

Un Algoritmo de Visión de Máquina para la Definición Automática de Superficies de Trabajo Planas para Tareas Repetitivas de un Robot Manipulador Industrial

Ing. Norma Yazmin Covarrubias Glez, UASLP
Ing. Felipe Antonio de la Peña Contreras, UASLP

Dr. Juan Antonio Cárdenas Galindo, UASLP
Dr. Emilio J. González Galván, UASLP

Centro de Investigación y Estudios de Posgrado,
Facultad de Ingeniería, UASLP

Ave. Dr. Manuel Nava No. 8, San Luis Potosí, S.L.P.
C.P.78290, San Luis Potosí, S.L.P., México.

[/yazmin960@yahoo.com](mailto:yazmin960@yahoo.com), felipep@uaslp.mx }

jjcardena@uaslp.mx, egonzale@uaslp.mx }

Resumen - En este trabajo se presenta un algoritmo para definir de forma automática una pieza de trabajo (o superficie) plana de un robot manipulador industrial. Se desarrolló para su uso una interfaz gráfica de usuario en código abierto bajo el sistema operativo Linux, de tal forma que el usuario no requiera conocimientos avanzados en programación de robots. Este algoritmo corresponde a un Modelo Lineal de Primer Orden para el mapeo (correspondencia) y segmentación de imágenes, tal y como es requerido para el desarrollo de un control basado en visión artificial de un robot manipulador industrial en una línea de producción continua.

I. Introducción

En las líneas de producción en las que existe un robot, se requieren una serie de sensores que garanticen la posición, dimensión y presencia de la pieza de trabajo con respecto al robot. Muchas veces estos sensores dificultan que se realice una tarea diferente, ya que un cambio en la línea de producción requiere cambiar la posición de los sensores utilizados. Ésto impide utilizar la característica de reprogramación propia de los robots. Buscando resolver este problema, se han añadido sensores de visión. Este tipo de sensores son fácilmente reconfigurables y otorgan mayor versatilidad en los procesos industriales.

En el Laboratorio de Robótica (LR) de la Facultad de Ingeniería de la UASLP se han implementado técnicas de control basadas en visión artificial aplicadas a robots industriales. Este artículo describe los trabajos recientes desarrollados en este laboratorio en el análisis de imágenes para identificar de forma

automática la pieza de trabajo del robot, usando técnicas para segmentación y correspondencia de imágenes.

En el LR se ha utilizado el método de Manipulación de Espacio de Cámara (CSM) [1] para desarrollar algoritmos de control para el posicionamiento y seguimiento de trayectorias de un robot soldador industrial sobre una superficie de trabajo [3]. El método de CSM es un enfoque libre de calibración para el control de robots manipuladores usando visión artificial. CSM es un método basado en estimación de lazo abierto que establece una relación matemática entre la apariencia de marcas visuales en el manipulador y la configuración interna de las articulaciones del robot [2]. Este método necesita por lo menos dos cámaras para realizar el control basado en visión.

La implementación mencionada en el párrafo anterior requirió desarrollar una interfaz gráfica de usuario (en *windows*) por medio de la cual se definen al inicio del proceso tanto los límites de la superficie de trabajo del robot como la trayectoria a seguir sobre la superficie definida [4] y sobre superficies arbitrarias [8]. Una vez definido lo anterior, la tarea asignada al robot puede ser repetida para un número cualquiera de piezas, con la restricción de que las piezas siguientes deben ser colocadas en la misma forma dentro del espacio de trabajo del robot.

En este trabajo se desarrolló un algoritmo para automatizar la tarea de identificación de la superficie de trabajo. Se programó para esto una nueva GUI sobre

internet y el control se desarrolló en el sistema operativo Linux. En esta implementación, el usuario únicamente selecciona un punto al inicio del proceso. La restricción es que el punto seleccionado pertenezca a cada una de las nuevas piezas de trabajo. Estas nuevas piezas podrán llegar al espacio de trabajo del robot con diferente geometría y orientación.

