

## **Sistema de planificación de trayectorias utilizando visión artificial**

Jorge Cerezo-Sánchez<sup>1</sup>, Griselda Saldaña-González<sup>1</sup>, Mario M. Bustillo-Díaz<sup>2</sup>,  
Apolonio Ata-Pérez<sup>2</sup>

<sup>1</sup> Universidad Tecnológica de Puebla, División de Mecatrónica,  
Puebla, México

<sup>2</sup> Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación,  
Puebla, México

{Jorge.cerezo, Griselda.saldana}@utpuebla.edu.mx, {bustillo, apolonio}@cs.buap.mx

**Resumen.** En este trabajo se presenta un sistema de visión artificial para el análisis de la trayectoria de un objeto, lo que permite resolver un laberinto robotizado analizando las rutas de un objeto móvil en el espacio. El prototipo consta de un laberinto que tiene solución al centro, de manera que independientemente de donde se coloque un objeto móvil, éste se reconoce y se mapea a un sistema coordenado. En función de la posición, un sistema de control de lazo cerrado se encarga de que el robot realice acciones que permiten la solución de dicho laberinto. El sistema cuenta con una interface de usuario realizada en LabVIEW desde donde el usuario puede proporcionar parámetros y visualizar el proceso. La PC se conecta con un PIC que recibe instrucciones, las interpreta y traduce posicionando los servomotores que permiten el movimiento del laberinto para su solución.

**Palabras clave:** Clasificación de imágenes, LabVIEW, sistema de visión, seguimiento de trayectoria, reconocimiento de patrones.

### **Path Planning System Using Artificial Vision**

**Abstract.** In this paper an artificial vision system for the analysis of the trajectory of an object is presented, which allows to solve a robotic labyrinth analyzing the routes of a moving object in the space. The prototype consists of a labyrinth with a unique solution at the center, so that regardless the position where a mobile object is placed, it is recognized and mapped to a coordinate system. Depending on the position, a closed loop control system ensures that the robot performs actions that allow the solution of the labyrinth. The system has a user interface made in LabVIEW from where the user can provide parameters and visualize the process. The PC connects with a PIC that receives instructions, interprets and translates them to position the servomotors that allow the movement of the labyrinth for its solution.

**Keywords:** Images classification, LabVIEW, vision system, path planning, pattern recognition.

## **1. Introducción**

Uno de los procesos más importantes en las aplicaciones de visión, es la localización de un objeto de interés dentro del campo de visión de la cámara, una tarea que se puede lograr mediante el reconocimiento de patrones [1-2].

El reconocimiento de patrones es un método para encontrar regiones en una imagen en escala de grises que coincidan con un patrón de referencia. A menudo se utiliza para localizar las posiciones del patrón, o rasgos característicos de un objeto en una imagen. Las posiciones pueden utilizarse para calcular longitud, ángulos, y otros parámetros. Como resultado, el reconocimiento de patrones se ha usado ampliamente en diversas aplicaciones tales como alineación, medición, e inspección [3-4]. El reconocimiento de patrones tiene una ventaja sobre el análisis de partículas o la detección de bordes debido a que la búsqueda del patrón no se basa en distinguir el brillo del objeto en comparación con el fondo de la imagen.

El objetivo de este trabajo consiste en el diseño e implementación de un sistema automático de reconocimiento de objetos por visión aplicado a la solución de un laberinto utilizando LabVIEW [5-6] y las herramientas de visión IMAQ. El toolbox de IMAQ ofrece un conjunto muy completo de funciones para adquisición y procesamiento digital de imágenes que mejoran la eficiencia de los proyectos y reducen el esfuerzo de programación de los usuarios obteniendo mejores resultados en menos tiempo. El sistema propuesto presenta la ventaja de realizar una inspección sin necesidad de entrar en contacto con el producto, alcanzando un alto nivel de flexibilidad y repetitividad.

Como primer punto en este trabajo, en la sección 2 se menciona la herramienta IMAQ de LabVIEW y las funciones que permiten realizar el reconocimiento de patrones, en la sección 3 se abordará una descripción del sistema desarrollado, incluyendo la parte de hardware y software del prototipo, a continuación en la sección 4, se discutirán brevemente algunas pruebas realizadas al sistema y los resultados obtenidos, para finalmente presentar las conclusiones en la sección 5.

