



Clasificador Inteligente de Objetos con Visión Artificial utilizando un Brazo Articulado

Robótica

Fernández Diego (dieg.fernandez@yahoo.com.ar), Ibarra Alexis Gabriel (axs.ibarra@gmail.com)
Facultad de Ciencias Exactas, Naturales y Agrimensura
Universidad Nacional del Nordeste
Corrientes, Argentina

RESUMEN

En este proyecto final se implementó un sistema clasificador de objetos, en donde se utilizó la visión artificial para la detección de las posiciones y las características de los objetos (color y tamaño), para que luego puedan ser manipulados por un brazo robot. Adicionalmente se desarrolló una aplicación que permite, utilizando el mismo hardware desarrollado y a través de la captura de los movimientos de la mano del usuario, el control del brazo robot. Para el logro del proyecto se desarrollaron programas y algoritmos utilizando el lenguaje de programación Python, las librerías OpenCV y Qt, y las plataformas de hardware Arduino y Pinguino (herramientas de código abierto). Los mismos son capaces de adquirir imágenes, procesarlas y controlar el movimiento del brazo robótico.

Vision Computarizada

La visión artificial, también conocida como visión por computador, es un subcampo de la inteligencia artificial. Los objetivos típicos de la visión artificial incluyen:

- La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes.
- Registro de diferentes imágenes de una misma escena u objeto.
- Seguimiento de un objeto en una secuencia de imágenes.
- Generación de modelo tridimensional de una escena.
- Estimación de las posturas tridimensionales de humanos.
- Búsqueda de imágenes digitales por su contenido.

Estos objetivos se consiguen por medio de reconocimiento de patrones, aprendizaje estadístico, geometría de proyección, procesamiento de imágenes, teoría de grafos y otros campos.

Hardware

Una plataforma de visión artificial consiste en la implementación de un sistema de captura y procesamiento de imagen a un sistema dinámico. La plataforma desarrollada consiste en una cámara webcam como sistema de captura, una PC para el procesamiento de imágenes, y un brazo robot de 5 DOF (grados de libertad) como sistema dinámico. Es controlado por la plataforma de hardware Arduino UNO.



Figura 1: Arquitectura del proyecto.

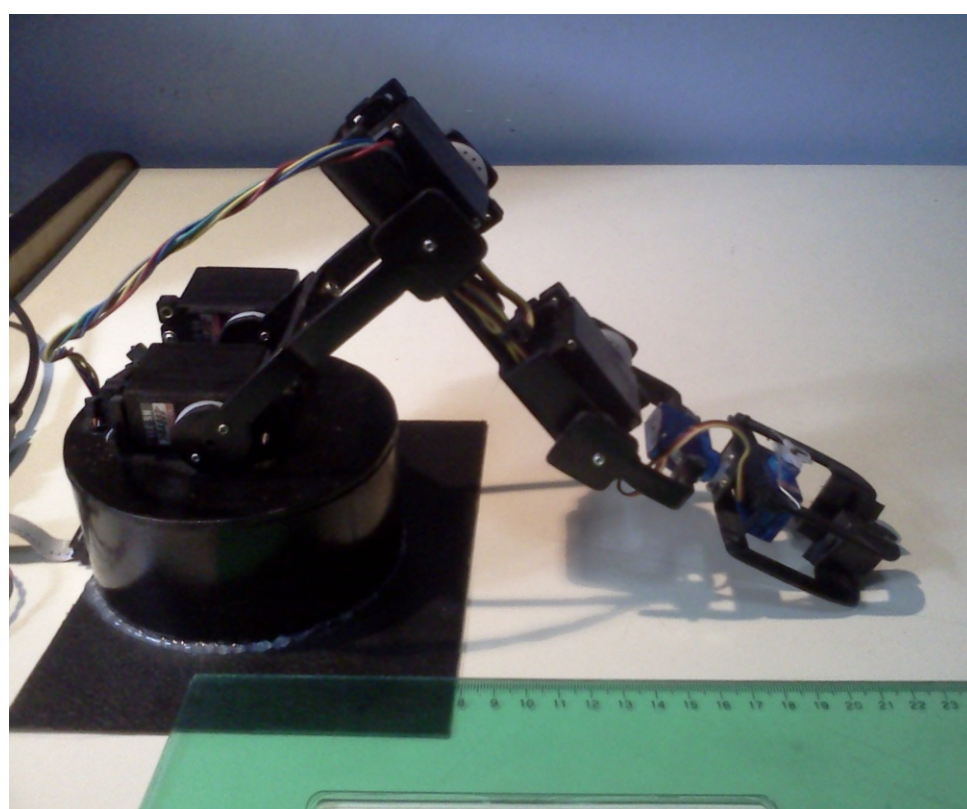


Figura 2: Brazo robot desarrollado.