Para permitir que el usuario seleccione el punto inicial que permitirá definir la superficie de trabajo, se desarrolló una interfaz gráfica en Linux.

II. Ubicación de Objetivos Seleccionados en Espacio de Cámara en el Espacio de Trabajo

Una vez que el usuario selecciona desde la terminal el punto inicial, una unidad de dos grados de libertad como la que se muestra en la Fig. 4, conocida como unidad *Pan-Tilt* posiciona un láser puntual sobre la superficie de trabajo en el punto especificado [9].

La unidad de *Pan-Tilt* se compone de dos motores a pasos controlados a través del puerto paralelo de la computadora. Debido a que es necesario conocer la posición en espacio de cámara del punto de luz láser proyectada sobre la superficie, se utiliza una expansión en series de Taylor para definir una relación entre el cambio de posición angular de cada uno de los ejes y el cambio de posición del punto de láser tal y como es registrada en espacio de cámara, esta expansión se define como [6]:

$$\begin{aligned} x_c &= x_{c_0} + \frac{\partial x_c}{\partial \theta_1} \Delta \theta_1 + \frac{\partial x_c}{\partial \theta_2} \Delta \theta_2 + T.O.S. \\ y_c &= y_{c_0} + \frac{\partial y_c}{\partial \theta_1} \Delta \theta_1 + \frac{\partial y_c}{\partial \theta_2} \Delta \theta_2 + T.O.S. \end{aligned} \quad (1)$$

donde (x_{c_0}, y_{c_0}) son las coordenadas de partida (en espacio de cámara) del punto de láser, (x_c, y_c) son las coordenadas en espacio de cámara en donde se desea proyectar el punto de láser y $(\Delta \theta_1, \Delta \theta_2)$ son los pasos de rotación respectivos a cada eje. Las siglas T.O.S. se refieren a los términos de orden superior. Es posible reescribir la ecuación anterior utilizando una matriz del Jacobiano para englobar las derivadas parciales y eliminando los términos de orden superior.

Es decir, si linealizamos las ecuaciones descritas en (1), tenemos:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = [J][\Delta \theta] = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \end{bmatrix} \quad (2)$$

donde los parámetros J_{11} , J_{12} , J_{21} y J_{22} componen la matriz del Jacobiano y se definen de la siguiente manera:

$$\begin{aligned} J_{11} &= \frac{\partial x_c}{\partial \theta_1} \approx \frac{x_{c_1} - x_{c_0}}{\Delta \theta_1} \\ J_{12} &= \frac{\partial x_c}{\partial \theta_2} \approx \frac{x_{c_2} - x_{c_0}}{\Delta \theta_2} \\ J_{21} &= \frac{\partial y_c}{\partial \theta_1} \approx \frac{y_{c_1} - y_{c_0}}{\Delta \theta_1} \\ J_{22} &= \frac{\partial y_c}{\partial \theta_2} \approx \frac{y_{c_2} - y_{c_0}}{\Delta \theta_2} \end{aligned} \quad (3)$$

Para mover el punto de láser montado en la unidad de *Pan-Tilt* a la posición deseada, se deben inicializar los parámetros J_{11} , J_{12} , J_{21} y J_{22} definidos en (3). Para inicializar los parámetros anteriores, se toma una imagen del punto láser incidiendo sobre la superficie de trabajo en algún punto determinado y se obtiene la coordenada (en espacio de cámara) del centroide aproximado de dicho punto; posteriormente, se mueve la unidad hacia la izquierda un cierto ángulo, se enciende el punto láser para adquirir una nueva imagen y se regresa la unidad *Pan-Tilt* a su posición inicial. Después se mueve la unidad hacia abajo y se repite el procedimiento anterior. Con las coordenadas del punto inicial y los dos puntos posteriores, se inicializan los parámetros mencionados.

