CAPÍTULO 1

POSIT: *POS* with *IT*erations

1.1. Introducción

En este capítulo se explica el algoritmo utilizado para el cálculo de la pose a partir de una imagen capturada por la cámara. Como lo dice el nombre de algoritmo se utiliza una técnica llamada *POS* (*P*ose from *O*rtography and *S*caling), esta técnica consiste en aproximar la pose de la cámara a partir de la proyección *SOP*(*S*caled *O*rtographic *Projection*). Se comienza el capítulo explicando en que consiste la proyección *SOP* y como se estima la pose a partir ella. Con esto como fundamento teórico se explican las diferentes variantes de POSIT y finalmente se explica la implementación utilizada en la aplicación.

1.2. POSIT clásico

La primera versión de POSIT presentada por *Daniel DeMenthon* y *Larry Davis* en [**?**] resuelve el problema de calcular la pose de la cámara dados 4 o más puntos detectados en la imagen y sus correspondientes en el mundo real, con la condición de que estos puntos no sean coplanares. Si bien no es la versión final que se utilizó vale la pena ser explicada ya que ayuda a sentar las bases de la implementación utilizada.

1.2.1. Notación y definición formal del problema de estimación de pose

En la figura 1.1 se puede ver un modelo de cámara pinhole, donde el centro es el punto O, G es el plano imagen ubicado a una distancia focal f de O. O_x y O_y son los ejes que apuntan en las direcciones de las filas y las columnas del sensor de la cámara respectivamente. O_z es el eje que esta sobre el eje óptico de la cámara y apunta en sentido saliente. Los versores para estos ejes son **i**, **j** y **k**.

Se considera ahora un objeto con puntos característicos M_0, M_1, ..., M_i, ..., M_n, cuyo eje de coordenadas esta centrado en M_0 y está compuesto por los versores (M_0u, M_0v, M_0w). La geometría del objeto se asume conocida, por lo tanto las coordenadas de lo puntos característicos del objeto en el eje de coordenadas del mismo son conocidas. Por ejemplo (U_i, V_i, W_i) son las coordenadas del punto M_i en el marco de referencia del objeto. Los puntos correspondientes a los puntos del objeto M_i en la imagen son conocidos y se identifican como m_i, (x_i, y_i) son las coordenadas de este punto en la imagen. Las coordenadas de los puntos M_i en el eje de coordenadas de la cámara, identificadas como (X_i, Y_i, Z_i), son desconocidas ya que no se conoce la pose del objeto respecto a la cámara.

Se busca entonces computar la matriz de rotación y el vector de traslación del objeto respecto a la cámara. La matriz de rotación **R** del objeto, es la matriz cuyas filas son las coordenadas del los versores i, j y k del sistema de coordenadas de la cámara expresados en el sistema de coordenadas del objeto (u, v, w), se puede ver como la matriz de cambio de base que pasa coordenadas en la base del objeto a coordenadas en la base de la cámara.

La matriz **R** queda:

$$R = \begin{pmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{pmatrix}$$

Para obtener la matriz de rotación solo es necesario obtener los versores **i** y **j** , el versor **k** se obtiene de realizar el producto vectorial **i** $\times$ **j**. El vector de traslación es el vector que va del centro del objeto M_0 a el centro del sistema de coordenadas de la cámara O. Por lo tanto las coordenadas del vector de traslación son (X_0,Y_0,Z_0). Si este punto M_0 es uno de los puntos visibles en la imagen, entonces el vector **T** esta alineado con el vector Om_0 y es igual a $(Z_0/f)Om_0$. Por lo tanto la pose queda determinada si se conocen **i**, **j** y Z_0.

Como detalle a tener en cuenta, se observa que para calcular la traslación es necesario conocer el punto de referencia del objeto M_0, esta es la principal diferencia con la versión moderna de POSIT en la que para calcular la traslación no es necesario suponer nada acerca del punto de referencia del objeto.

Figura 1.1: Proyección en perspectiva (m_i) y SOP (p_i) para un punto del modelo 3D M_i y un punto de referencia del modelo M_O. Fuente: [**?**] .

1.2.2. SOP: *S*caled *O*rtographic *P*rojection

La proyección ortogonal escalada(SOP) es una aproximación a la proyección perspectiva. En esta aproximación se supone que las profundidades Z_i de diferentes puntos M_i en el eje de coordenadas de la cámara no difieren mucho entre sí, y por lo tanto se asume que todos los puntos M_i tienen la misma profundidad que el punto M_0.