Para la estructura del brazo robot se reciclaron tubos de PVC para la conducción de agua.

Se utilizaron los servomotores comerciales Hitec HS-311 (base, hombro, codo y muñeca) y Hitec HS-55 (muñeca rotacional y pinzas).

Se realizó un estudio de cinemática inversa del brazo robot, para contar con una expresión matemática que permita obtener las posiciones que deben adoptar cada servo para que el extremo del brazo alcance una coordenada deseada. Se optó por utilizar el método geométrico. El método requirió primero la obtención de las ecuaciones de cinemática directa, para luego, por despeje, obtener las ecuaciones de cinemática inversa.

Software

Se utilizó la librería de visión artificial OpenCV, especialmente sus funciones Histograms, Contours, Moments, ConvexHull, ConvexDefects, Meanshift y Camshift, con los cuales se implementaron los algoritmos de reconocimiento de las características de los objetos en imágenes (color y tamaño) y seguimiento de objetos en una secuencia de imágenes.

Para lograr flexibilidad, portabilidad y la posibilidad de realizar futuras mejoras al software se desarrolló íntegramente con herramientas libres y de código abierto (lenguaje de programación Python, PyQt y PyQt4 Designer para la interfaz gráfica).

La aplicación desarrollada consiste en la utilización de los algoritmos de procesamiento de imagen para el reconocimiento del color, tamaño y las posiciones de los objetos dentro de un espacio de trabajo, para que luego el brazo pueda tomarlos y depositarlos en otro lugar.

Se implementó el siguiente algoritmo (se aclaran las funciones de OpenCV utilizadas):

- 1) Búsqueda de objetos (cv.CalcArrBackProject).
- 2) Selección de objeto (cv.FindContours y cv.Moments).
- 3) Obtención de las posiciones (cv.CamShift).

- 4) Cálculo de posición futura.
- 5) Cinemática inversa.
- 6) Comunicación con el hardware.

Se realizó, además, una aplicación 2 en la que el usuario puede controlar la posición del brazo robot mediante la captura de los movimientos de su mano. El objetivo principal del software, en este caso, es detectar la posición de la mano en las 3 dimensiones. Para la obtención de la coordenada Z, se realizó un ensayo para hallar la relación entre la distancia de la mano al lente y el área de la mano en pixeles cuadrados. Dicha relación resultó prácticamente lineal.

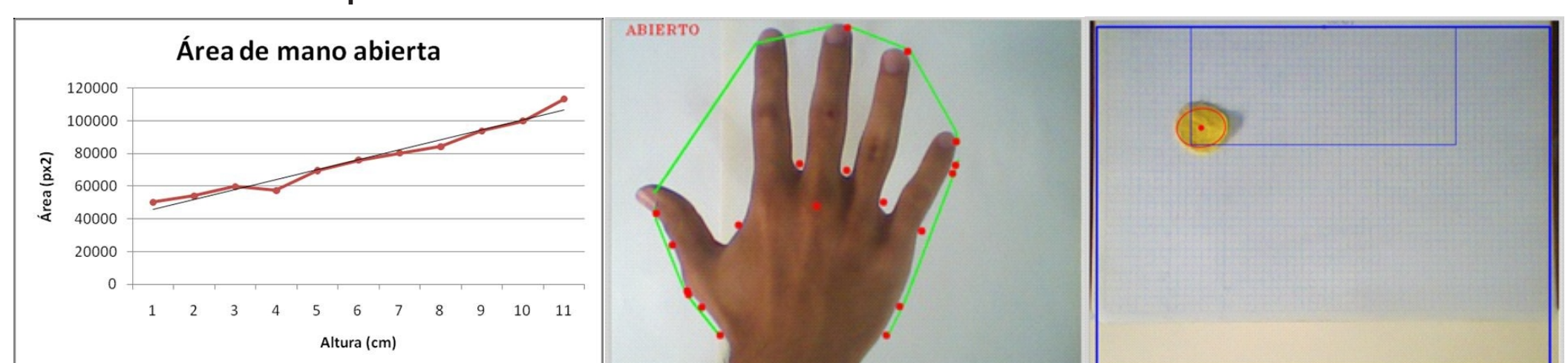


Figura 3: Relación área de la mano abierta a altura (izquierda), algoritmo de detección de la mano (centro) y ensayo del algoritmo de detección de objetos (derecha).

