imágenes de un entorno real, y que comprende secciones de código de software por medio de las cuales las etapas de acuerdo con cualquiera de las reivindicaciones 1 a 11 se realizan cuando dicho producto se está ejecutando en dicho ordenador.

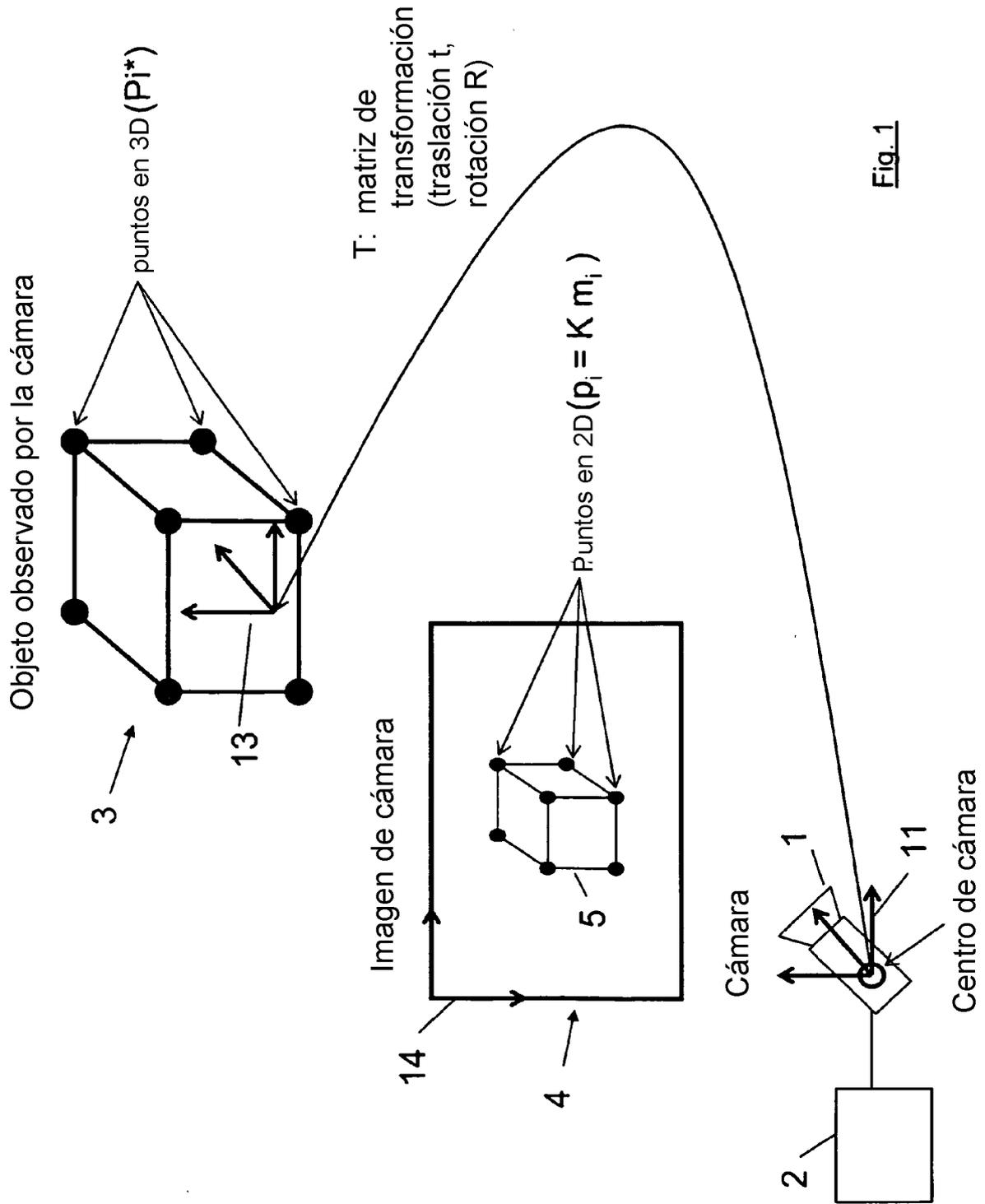


Fig.1

Fig. 2

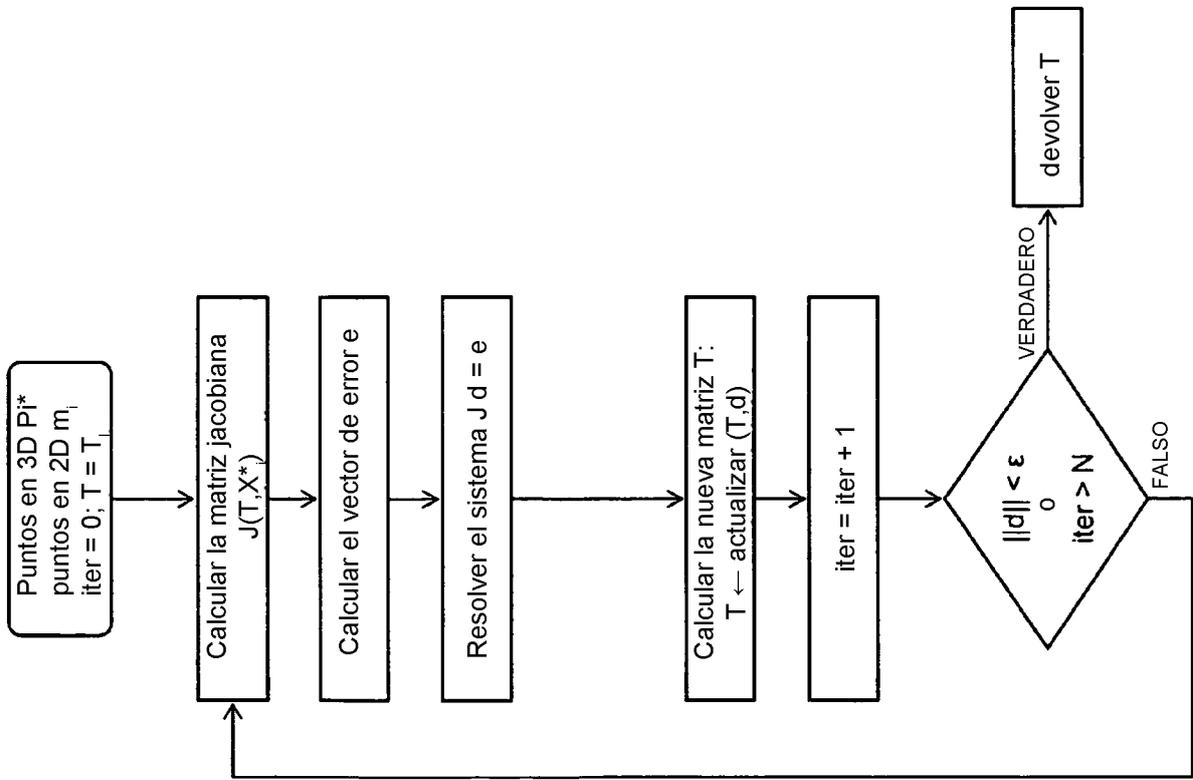


Fig. 3