Para un punto M_i la proyección perspectiva sobre el plano imagen estaría dada por:

$$x_i = fX_i/Z_i, \qquad y_i = fY_i/Z_i,$$

mientras que la proyección SOP esta dada por:

$$x_i^{'} = fX_i/Z_0, \qquad y_i^{'} = fY_i/Z_0.$$

De aquí en más las proyecciones SOP de los puntos M_i se identificaran como p_i mientras que las proyecciones perspectivas, que son los puntos que se detectan en la imagen, se identifican como m_i. Al término $s = f/Z_0$ se lo conoce como el factor de escala de la SOP. Se puede ver que para el caso particular del punto M_0 la proyección perspectiva m_0 y la SOP p_0 coinciden.

En la figura 1.1 se puede ver como se construye la SOP. Primero se realiza la proyección ortogonal de todos los puntos M_i sobre K, el plano paralelo al plano imagen que pasa por el punto M_0. Las proyecciones de los puntos M_i sobre K se llaman P_i. El segundo paso consiste en hacer la proyección perspectiva de los puntos P_i sobre el plano imagen G para obtener finalmente los puntos p_i. En la figura también se puede ver que el tamaño del vector m_0p_i es s veces el tamaõ de M_0P_i. Teniendo esto en cuenta se pueden expresar las coordenadas de p_i como:

$$\begin{aligned} x_i^{'} &= fX_0/Z_0 + f(X_i - X_0)/Z_0 = x_0 + s(X_i - X_0) \\ y_i^{'} &= y_0 + s(Y_i - Y_0) \end{aligned} \tag{1.1}$$

1.2.3. Ecuaciones para calcular la proyección perspectiva

Como se mencionó anteriormente la pose queda determinada si se conocen los vectores **i**,**j** y la coordenada Z_0 del vector de traslación. Las ecuaciones que vinculan estas variables son:

$$M_0M_i\frac{f}{Z_0}\mathbf{i} = x_i(1+\varepsilon_i) - x_0 \tag{1.2}$$

$$M_0M_i\frac{f}{Z_0}\mathbf{j} = y_i(1+\varepsilon_i) - y_0 \tag{1.3}$$

donde ε_i se define como

$$\varepsilon_i = \frac{1}{Z_0}M_0M_i\mathbf{k} \tag{1.4}$$

Se puede ver que los términos $x_i(1+\varepsilon_i)$ y $y_i(1+\varepsilon_i)$ son las coordenadas $(x_i^{'}, y_i^{'})$ de la SOP. Primero se descompone el vector M_0M_i en la suma de M_0P_i y P_iM_i. De la figura 1.1 se puede ver que sentido del vector P_iM_i se según el versor **k**. Al realizar el producto escalar con **i** y **j** se tiene que:

$$\begin{aligned} M_0M_i\mathbf{i} = M_0P_i\mathbf{i} + P_iM_i\mathbf{i} = \frac{Z_0}{f}m_0p_i\mathbf{i} \\ M_0M_i\mathbf{j} = M_0P_i\mathbf{j} + P_iM_i\mathbf{j} = \frac{Z_0}{f}m_0p_i\mathbf{j} \end{aligned} \tag{1.5}$$

Recordando que p_i es la SOP y que m_0 es la proyección del punto de referencia del objeto, se tiene:

$$\begin{aligned} m_0p_i\mathbf{i} = x_i^{'} - x_0 \\ m_0p_i\mathbf{j} = y_i^{'} - y_0 \end{aligned} \tag{1.6}$$

Finalmente si se sustituye en 1.5 y se compara con 1.2 y 1.3 se puede ver que:

$$\begin{aligned} x_i^{'} = x_i(1+\varepsilon_i) \\ y_i^{'} = y_i(1+\varepsilon_i) \end{aligned} \tag{1.7}$$

1.2.4. Algoritmo

Las ecuaciones 1.2 y 1.3 se puede reescribir como:

$$M_0M_i\mathbf{I} = x_i(1+\varepsilon_i) - x_0 \tag{1.8}$$

$$M_0M_i\mathbf{J} = y_i(1+\varepsilon_i) - y_0 \tag{1.9}$$

en donde

$$\mathbf{I} = \frac{f}{Z_0}\mathbf{i}, \mathbf{J} = \frac{f}{Z_0}\mathbf{j} \tag{1.10}$$

Si se conociera el valor de ε_i, las ecuaciones 1.8 y 1.9 representan un sistema de ecuaciones en que las incógnitas son los vectores **I** y **J**. Una vez obtenidos estos vectores es si pueden obtener los versores **i** y **j** normalizando, y Z_0 se obtiene de la norma de cualquiera de los vectores **I** o **J**. A esta parte del del algoritmo se le llama *POS* (*P*ose from *O*rthography and *Scaling*), ya que estima la pose a partir de las proyecciones SOP de los puntos M_i.