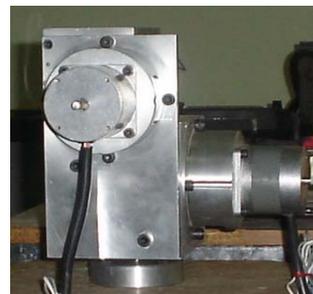


Fig. 4. Unidad de dos grados de libertad

Con los parámetros seleccionados, y conocida la coordenada a la que debe llegar el punto de láser, por medio de la ecuación (2) se determinan los ángulos de cada grado de libertad de la unidad de *Pan-Tilt* necesarios para posicionar la luz láser en el punto determinado. El proceso por medio del cual la unidad de *Pan-Tilt* se desplaza desde su posición inicial hacia

el punto seleccionado por el usuario es iterativo debido a la linealización realizada para obtener la ecuación (2). Cuando la unidad de *Pan-Tilt* converge al punto deseado, realiza dos desplazamientos más (cercanos al punto definido por el usuario), estos puntos que describen un triángulo son suficientes para definir un plano.

III. Modelo Lineal para el mapeo de puntos en la superficie de trabajo

Para realizar la definición de la superficie de trabajo, se utilizan dos cámaras CCD colocadas en diferente posición respecto a la pieza de trabajo. Estas cámaras son independientes de la cámara de "selección" con la cual se envía la imagen a la GUI.

A fin de obtener información que permita caracterizar la superficie de trabajo (Fig. 5a) sin dejar marcas permanentes en ella, se proyectan sobre esta superficie marcas visuales (Fig. 5b). Estas marcas se generan por medio de una matriz de luces de láser que también esta montada en la unidad de *Pan-Tilt*. De esta forma las luces se proyectan varias veces sobre la superficie formando una "malla" y se almacena también el número de toma al que pertenece.

Por medio de técnicas de diferencia de imágenes, se elimina el entorno de la imagen y así se obtiene una imagen final que contiene únicamente información de los puntos de láser sobre la superficie de interés (Fig. 5c). Esta información se refiere a las coordenadas en espacio de cámara de los centroides aproximados de cada uno de los puntos en cada cámara. Se tiene ahora, un conjunto de coordenadas de los mismos puntos sobre la superficie vistos desde dos ángulos diferentes. Para lograr la definición de la superficie es necesario realizar la correspondencia de cada uno de los puntos en ambas cámaras.

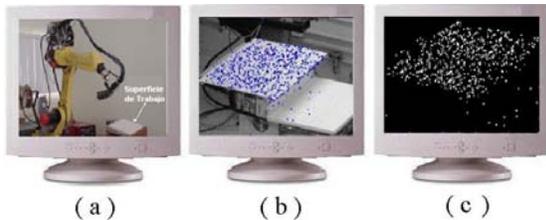


Fig. 5 Extracción de características por medio de diferencia de imágenes y puntos láser

Las piezas que se utilizaron cuentan con una superficie plana y que cada pieza aparece en las dos cámaras participantes, es posible utilizar un Modelo Lineal de Primer Orden para realizar el mapeo de

puntos entre imágenes usando la función (4). Tal como lo define Robinson [7]:

$$\begin{aligned} x_m &= A_1 x_n + A_2 y_n + A_3 \\ y_m &= A_4 x_n + A_5 y_n + A_6 \end{aligned} \quad (4)$$

donde (x_n, y_n) representan las coordenadas de los puntos en cámara 1 y (x_m, y_m) los puntos en cámara 2.

A. Demostración del Modelo Lineal

Este mapeo es consistente con el modelo ortográfico de cámara y una superficie plana.