## **2. Módulos de visión de LabVIEW**

LabVIEW soporta las técnicas de procesamiento de imágenes con la ayuda de bloques funcionales divididos en dos categorías, de bajo y alto nivel. El asistente de visión IMAQ es un módulo de alto nivel que permite realizar las funciones de visión de cinco dominios, procesamiento de imágenes, procesamiento de color, procesamiento en escala de grises, procesamiento binario, visión artificial e identificación. Cada una de estas funciones de procesamiento contiene otras funciones entre las cuales se incluyen histogramas, reconocimiento de patrones, detección de bordes, seguimiento de objetos, análisis de contornos, detección de forma, análisis de partículas, filtros, detección de textura, segmentación, entre otras.

El reconocimiento de patrones localiza rápidamente regiones en imágenes en escala de grises que corresponden a un patrón conocido. Este proceso puede implementarse en dos etapas. La primera es la de aprendizaje, mientras que el reconocimiento es la segunda. El valor de escala de grises se extrae del patrón proporcionado por el usuario

durante la etapa de aprendizaje. El algoritmo organiza y almacena la información en una forma que facilita la búsqueda en la imagen a inspeccionar.

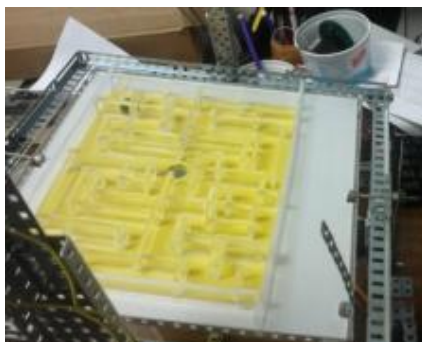
En la etapa de reconocimiento, el algoritmo extrae el valor de gris de la imagen a inspeccionar, y encuentra coincidencias localizando las regiones donde se observa la más alta correlación con la información aprendida del patrón.

Existen varios algoritmos implementados para realizar el reconocimiento de patrones, entre los que se incluye la correlación cruzada normalizada, el reconocimiento piramidal, el método de valor de gris y el método del gradiente.

En la implementación del sistema, se utilizó el método de Gray Scale Value Pyramid configurado con coincidencia para un solo objeto, y un score mínimo de 700.

### **3. Descripción del sistema**

El presente proyecto consiste en la implementación de un laberinto, el cual se soluciona empleando técnicas de visión. El sistema cuenta con una base cuadrada donde se ha implementado un laberinto de acrílico, con un ángulo de  $90^\circ$  en las esquinas y que tiene una única solución al centro. Esta base está montada sobre una estructura metálica que puede moverse en la dirección  $x$  e  $y$ , con referencia al plano que forma la base empleando servomotores que son controlados por microcontroladores PIC.



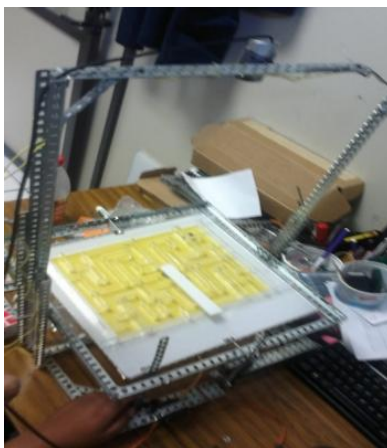
**Fig. 1.** Laberinto con solución al centro.

Para la implementación del sistema de visión se utiliza una cámara web ubicada en el plano  $z$ , calibrada por mediciones con respecto de las esquinas del laberinto. Dicha cámara se encarga de identificar una canica de color negro para determinar el siguiente movimiento que debe realizarse para hacer que la canica llegue a un orificio central donde se resuelve el laberinto.