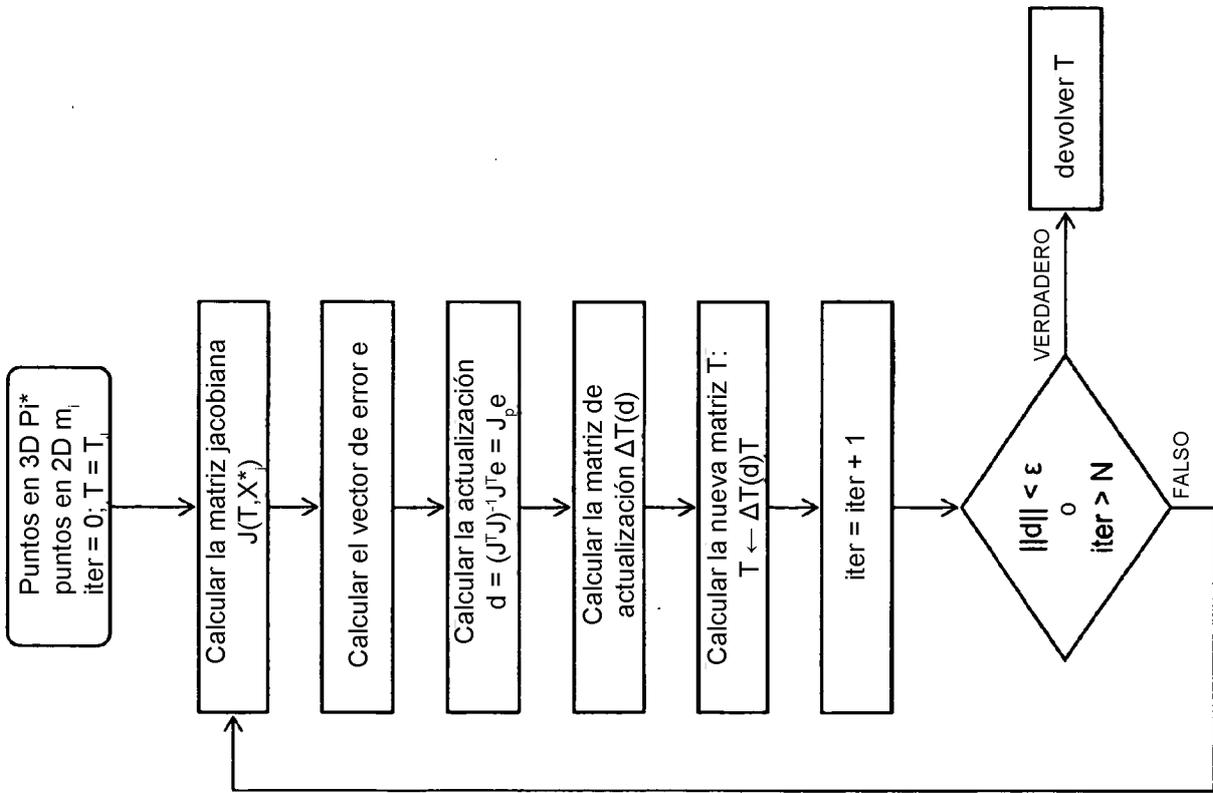


Fig. 4

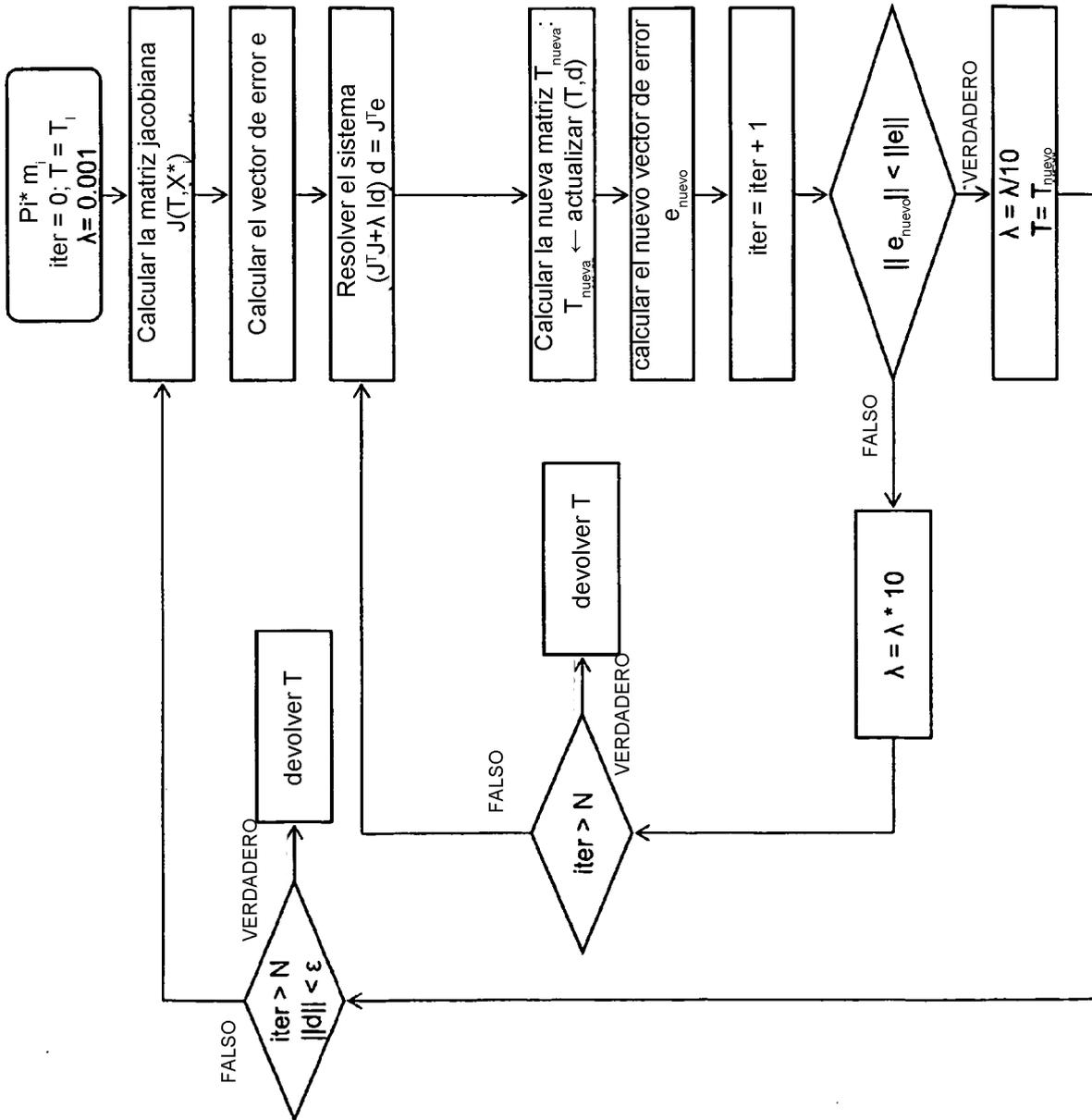


Fig. 5