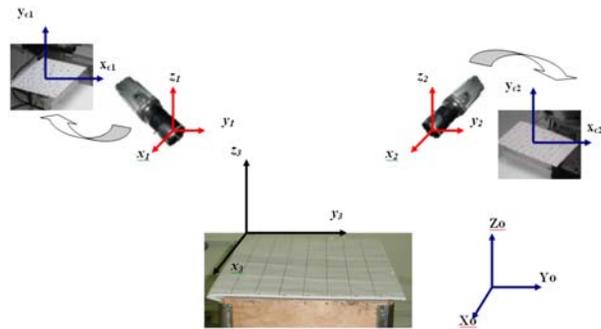


Fig. 6 Sistemas coordenados en el mapeo de superficies

De acuerdo a la figura anterior es posible definir las siguientes matrices de transformación homogénea:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = T_{12} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} \quad (5)$$

Las matrices anteriores se representan como una función de los parámetros de *Euler* [6], tal como se representa la matriz T_{12} :

$$T_{12} = \begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2 e_3 + e_1 e_4) & 2(e_2 e_4 - e_1 e_3) & X_0 \\ 2(e_2 e_3 - e_1 e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3 e_4 + e_1 e_2) & Y_0 \\ 2(e_2 e_4 + e_1 e_3) & 2(e_3 e_4 - e_1 e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

donde los parámetros de *Euler* cumplen con $e_1^2 + e_2^2 + e_3^2 + e_4^2 = 1$ y las coordenadas (X_0, Y_0, Z_0) representan el origen del sistema coordinado relativo a la cámara. Haciendo uso de una proyección ortográfica, las coordenadas en el espacio de cámara (x_c, y_c) de un punto en el espacio físico pueden escribirse como:

$$\begin{aligned} x_{ci} &= f \frac{X_i}{Z_i} = Fx(x, y, z; P) \\ y_{ci} &= f \frac{Y_i}{Z_i} = Fy(x, y, z; P) \end{aligned} \quad (7)$$

donde $P = [e_1, \dots, e_4, X_0, Y_0, Z_0, f]^T$ (f es la longitud focal de la cámara) [10].

Se realiza una expansión de series de Taylor de las ecuaciones definidas en (7) y se considera solo la primer aproximación, se obtiene:

$$\begin{aligned} x_{ci} &= (C_1^2 + C_2^2 - C_3^2 - C_4^2)x_1 + 2(C_2C_3 + C_1C_4)y_1 + 2(C_2C_4 - C_1C_3)z_1 + C_5 \\ y_{ci} &= 2(C_1C_3 - C_1C_4)x_1 + (C_1^2 - C_2^2 + C_3^2 - C_4^2)y_1 + 2(C_3C_4 + C_1C_2)z_1 + C_6 \end{aligned} \quad (8)$$

Los términos C_1, C_2, \dots, C_6 son llamados *parámetros de visión*. Las ecuaciones descritas en (8) definen la relación que existe entre la localización de un punto en el espacio físico tridimensional y su correspondiente localización en el espacio de cámara.

Sustituyendo (6) en (5) y sustituyendo en éstas las ecuaciones de (8) se tiene para x_{c1} :

$$\begin{aligned} x_{c1} &= (C_1^2 + C_2^2 - C_3^2 - C_4^2) \{ (e_1^2 + e_2^2 - e_3^2 - e_4^2)x_1 + 2(e_2e_3 + e_1e_4)y_1 + 2(e_2e_4 - e_1e_3)z_1 + X_0 \} + \\ &2(C_2C_3 + C_1C_4) \{ 2(e_2e_3 - e_1e_4)x_1 + (e_1^2 - e_2^2 + e_3^2 - e_4^2)y_1 + 2(e_2e_4 + e_1e_3)z_1 + Y_0 \} + \\ &2(C_2C_4 - C_1C_3) \{ 2(e_2e_4 + e_1e_3)x_1 + 2(e_2e_4 - e_1e_3)y_1 + (e_1^2 - e_2^2 - e_3^2 + e_4^2)z_1 + Z_0 \} \end{aligned} \quad (9)$$

agrupando términos se tiene:

$$x_{c1} = ax_2 + by_2 + cz_2 + d \quad (10)$$

En un modelo ortográfico de cámara las siguientes relaciones son válidas:

$$\begin{aligned} x_{c1} &\propto x_1 \\ y_{c1} &\propto y_1 \end{aligned} \quad (11)$$

