

Aplicación de Técnicas de Visión al Ensamble con Robots Manipuladores

Mejía Rodríguez Luis Felipe¹, Pedraza Ortega Jesús Carlos², Santoyo Rodríguez Joaquín¹

¹Universidad Tecnológica de Querétaro (UTEQ), ²Centro de Ingeniería y Desarrollo Industrial (CIDESI)

Resumen

El propósito de este reporte es presentar la metodología seguida para la aplicación de visión artificial en la generación de programas para un robot manipulador, que permitan la realización de operaciones de ensamble básicas.

Para las operaciones de ensamble se consideran placas perforadas con huecos para colocar piezas, que pueden tener sección circular o sección cuadrada, y tomadas de alimentadores en posiciones predeterminadas.

Una placa particular es colocada frente al manipulador, y mediante una cámara, ésta y sus características son reconocidas.

El robot manipulador considerado es un Pegasus II de Amatrol, un robot educativo con calidad industrial.

Las herramientas utilizadas son Matlab 7 y Visual C++ 6.

Palabras clave: Visión artificial. Robot manipulador. Matlab.

1. Introducción

La visión artificial ha sido empleada por casi 4 décadas, pero desde hace algunos años es una posibilidad real para muchas aplicaciones, debido al incremento del poder de cómputo de los procesadores recientes y a la disponibilidad de cámaras de bajo costo de calidad razonable y conexión directa a la computadora.

Existe software comercial disponible para realizar tareas de adquisición y procesamiento de imágenes de fácil uso, como lo es el caso de Matlab, que mediante sus cajas de herramientas pone a disposición del desarrollador un conjunto amplio de funciones para adquirir, convertir a escala de grises, binarizar, determinar componentes, calcular parámetros de los componentes, etc.

Para una situación de ensamble básica, como lo es el montaje de piezas con forma geométrica regular, tales como cilindros y cubos, es posible reconocer las perforaciones para inserción en bases y calcular sus posiciones.

Los robots manipuladores son rápidos y precisos, y su programación con coordenadas cartesianas permite aprovechar los parámetros extraídos de un montaje requerido para la generación automática del programa de ensamble requerido.

Existen ventajas importantes de la generación automática de programas de cualquier tipo, en particular para robots manipuladores: los tiempos de desarrollo se reducen de forma considerable y la prueba y depuración es más rápida y simple. El propósito de este trabajo es mostrar una metodología seguida para generar de forma automática los programas para robot que permitan la realización de operaciones básicas de ensamble, a partir de una situación reconocida utilizando visión.

La situación de ensamble aquí abordada asemeja a la colocación de tornillos o puntos de soldadura en ensambles industriales, por lo que los resultados que puedan obtenerse de inmediato pueden ser aprovechados en el desarrollo de aplicaciones industriales, con beneficio económico directo en distintos sectores.

Algunos trabajos en esta misma área que han sido consultados son [1] y [2].

2. Metodología

La metodología seguida en este trabajo se muestra en el siguiente diagrama. Las etapas se detallan en las subsecciones.