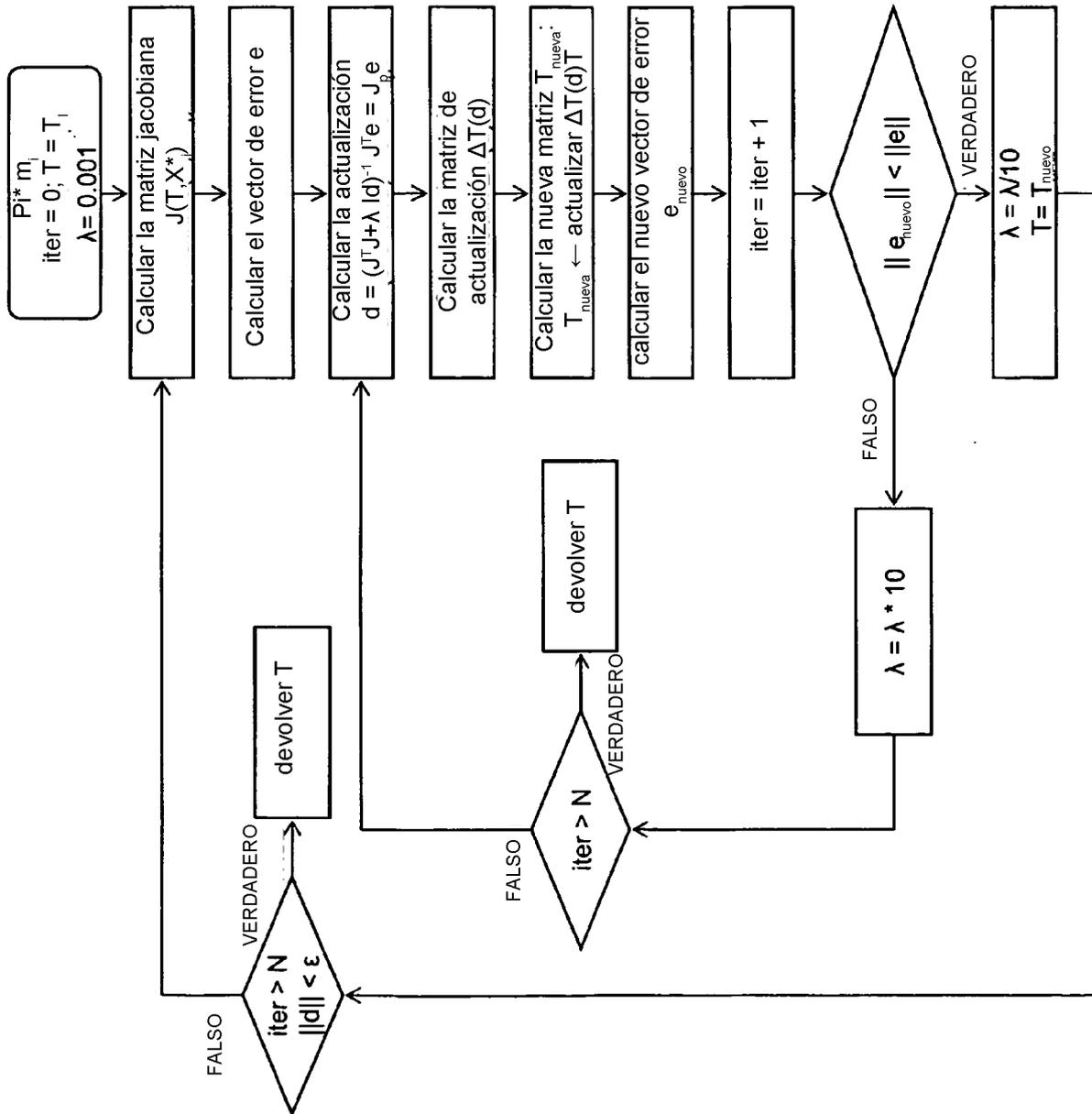


Fig. 6

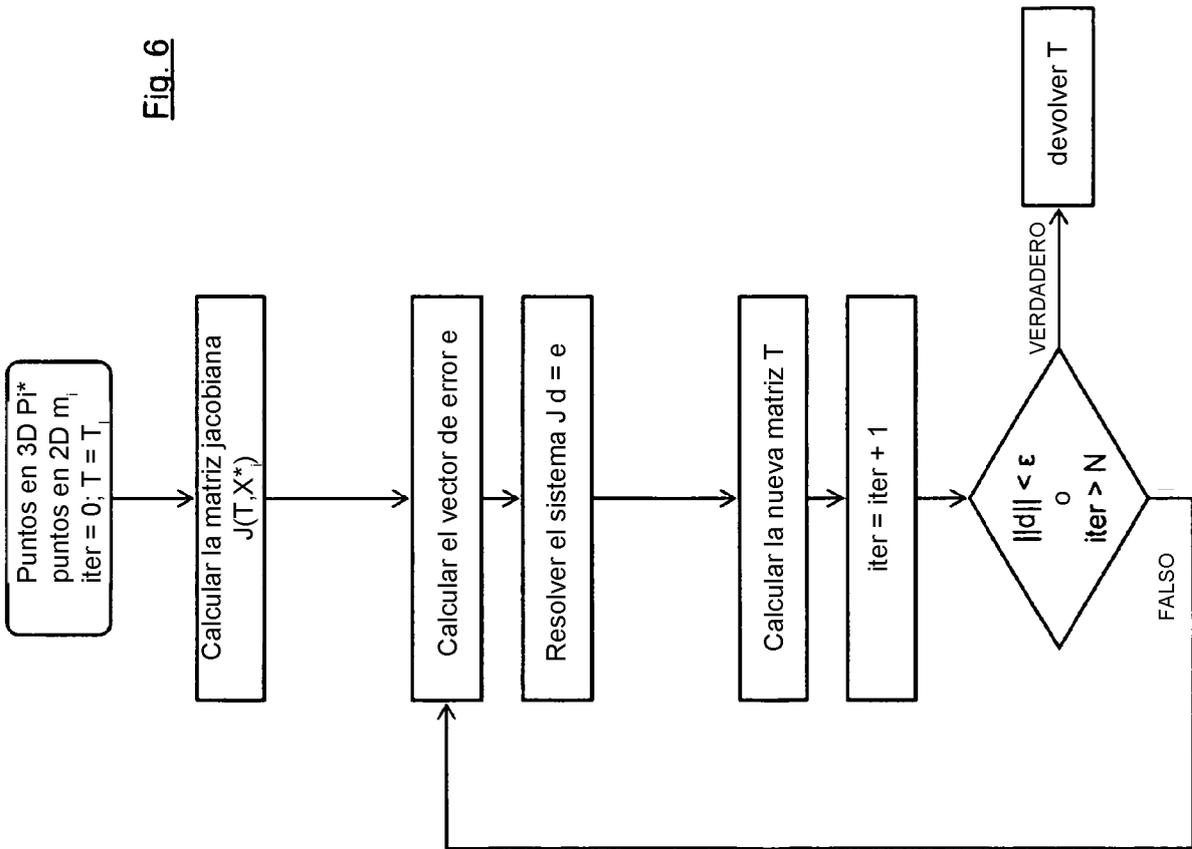
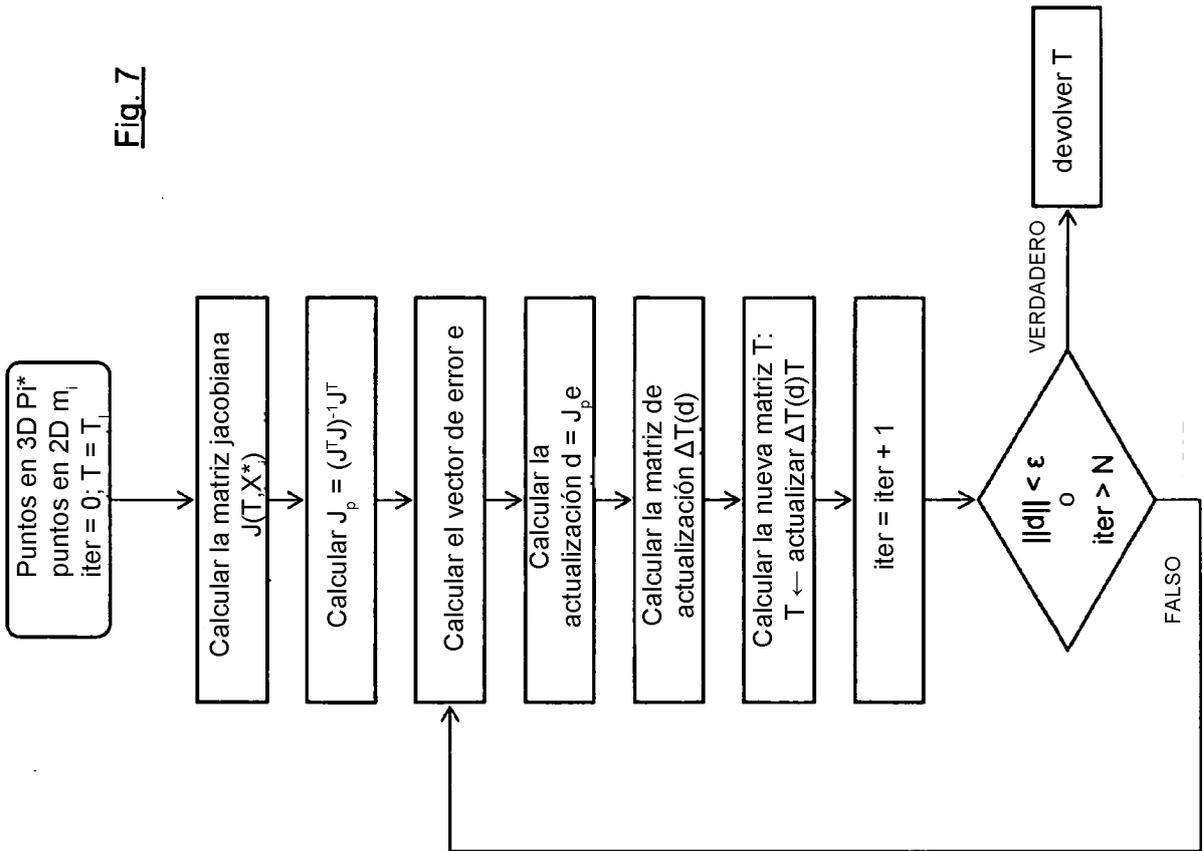