Sustituyendo (11) en (10):

$$\begin{aligned} x_{c1} &= ak_1x_{c2} + bk_2y_{c2} + cz_2 + d = ak_1x_{c2} + bk_2y_{c2} + \alpha \\ x_{c1} &= k(ax_{c2} + by_{c2} + \beta) = Ax_{c2} + By_{c2} + C \end{aligned} \quad (12)$$

Resolviendo de igual forma para y_{c1} se tiene:

$$\begin{aligned} x_{c1} &= Ax_{c2} + By_{c2} + C \\ y_{c1} &= Dx_{c2} + Ey_{c2} + F \end{aligned} \quad (13)$$

Las ecuaciones descritas por las ecuaciones anteriores (13) son equivalentes al modelo lineal utilizado (4).

B. Secuencia de Mapeo

Utilizando las coordenadas de los centroides referentes a los tres puntos que forman un triángulo (el

punto definido por el usuario y los dos puntos generados automáticamente) se inicializan los parámetros $[A_1 \dots A_6]$ del modelo lineal (4) de acuerdo a la siguiente ecuación:

$$A = (a^T a)^{-1} a^T b \quad (14)$$

donde:

$$a = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \quad y \quad b = \begin{bmatrix} x_1' & y_1' & 1 \\ x_2' & y_2' & 1 \\ x_3' & y_3' & 1 \end{bmatrix}$$

Con los parámetros inicializados, se inicia el proceso de correspondencia de imágenes iniciando en el punto definido por el usuario y definiendo una región de 30x30 píxeles alrededor de éste llamada *región de búsqueda*. Cada una de las coordenadas de los puntos de la cámara 1 dentro de la región se “mapean” hacia los puntos de la cámara 2 utilizando la ecuación (4).

Se tiene entonces para cada punto de cámara 1 un punto “predicho” en cámara 2, debido a la no linealidad del sistema de adquisición de cada imagen (distorsión ocasionada por los lentes de las cámaras, ruido en la imagen, pérdida de información al momento de la digitalización, etc.), el punto predicho no tiene las mismas coordenadas que el punto correspondiente en la cámara 2 y existe entonces más de un posible punto correspondiente, cada uno de estos puntos se denominan “punto candidato”. Por ello se recorren los puntos más cercanos en la cámara 2 (típicamente 20) buscando el punto correspondiente al predicho, el criterio utilizado para determinar cuál de los puntos en cámara 2 corresponde al punto de la cámara 1 es que la distancia entre el punto predicho y el punto candidato (6 sea menor que un valor de umbral T definido de manera experimental. En esta implementación se definió T= 10.

$$r = \sqrt{(x_p - X_d)^2 + (y_p - Y_d)^2} \quad (15)$$

donde r es el radio o la distancia entre el punto predicho (x_p, y_p) y el punto candidato (X_d, Y_d) .

Sí, además de que r sea menor que el valor de umbral T, los puntos pertenecen al mismo número de toma y existe un único punto coincidente, entonces se determina que el punto candidato corresponde al punto en cámara 1. Cada vez que un punto se mapea, se agrega al modelo y se recalculan los parámetros A.

Una vez que se han recorrido todos los puntos (en cámara 1) dentro de la región inicial de búsqueda, se define una nueva región de la misma dimensión pero

traslapada con la anterior, de tal forma que con los puntos compartidos por ambas regiones se reinician los parámetros A por medio de (4). Esta redefinición de regiones de búsqueda se realiza de forma espiral (fig. 7) hasta que ya no se encuentran puntos en las regiones.

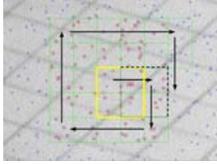


Fig. 7 Secuencia de búsqueda de forma espiral

El modelo elimina los puntos que se encuentran fuera de la superficie de interés debido a que el valor de r definido en la ecuación (15) se vuelve grande (mayor de 100) debido a que esos puntos no corresponden al modelo definido en la ecuación (4). De esta manera se logra identificar el dominio correspondiente a la superficie plana. Esto permite segmentar la imagen e identificar automáticamente la superficie de trabajo (Fig. 8).