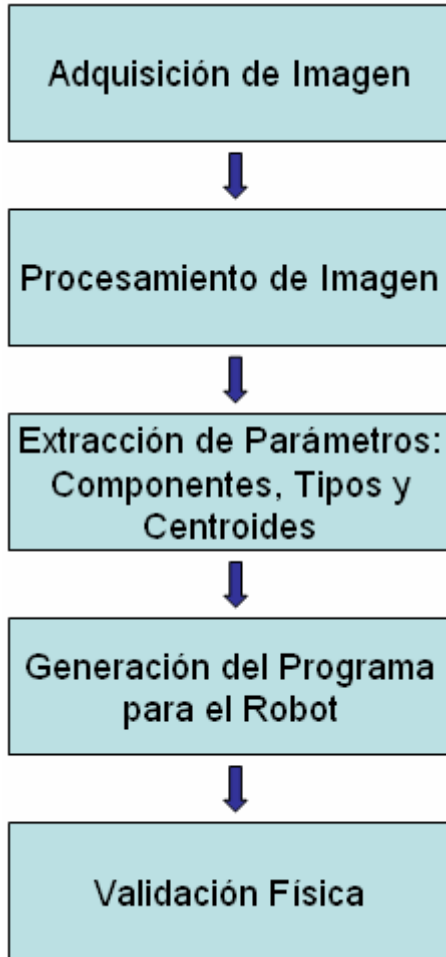


Figura 1. Estructura general del sistema.

2.1 Adquisición de la imagen

El sistema de visión empleado es una básico, y consiste esencialmente de una computadora personal con procesador de doble núcleo y una cámara web con capacidad para adquirir imágenes con una resolución de 640 x 480 píxeles y 24 bits. Un aspecto de suma importancia en las aplicaciones de visión artificial es la iluminación, que a pesar de que no se ha controlado, sí se puede pensar que se han tenido condiciones estandarizadas, ya que las adquisiciones se han realizado en horario diurno con iluminación aproximadamente constante.

2.2 Procesamiento de la imagen

El software empleado para realizar tanto la adquisición como el procesamiento de la imagen es Matlab, que proporciona cajas de herramientas completas para estas tareas, con la facilidad adicional de usar matrices para almacenar las imágenes, permitiendo de esta forma el desarrollo de nuevos algoritmos y su experimentación.

Las imágenes típicas empleadas son como la siguiente.

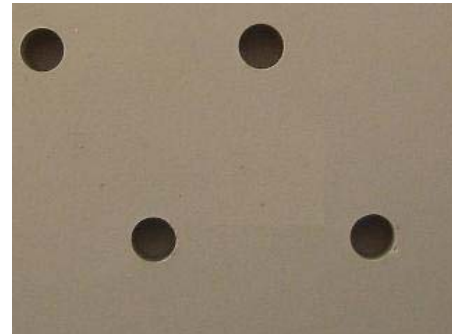


Figura 2. Una placa de prueba (el color real es gris).

El primer paso consiste de la conversión de la imagen a escalas de gris (8 bits), y una vez convertida a este formato se realiza su binarización, donde a partir de un umbral definido se ajustan todos los píxeles exclusivamente a blanco o negro. El resultado de las conversiones de la imagen previa se muestra a continuación.



Figura 3. Una placa binarizada.

2.3 Extracción de parámetros: componentes, tipos y centroides

El tener la imagen binaria nos da múltiples ventajas, ya que podemos determinar los componentes presentes en la imagen y sus parámetros principales, tales como área, centroide, longitudes máximas horizontal y vertical, contorno, etc.

Para reconocer los componentes se hace un análisis de la vecindad de cada punto blanco, utilizando conexiones-4 o conexiones-8, y codificando con valores distintos (consecutivos) a cada grupo de píxeles contiguos, que definen un componente.

Posteriormente se determinan los parámetros de los componentes. Para este trabajo se consideran los centroides y las áreas.

El cálculo básico se realiza a nivel de pixel, lo que nos da las fórmulas siguientes para el cálculo de los centroides.

$$x = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

$$y = \frac{\sum_{i=1}^n y_i}{n} \quad (2)$$

Para el área basta con contar los pixeles presentes en cada componente.

Las únicas secciones que se considera pueden estar presentes en las placas son circulares y cuadradas. Si se considera que el diámetro de los círculos y los lados de los cuadrados son iguales, las áreas nos permiten determinar el tipo de sección particular de un componente.

En este punto Matlab genera un archivo de texto con la información obtenida, que será aprovechado por programas escritos en C.