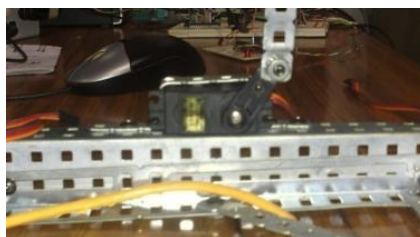
#### **3.1. Desarrollo mecánico**

El laberinto se encuentra montado en una estructura metálica elaborada con varillas de aluminio, la cual cuenta con servomotores de 15 Kg. Conectados a flechas de 20 cm. éstas a su vez están conectadas a bases móviles. El movimiento de los motores en ambos ejes es sincronizado a través de tres microcontroladores de la familia 16F876A. Los microcontroladores están conectados en una arquitectura maestro-esclavo. El PIC

maestro recibe las instrucciones desde el sistema de visión y las interpreta, a su vez envía instrucciones a los PIC'S esclavos posicionando los servomotores en el ángulo deseado.



**Fig. 2.** Estructura mecánica del prototipo.



**Fig. 3.** Ubicación de los motores para posicionamiento del laberinto.

### **3.2. Desarrollo del software**

El sistema de visión se implementó empleando el entorno de desarrollo de LabVIEW, debido a la facilidad con que permite utilizar diferentes dispositivos de adquisición de imágenes. Además permite desarrollar interfaces de usuario de forma intuitiva y los algoritmos de visión y reconocimiento de patrones pueden ser implementados con mayor facilidad que con otro tipo de programas.

La implementación del sistema inicia con la calibración y alineamiento de la cámara con respecto del plano, para evitar procesos de calibración demasiado pesados para el sistema, este procedimiento se logró identificando cuatro puntos en el laberinto y alienado la cámara de manera manual con cuatro regiones de interés que corresponden a las esquinas del laberinto, posteriormente se realiza la adquisición de una imagen en tres dimensiones. A continuación se identifica el objeto de interés (canica) y se separa del fondo de la imagen, para finalmente comparar sus características con las de un patrón previamente definido. Finalmente, el algoritmo permite obtener como resultado la posición de dicho objeto en el plano de la imagen.

Para ubicar el objeto de interés, se generó un patrón esférico con la imagen de la canica que se desea encontrar y se indicó al programa que la discriminación se hiciera por forma.

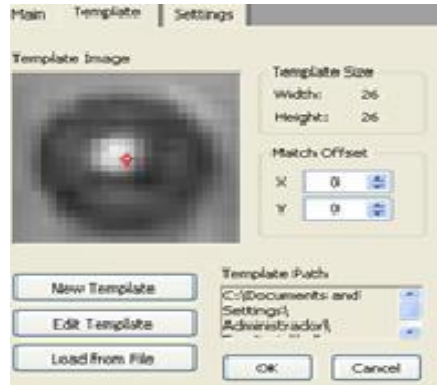


Fig. 4. Imagen del patrón.

Se realiza un filtro a la imagen adquirida para convertirla en escala de grises y a continuación se determinó una Región de Interés (ROI) que consiste en todas las áreas que no pertenecen al patrón y que corresponde con el área del laberinto.

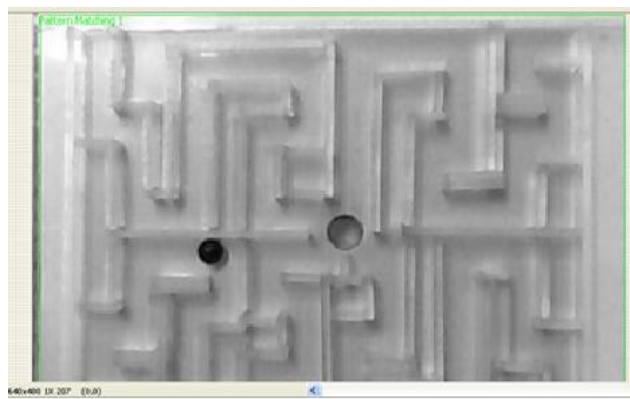
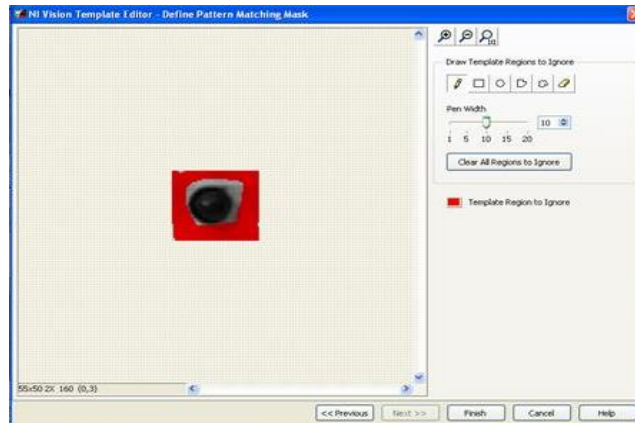


Fig. 5. Región de interés (ROI).