Si se conocieran los valores exactos de los ε_i la pose obtenida de resolver el sistema de ecuaciones sería la pose real del objeto, como no se conocen los valores exactos de ε_i se utiliza un método iterativo que tiende a la solución buscada. En la primera iteración se le toma $\varepsilon_i = 0$, es decir, $p_i = m_i$.

Esta suposición es razonable si se tiene que la relación distancia cámara objeto - profundidad del objeto es grande. La ecuación para un punto cualquiera está dada por:

$$\begin{aligned} M_0M_i \cdot \mathbf{I} &= x_i^{'} - x_0 \\ M_0M_j \cdot \mathbf{J} &= y_i^{'} - y_0 \end{aligned} \tag{1.11}$$

Si se escribe la ecuación 1.11 para los n puntos del modelo, se tiene un sistema de ecuaciones con **I** y **J** como incógnitas

$$\begin{aligned} AI &= x^{'} - x_0 \\ AJ &= y^{'} - y_0 \end{aligned} \tag{1.12}$$

A es una matriz $n \times 3$ con las coordenadas de los puntos del modelo M_i en el marco de coordenadas del objeto. Si se tienen mas de 4 puntos y no son coplanares, la matriz **A** es de rango 3, y las soluciones al sistema están dadas por

$$\begin{aligned} I &= \mathbf{B}x^{'} - x_0 \\ J &= \mathbf{B}y^{'} - y_0 \end{aligned} \tag{1.13}$$

donde **B** es la pseudo inversa de la matriz A. Se debe notar que la matriz **B** depende únicamente de la geometría del modelo que se asume conocida, por lo tanto solo es necesario calcular la matriz **B** una sola vez.

Una vez obtenidos **I** y **J** se calculan los versores **i**, **j** y **k**

$$\begin{aligned} i &= \frac{\mathbf{I}}{\|\mathbf{I}\|} \\ j &= \frac{\mathbf{J}}{\|\mathbf{J}\|} \\ k &= i \times j \end{aligned} \tag{1.14}$$

El vector traslación del centro del objeto al centro de la cámara es el vector OM_0

$$OM_0 = \frac{Z_0}{f} Om_0 = \frac{Om_0}{s} \tag{1.15}$$

El vector Om_0 es conocido ya que se conocen las coordenadas de los puntos m_i y el factor de escala s se obtiene de la norma de los vectores **I** o **J**.

Una vez que se calcularon **i**, **j**, **k** y **T** se calculan los valores actualizados de ε_i según la ecuación 1.4. Si la variación de los ε_i es mayor a un determinado umbral, se repite el procedimiento actualizando las proyecciones SOP en 1.12, si es menor al umbral se deja de iterar y se guarda la pose calculada.

1.2.5. POSIT para puntos coplanares

Como se mencionó anteriormente, el algoritmo POSIT no funciona en el caso en que los puntos puntos del modelo pertenecen a un mismo plano. Como los marcadores utilizados son planos, se buscó una versión de POSIT que resuelve el problema de la estimación de pose para este caso. El algoritmo fue escrito por *Denis Oberkampf*, *Daniel DeMenthon* y *Larry Davis* en [**?**].

Para entender cual es el problema de trabajar con puntos coplanares se explica la situación desde un punto de vista geométrico. Como se vio anteriormente

$$M_0M_i \cdot \mathbf{I} = x_i^{'} - x_0.$$

Esto quiere decir que si se toma que la base de **I** en M_0 , la punta del vector **I** se proyecta sobre el vector M_0M_i en un punto H_{xi}, entonces todas las posibles puntas del vector **I** se encuentran en el plano perpendicular a M_0M_i que pasa por el punto H_{xi}. Si se tuvieran 4 puntos no coplanares M_0, M_1, M_2 y M_3, el vector **I** quedaría determinado. La base de **I** estaría en M_0 y la punta estaría en la intersección de los planos perpendiculares a M_0M_1, M_0M_2 y M_0M_3 por los puntos H_{x1}, H_{x2} y H_{x3} respectivamente. Para este caso el sistema definido en 1.12 es de rango 3.