2.4 Generación del programa para el robot

Una vez almacenada la información de componentes de una placa en el archivo de texto, el punto siguiente consiste de la generación del programa asociado de robot para realizar el ensamble. La razón de hacer esta generación en C y no directamente en Matlab es porque se consideró también el desarrollar un modelo virtual del robot para poder validar las trayectorias.

Como se consideran posiciones fijas para la cámara y para la placa de ensamble, la información de pixeles puede ser aprovechada para definir posiciones en milímetros de los componentes.

Un ejemplo de la generación que se logra con la aplicación es el siguiente.

```
PMOVE <-3000,100,2000,-1221,0>
PMOVE <-3000,100,800,-942,0>
GRASP
PMOVE <-3000,100,2000,-1221,0>
```

```
PMOVE <-545,3250,2000,-1184,0>
PMOVE <-545,3250,800,-913,0>
RELEASE
PMOVE <-545,3250,2000,-1184,0>
```

HALT

El significado del código es:

1. Ir al alimentador.
2. Bajar por un cilindro.
3. Sujetar cilindro
4. Subir.
5. Desplazarse a posición superior a punto de inserción.
6. Bajar.
7. Soltar cilindro.
8. Subir.

2.5 Validación física

Para la validación física se consideran componentes con tolerancias grandes, por lo que se observa que las trayectorias que se siguen corresponden de forma adecuada a lo deseado.

Un problema que se presenta con frecuencia son los desajustes en el robot, debido a traslados del mismo, mal uso o exceso de uso, mala fabricación, etc.

Una posibilidad es caracterizar el error del robot para cada punto en una malla, y aprovechar esta información a manera de reglas para ajustar las posiciones generadas para el robot.

3. Análisis de resultados

Se han logrado algunos resultados interesantes hasta el momento. Se cuenta con código en C y Matlab para la realización de operaciones de ensamble básicas, con una experimentación inicial que arroja resultados prometedores.

Se cuenta también con un modelo virtual, basado en la cinemática del manipulador, que permite validar trayectorias.

4. Conclusiones

La metodología seguida ha sido simple, y sin embargo los resultados obtenidos son adecuados. De este trabajo se desprenden distintas posibilidades.

La primera es la posibilidad de sustitución del robot por uno industrial, de alguna de las marcas principales, lo que de inmediato ampliaría el impacto del proyecto.

Otra posibilidad es desarrollar nuevas aplicaciones que busquen realizar operaciones de ensamble más complejas, con secciones arbitrarias y que requieren de técnicas de reconocimiento más avanzadas.

Es posible también el desarrollo de proyectos con dos cámaras, que proporcionen información de profundidad, permitiendo lograr ensambles más generales.

Una categoría de especial interés para los autores es la robótica móvil guiada por visión, y proyectos futuros estarán encaminados en esta dirección.

Referencias

- [1] Soto Cajiga J. A., Vargas Soto J. E., Pedraza Ortega J. C. "Generación de trayectorias para un robot manipulador utilizando procesamiento de imágenes y splines", Segundo Congreso Internacional de Ingeniería, Querétaro, Qro., México, 2006.
- [2] Viramontes Reyna, J. L., Pedraza Ortega, J. C., González Galván, E. J. "Vision-based manipulator control using a monocular vision system", 5th International Symposium on Robotics and Automation, San Miguel Regla, Hidalgo, México, 2006.
- [3] Gamiño M., Pedraza J. C., Ramos J. M., Gorrostieta E. "Matlab-C++ interface for a flexible arm manipulator simulation using multi-language techniques", Proceedings of the fifth Mexican International Conference on Artificial Intelligence (MICAI'06), México, 2006.
- [4] Myler H., Weeks A. "Computer imaging recipes in C", Prentice-Hall, USA, 1993.
- [5] Gonzalez, Rafael C., Woods, Richard E.: Digital image processing, 2nd edition. Prentice-Hall, USA, 2002.
- [6] Barrientos A., Balaguer C.: Fundamentos de robótica, 1a edición. McGraw-Hill, España, 1997.