Fig. 7



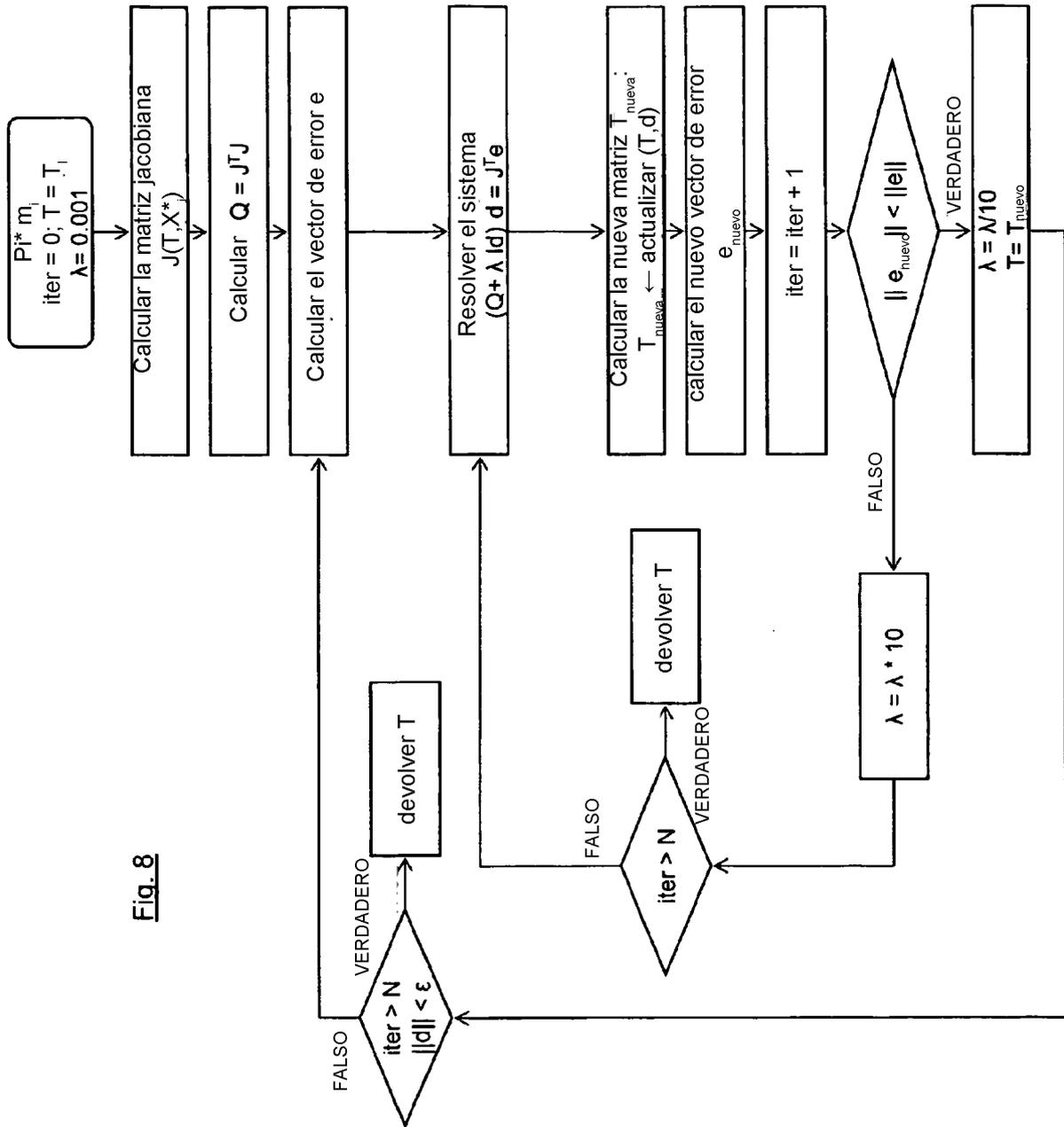


Fig. 8

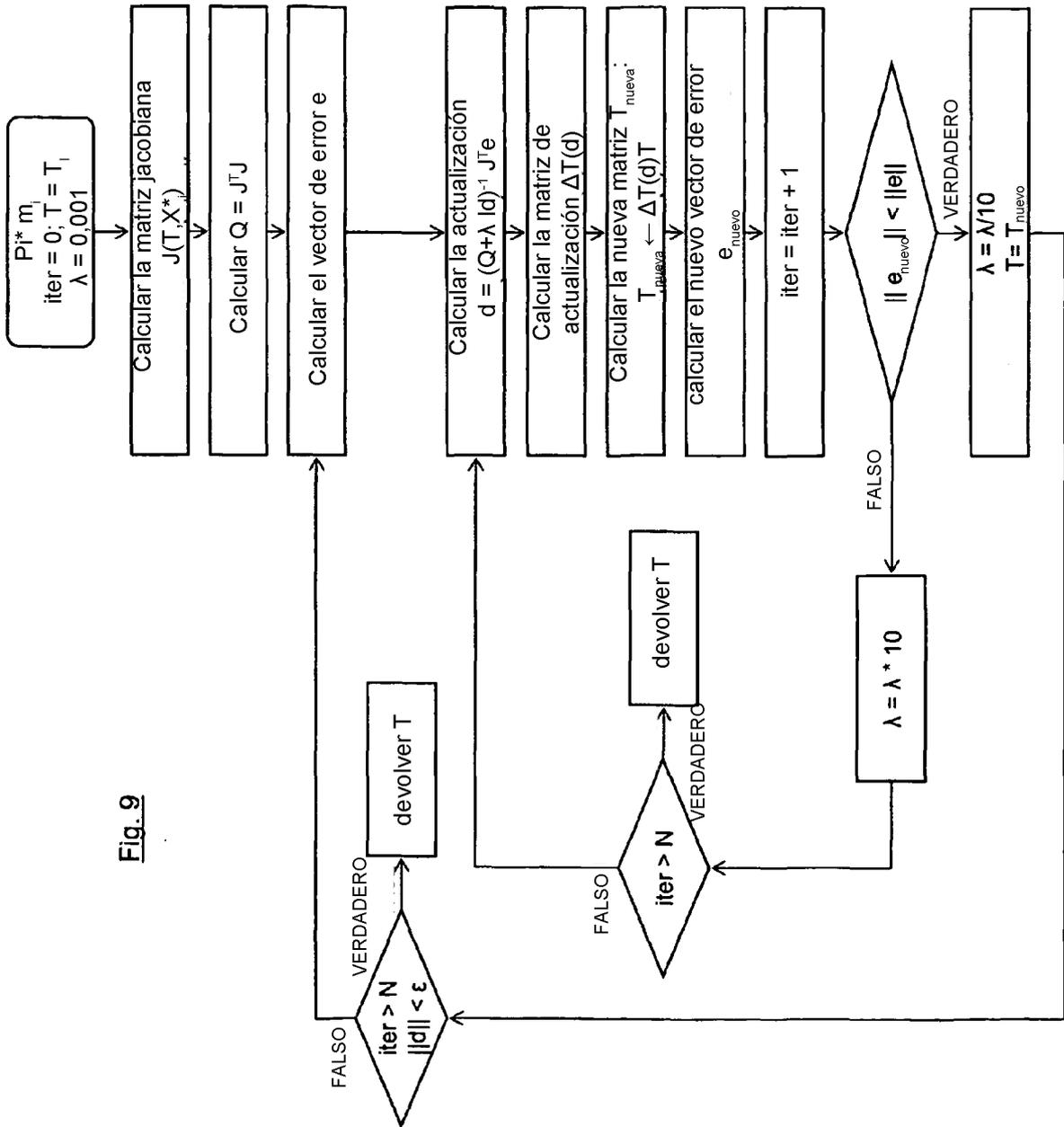


Fig. 9

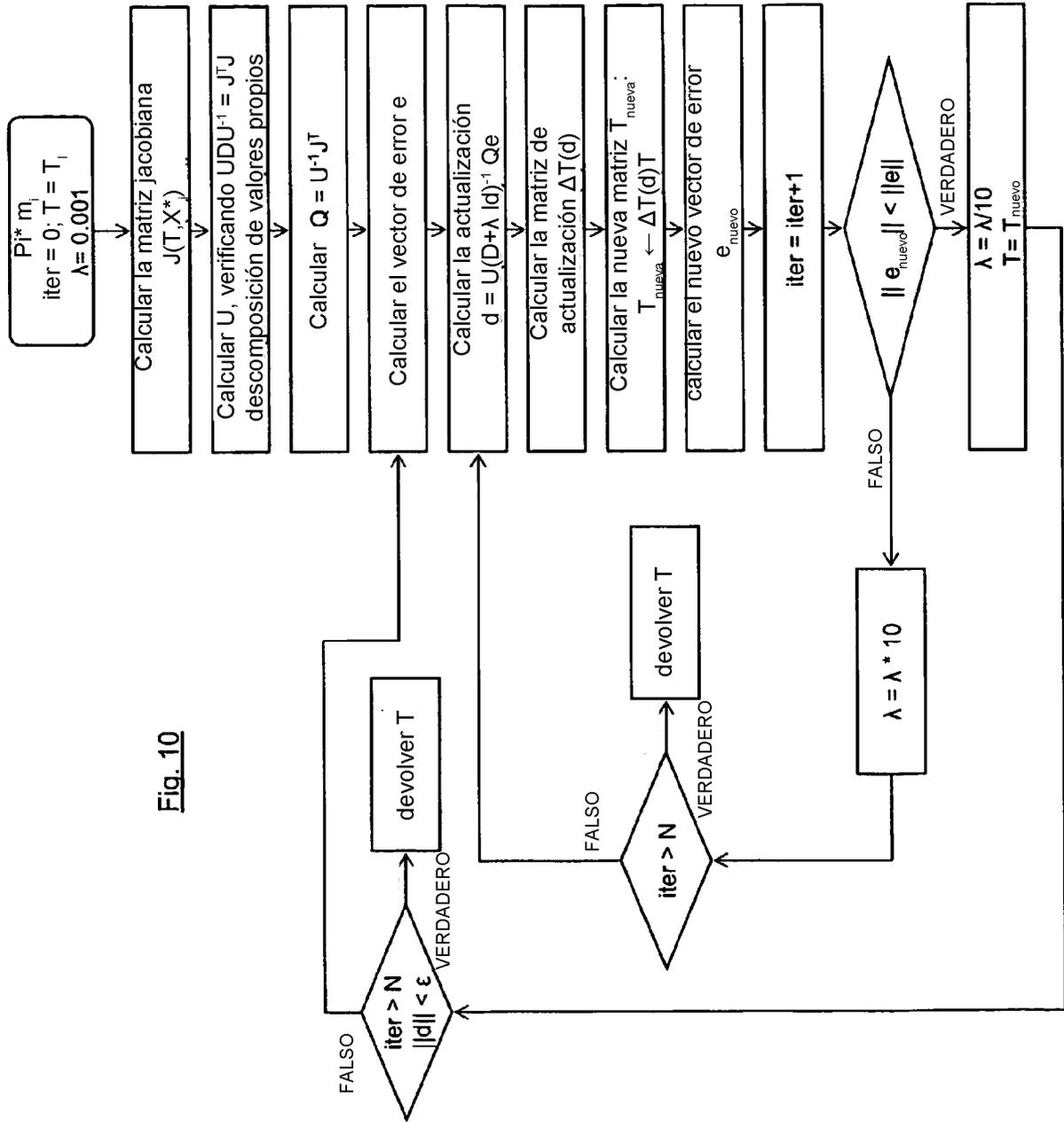


Fig. 11

Conjunto de ecuaciones

$m_i \times L \Pi_i^* = 0$ (x es el producto vectorial)

$$\begin{array}{l}
 [0 \quad 0 \quad 0 \quad -X_i^* \quad -Y_i^* \quad -Z_i^* \quad y_i X_i^* \quad y_i Y_i^* \quad y_i Z_i^*] \quad [a_1] = [0] \\