Figura 1.2: Configuración de puntos coplanares pertenecientes al plano D. Los planos perpendiculares que pasan por los puntos H_{x1} y H_{x2} se intersectan en un recta que pasa por el punto Q. Se puede ver que si hubiera un 4^{to} punto, el plano perpendicular correspondiente haría aparecer 2 rectas paralelas. Fuente: [**?**] .

Si los puntos son coplanares, los vectores M_0M_1, M_0M_2 y M_0M_3 son todos coplanares y los planos perpendiculares que pasan por los puntos H_{x1}, H_{x2} y H_{x3}, se intersectan todos un una línea o en dos líneas paralelas por lo tanto hay infinitas soluciones para el vector **I**. En este caso el sistema de ecuaciones 1.12 queda de rango 2. El vector solución que se obtiene al realizar la pseudo inversa de **A** es el que está a menor distancia de los planos, en la figura 1.2 es el vector $\mathbf{I}_0$. Esta la solución no es la solución al problema de los vectores de rotación, las soluciones se pueden expresar como

$$\begin{aligned} \mathbf{I} &= \mathbf{I}_0 + \lambda \mathbf{u} \\ \mathbf{J} &= \mathbf{J}_0 + \mu \mathbf{u} \end{aligned} \tag{1.16}$$

donde **u** es un versor perpendicular al plano de los puntos, $\mathbf{J}_0$ se calcula de manera análoga a $\mathbf{I}_0$ y λ y μ son las coordenadas de **I** y **J** según el versor **u**. Para encontrar las soluciones hay que calcular el versor **u** y los valores de λ y μ.

Como el vector **u** es perpendicular al plano de los puntos característicos se cumple $M_0M_i \cdot \mathbf{u} = 0$, se puede hallar entonces como la base del núcleo de la matriz **A**. En la práctica este vector se halla a partir de la descomposición *SVD* de la matriz **A**. La descomposición en valores singulares de la matriz **A** queda:

$$A = U\Sigma V^T \tag{1.17}$$

donde $U \in \Re^{n\times n}$ es ortogonal, $\Sigma \in \Re^{n\times 3}$ es diagonal, con los valores singulares en la diagonal y $V \in \Re^{3\times 3}$ es ortogonal. Como la matriz **A** es de rango 2, los dos primeros vectores columna de la matriz V corresponden a la base de todos los puntos que pertenecen al plano del modelo, mientras que el último vector de **V** es la base del núcleo de **A**, o sea el vector **u**. El cálculo **u** se realiza junto al calculo de la matriz **B**, ya que para ambos es necesario hacer la descomposición *SVD* de **A**.

Para calcular los valores de λ y μ se utilizan las condiciones de que **I** y **J** tienen que ser perpendiculares entre sí y del mismo largo. Como tienen que se perpendiculares se tiene que

$$\mathbf{I} \cdot \mathbf{J} = (\mathbf{I}_0 + \lambda \mathbf{u}) \cdot (\mathbf{J}_0 + \mu \mathbf{u}) = 0$$

entonces se tiene que

$$\lambda\mu = -\mathbf{I}_0 \cdot \mathbf{J}_0 \tag{1.18}$$

Como tienen que ser del mismo largo se tiene que

$$(\mathbf{I}_0 + \lambda \mathbf{u}) \cdot (\mathbf{I}_0 + \lambda \mathbf{u}) = (\mathbf{J}_0 + \mu \mathbf{u}) \cdot (\mathbf{J}_0 + \mu \mathbf{u}) \Leftrightarrow \lambda^2 - \mu^2 = \mathbf{J}_0^2 - \mathbf{I}_0^2 \tag{1.19}$$