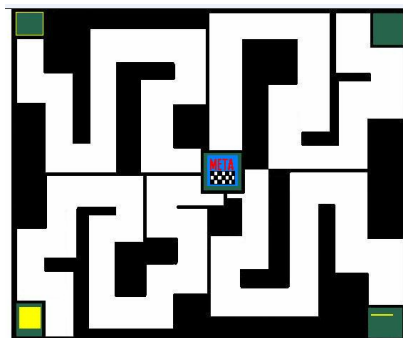
Una vez definido el patrón a seguir y eliminando las áreas que se requieren discriminar del mismo (Fig.6), el sistema de visión devuelve un clúster de información con los datos de origen en  $(x, y)$  que es la primera posición donde el sistema detecta la canica, y referencia  $(x, y)$  que es la posición actual del objeto, una vez conocida la posición de referencia y actual de la canica, se realiza un mapeo del laberinto obtenido con el sistema de visión en una imagen compuesta en formato JPG, y se dibuja sobre esta imagen una bola que corresponde a la posición de la canica real para verificar que las lecturas del sistema de visión corresponden a las medidas en el mundo real, actualizando ésta cada milisegundo con las nuevas posiciones de la canica.



**Fig. 6.** Eliminación de las áreas no deseadas del patrón.

Una vez que se mapea la imagen y se obtiene la posición del objeto en el plano, se analizan las regiones cercanas con respecto a las esquinas del laberinto y se define un movimiento hacia adelante, atrás, izquierda o derecha según sea el caso para hacer que la canica se vaya aproximando a la solución del centro. Esta información se envía a los motores por medio de una cadena de mando en formato RS232, en la estructura mecánica, ya que corresponde con un movimiento de la flecha para posicionar el laberinto en  $x$  e  $y$ .

En la implementación del prototipo, se crea un sistema de control de lazo cerrado, que considera las regiones donde se encuentra el objeto, para realizar un movimiento, de forma tal que, independientemente de donde se coloque la canica el programa será capaz de dar solución a cualquier caso presentado.



**Fig. 7.** Imagen del Laberinto en LabVIEW con la solución al centro.

### 3.3. Algoritmo de decisiones

Para que el sistema de visión pueda realizar un movimiento mecánico del laberinto se requiere de un sistema de toma de decisiones que sea capaz de reaccionar con un movimiento de la canica y trasladarla a un movimiento mecánico.

Las posiciones de la canica entran en un árbol de toma de decisiones, para poder determinar el movimiento de la bola por cada muestra proporcionada por la cámara, se realiza una máquina de estados para retroalimentar al sistema con las coordenadas de la bola y éstas son tomadas como parámetros para el árbol de toma de decisiones.

Una vez definido el movimiento que se tiene que realizar con el robot, este mediante un protocolo RS232, envía una cadena serial que corresponde al movimiento necesario para trasladar la canica al siguiente nodo de evaluación. La correspondencia en pixeles de la posición de la canica es mapeada hacia el dibujo del laberinto (Fig. 5) y es superpuesta sobre la imagen para corroborar la correspondencia de la posición física con lo reconocido por el sistema de visión.

La figura 8 muestra la máquina de estados que muestra el funcionamiento del sistema de seguimiento.