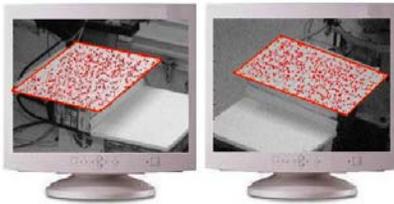


Fig. 8 Segmentación de la imagen y definición de la superficie de trabajo

IV. Resultados y Conclusiones

Para las pruebas experimentales de validación se proyectaron un total de 926 puntos de láser en la cámara 1 y 1216 puntos de láser en la cámara 2. Se mapearon correctamente el 93.4% de los puntos sobre la superficie de interés.

Como se esperaba a partir de la formulación matemática, en las pruebas realizadas se encontró que el Modelo Lineal de mapeo es capaz de realizar exitosamente la correspondencia de los puntos que inciden en la misma superficie física. Este mapeo lineal descarta los puntos fuera de la superficie de interés, de tal forma que es posible segmentar la imagen y por lo tanto definir de forma automática la superficie. Lo anterior permite que un robot manipulador industrial realice tareas repetitivas en una línea de producción.

A. Trabajo Futuro

Integrar los algoritmos de CSM para posicionar y orientar el efector final de un robot manipulador sobre una superficie definida por medio del modelo lineal mencionado. Realizar una GUI remota con intercambio de datos a través de internet para que un usuario desde cualquier PC conectada al internet pueda mediante la selección de un punto sobre la imagen iniciar el proceso de definición de la superficie de interés.

B. Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca otorgada, CUDI-CONACYT y al Fondo de Apoyo a la Investigación de las UASLP (FAI) por el apoyo recibido para la compra de equipo a través de convenio CO3-FAI-11-10.45.

V. Bibliografía

- [1] S.B. Skaar, W. H. Brockman and R. Hason. Camera Space Manipulation. International Journal of Robotic Research, 6(4):20-32, 1987
- [2] Skaar SB, Seelinger MJ, Robinson ML, González Galvan EJ. Means and method of robot control relative to an arbitrary surface using camera-space manipulation. United States Patent #6,304,050.2001
- [3] Rolando Cruz. Desarrollo de un algoritmo de control basado en visión para la ejecución de tareas robotizadas sobre superficies arbitrarias curvas. Tesis de Maestría. UASLP-FI. México 2002.
- [4] Ambrocio Loredó. Desarrollo de un algoritmo de control basado en visión para el seguimiento óptimo de trayectorias. Tesis de Maestría. UASLP-FI. México 2002.
- [5] E. González, A. Loredó, J. Cervantes "An optimal path-tracking algorithm for unstructured environments based on uncalibrated vision," International Conference on Robotics and Automation Proceedings, (Barcelona), 2005.
- [6] Miguimi Lukito. A point-and-click palletizing/depalletizing control system using camera space manipulation. Master Thesis. Department of Aerospace and Mechanical Engineering. Notre Dame, In. 2001.
- [7] Matthew L. Robinson. A structured lighting approach to Image Analysis for robotic applications using camera space manipulation. PhD Thesis. Department of Aerospace and Mechanical Engineering. Notre Dame, In. 2002
- [8] E. González, A. Loredó, J. Cervantes, Laborico A. "An optimal vision-based, path tracking algorithm for unstructured environments," Proceeding of Symposium on Robotics and Automation Proceedings, (México), 2004.
- [9] E. González, E. Rivera, R. D. González, O. Garay, V. Miranda. "A graphical user interface for industrial robot programming in non-repetitive tasks" In Human friendly Mechatronics. Arai, E. Arai T. and Tanako M. editors. Elsevier Science B.V. 2001 ISBN 0444506497 pp.67-72
- [10] E. González, S. Skaar, U. Korde, W. Chen. "Application of a Precision-Enhancing Measure in 3D Rigid-Body Positioning Using Camera Space Manipulation". The International Journal of Robotics Research. 16[2], pp. 240-257. MIT, 1997