Se define el número complejo $C = \lambda + i\mu$, si se eleva al cuadrado queda $C^2 = \lambda^2 - \mu^2 + i\lambda\mu$. Utilizando 1.18 y 1.19 se llega a que

$$C^2 = \mathbf{J}_0^2 - \mathbf{I}_0^2 - 2i\mathbf{I}_0 \cdot \mathbf{J}_0 \tag{1.20}$$

por lo que λ y μ pueden calcularse como las partes real e imaginaria del complejo C^2. Para hallar la raíces de C^2, se expresa en forma polar:

$$C^2 = [R, \Theta], \text{ donde}$$
$$R = \left(\left(\mathbf{J}_0^2 - \mathbf{I}_0^2 \right)^2 + 4 \left(\mathbf{I}_0 \cdot \mathbf{J}_0 \right)^2 \right)^{1/2}$$
$$\Theta = \arctan\left(\frac{-2\mathbf{I}_0 \cdot \mathbf{J}_0}{\mathbf{J}_0^2 - \mathbf{I}_0^2} \right), \text{ si } \mathbf{J}_0^2 - \mathbf{I}_0^2 > 0, \text{ y}$$
$$\Theta = \arctan\left(\frac{-2\mathbf{I}_0 \cdot \mathbf{J}_0}{\mathbf{J}_0^2 - \mathbf{I}_0^2} \right) + \pi, \text{ si } \mathbf{J}_0^2 - \mathbf{I}_0^2 < 0$$
$$\text{si } \mathbf{J}_0^2 - \mathbf{I}_0^2 = 0 \text{ se toma } \Theta = -signo\left(\mathbf{I}_0 \cdot \mathbf{J}_0 \right) \frac{\pi}{2}, \text{ y } R = \left| 2\mathbf{I}_0 \cdot \mathbf{J}_0 \right|$$

Se obtienen 2 raíces, $C = [\rho, \theta]$, y $C = [\rho, \theta + \pi]$, donde

$$\rho = \sqrt{R}, \text{ y } \theta = \frac{\Theta}{2}$$

como se mencionó anteriormente λ y μ son las partes real e imaginaria de C, por lo tanto

$$\lambda_1 = \rho \cos\theta, \qquad \mu_1 = \rho \sin\theta \tag{1.21a}$$
$$\lambda_2 = -\rho \cos\theta, \qquad \mu_2 = -\rho \sin\theta \tag{1.21b}$$

Esto quiere decir que se obtienen dos soluciones para $\mathbf{I}$ y $\mathbf{J}$

$$\mathbf{I}_1 = \mathbf{I}_0 + \rho \cos\theta \mathbf{u}, \qquad \mathbf{J}_1 = \mathbf{J}_0 + \rho \sin\theta \mathbf{u} \tag{1.22a}$$
$$\mathbf{I}_2 = \mathbf{I}_0 - \rho \cos\theta \mathbf{u}, \qquad \mathbf{J}_2 = \mathbf{J}_0 - \rho \sin\theta \mathbf{u} \tag{1.22b}$$

Como el vector $\mathbf{u}$ es perpendicular la plano del objeto, la solución encontrada en 1.22a es simétrica a 1.22b. Desde el punto de vista de la cámara, se puede ver que las dos posibles soluciones son aquellas que tienen la misma proyección SOP. Esto es equivalente a decir que para una misma proyección SOP hay dos posibles poses que que verifican las ecuaciones 1.12.

Figura 1.3: Dos objetos dando la misma proyección SOP. Fuente: [**?**].

Por lo tanto se toman las soluciones $(\mathbf{I}_1, \mathbf{J}_1)$ y $(\mathbf{I}_2, \mathbf{J}_2)$ y se calculan las poses. Como las dos poses son simétricas respecto a un plano paralelo al plano imagen, puede pasar que una pose de una solución en la que los los puntos del objeto queden ubicados detrás de la cámara. Por lo tanto previo a dar las dos soluciones como válidas hay que verificar esto.

En el caso en que las dos soluciones sean validas para todas las iteraciones, el número de poses posibles sería 2^n a lo largo de n iteraciones. En la práctica se manejan menos soluciones posibles. Se diferencian dos casos:

. Si se tiene que solo una de las dos primeras poses calculadas es válida, en las siguientes iteraciones se da mismo comportamiento, por lo que hay solo un camino a seguir.

. Si se tiene que las dos primeras poses calculadas son válidas, se abren dos posibles ramas. En la segunda iteración cada rama da lugar a dos nuevas poses, pero en este caso se toma la pose que da menos error de reproyección.

1.3. SoftPOSIT

1.4. POSIT Coplanar

(a)(b)

Figura 1.4: (a):Caso en el que solo una pose de las dos iniciales es coherente, también en las siguientes iteraciones solo una de las dos poses es posible, se tiene un única solución. (b): Caso en el que en cada paso hay dos posibilidades, se opta por la mejor pose(++ mejor pose, + peor pose) en cada rama. Fuente: [**?**].]