 [X_i^* \quad Y_i^* \quad Z_i^* \quad 0 \quad 0 \quad 0 \quad -x_i X_i^* \quad -x_i Y_i^* \quad -x_i Z_i^*] \quad [a_2] \quad [0] \\
 [-y_i X_i^* \quad -y_i Y_i^* \quad -y_i Z_i^* \quad x_i X_i^* \quad x_i Y_i^* \quad x_i Z_i^* \quad 0 \quad 0 \quad 0] \quad [a_3] \quad [0] \\
 \quad [b_1] \\
 \quad [a_4] \\
 \quad [a_5] \\
 \quad [a_6] \\
 \quad [b_2] \\
 \quad [a_7] \\
 \quad [a_8] \\
 \quad [a_9] \\
 \quad [b_3]
 \end{array}$$

Que puede escribirse como $C_i = 0$ en la que

$$\begin{array}{l}
 C_i = [0 \quad 0 \quad 0 \quad -X_i^* \quad -Y_i^* \quad -Z_i^* \quad y_i X_i^* \quad y_i Y_i^* \quad y_i Z_i^*] \\
 \quad [X_i^* \quad Y_i^* \quad Z_i^* \quad 0 \quad 0 \quad 0 \quad -x_i X_i^* \quad -x_i Y_i^* \quad -x_i Z_i^*] \\
 \quad [-y_i X_i^* \quad -y_i Y_i^* \quad -y_i Z_i^* \quad x_i X_i^* \quad x_i Y_i^* \quad x_i Z_i^* \quad 0 \quad 0 \quad 0]
 \end{array}$$

$$Y \quad I = [a_1 \ a_2 \ a_3 \ b_1 \ a_4 \ a_5 \ a_6 \ b_2 \ a_7 \ a_8 \ a_9 \ b_3]$$