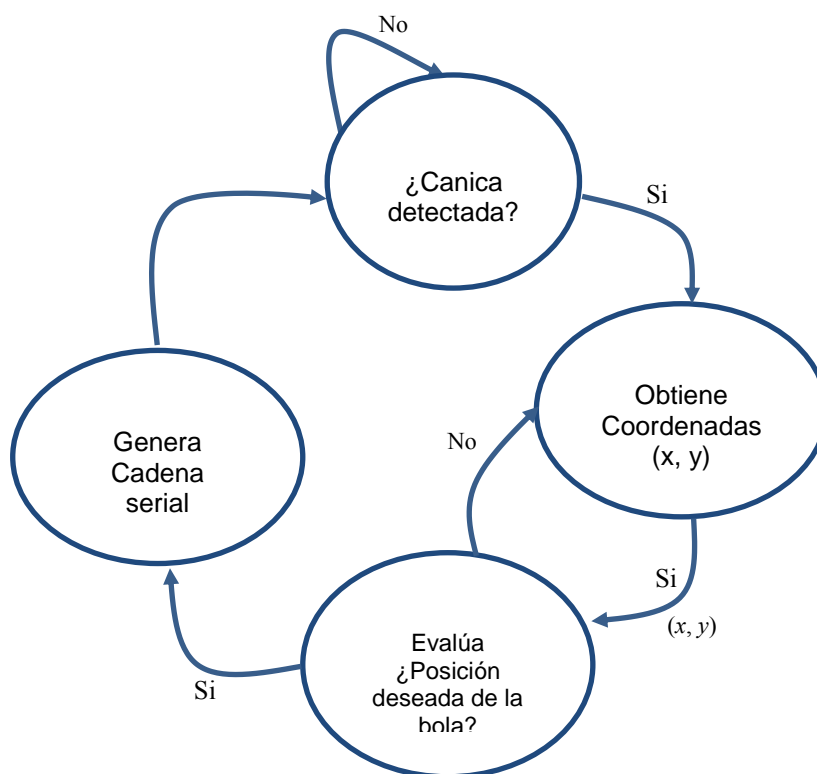


Fig. 8 Máquina de estados para el sistema de seguimiento.

Una vez definida la cadena serial que corresponda al caso para región de interés y recibida por un PIC maestro, este interpreta cada comando proveniente del sistema de visión para controlar dos microcontroladores esclavos que son los que físicamente modulan el PWM correspondiente para cada motor. Dando como resultado una cadena no determinística de movimientos en el plano físico X, Y y llevando a la canica a solucionar en el centro.

El diseño de esta máquina de estados permite resolver casos especiales, por ejemplo si la canica por alguna situación regresa o rebota en alguna pared, el diseño dinámico de la máquina de estados permite reorientar la canica a la siguiente posición.

#### **4. Pruebas y resultados**

Para comprobar la funcionalidad del Sistema, se realizaron múltiples pruebas colocando la canica dentro del laberinto desde diferentes puntos de inicio. Se observó que el movimiento de los motores fuera el adecuado para dar solución al laberinto. Desde la interface de usuario se observa que en todos los casos se podía identificar la canica y que el sistema de visión ejecuta de manera satisfactoria el algoritmo de seguimiento. El sistema es capaz de identificar y corregir de forma adecuada varios casos incluyendo cuando la canica por acciones mecánicas regresa a un punto anterior, el sistema de visión detecta y realiza el movimiento para reenviar la canica al punto siguiente.

#### **5. Conclusiones**

En este trabajo se presentó un prototipo funcional de un sistema de visión para seguimiento de objetos aplicado a la solución de un laberinto. El sistema se desarrolló exitosamente utilizando el toolbox IMAQ de LabVIEW, que permitió un desarrollo rápido y más simple en comparación con una programación que no es de tipo gráfica. Esta aplicación permitió desarrollar un Sistema de visión bastante robusto para el seguimiento de objetos, definición de trayectorias, y corrección de casos en tiempo real.

#### **Referencias**

1. Kwon, K S., Ready, S.: Practical Guide to Machine Vision Software: An Introduction with LabVIEW. N.J., USA, Wiley, pp. 279 (2014)
2. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2nd ed, USA, Academic Press, pp. 615, (1990)
3. Saldaña, G., Estévez, J., Gracios, C. A.: Sistema Automático de Clasificación Aplicado a la Industria Automotriz. Revista de Ciencia e Ingeniería del Instituto Tecnológico Superior de Coatzacoalcos, Vol.1, No.1, pp. 266–270 (2014)
4. Malamas, E. N.: A survey on industrial vision systems, applications and tools; Image and Vision Computing. Vol.21, No.2, pp. 171–188 (2003)
5. Travis, J., Kring, J.: LabVIEW for Everyone: Graphical Programming Made Easy and Fun. 3er Edition, New Jersey, Prentice Hall, pp. 960 (2006)
6. Lajara, J. R., Pelegrí, J. P.: LabVIEW: Entorno gráfico de programación. 2nd Edition, México, Alfaomega, pp. 477 (2013)