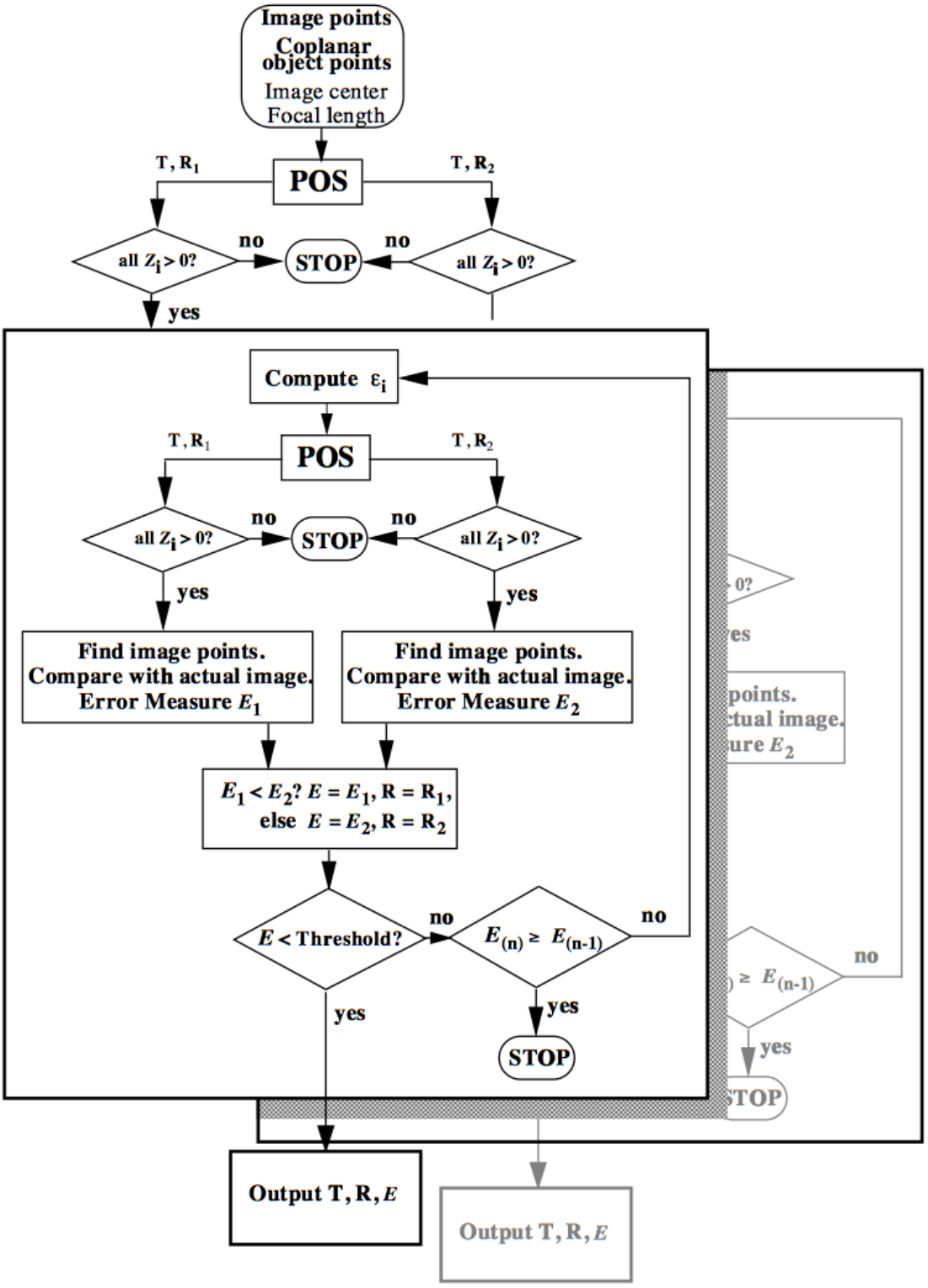

Figura 1.5: Diagrama de flujo del algoritmo para puntos coplanares, E es el error de reproyección, la condición $Z_i > 0$ verifica que los puntos reproyectados están por delante del plano imagen. Fuente: [**?**].]

[**?**].