Ensayos y Resultados

Se realizaron pruebas de posicionamiento del brazo logrando un error máximo relativo de 0,087 (8,7 %). A su vez se probó el algoritmo de detección de objetos (fig. 3 derecha) en donde las posiciones detectadas contuvieron menos del 8% de error, por lo que el funcionamiento del algoritmo resultó muy aceptable.

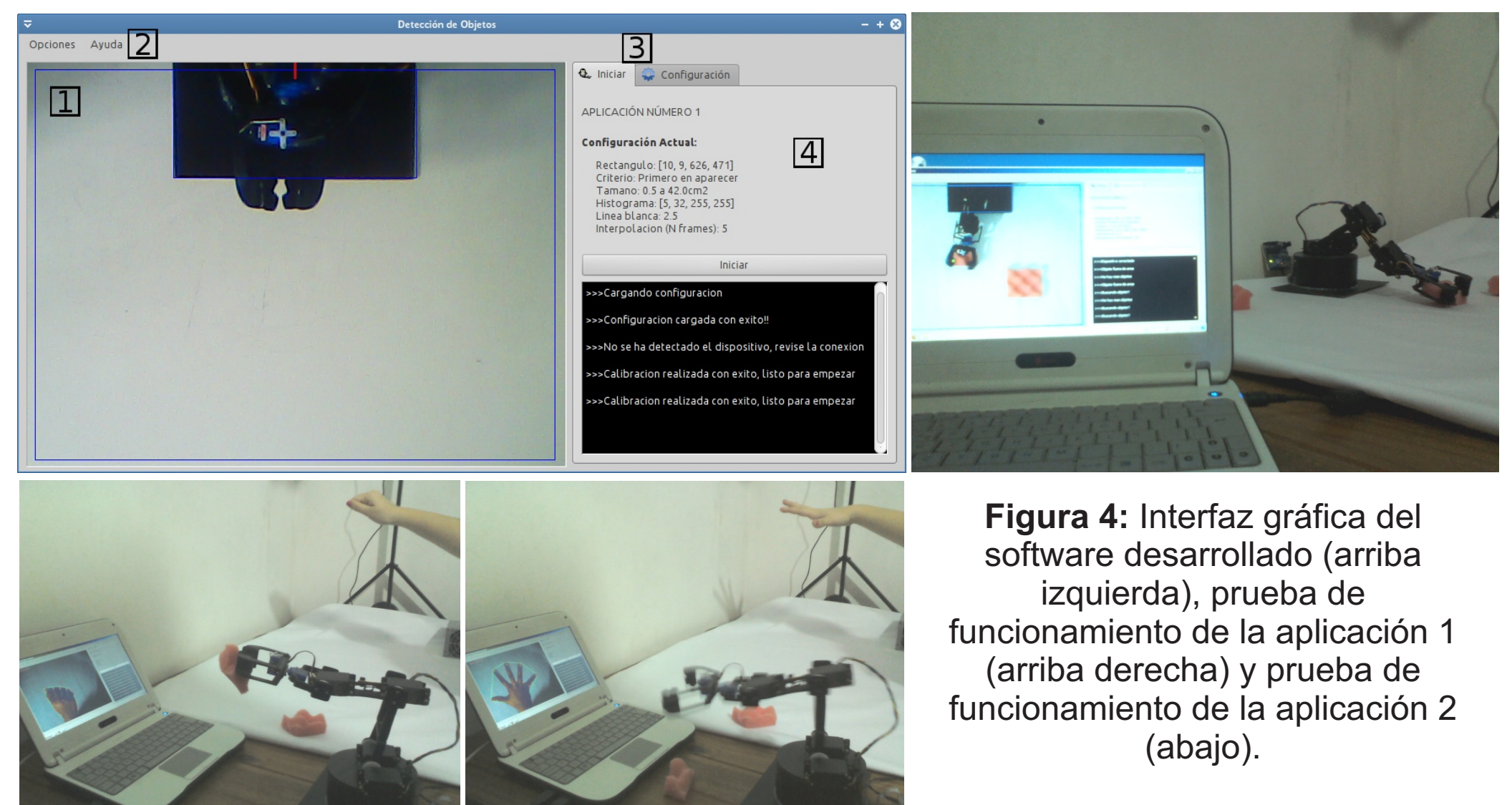


Figura 4: Interfaz gráfica del software desarrollado (arriba izquierda), prueba de funcionamiento de la aplicación 1 (arriba derecha) y prueba de funcionamiento de la aplicación 2 (abajo).

Conclusiones

Se logró desarrollar un sistema clasificador de objetos, en donde se utiliza la visión artificial para la detección de las posiciones y características de los objetos (color y tamaño), cumpliendo con el objetivo principal del proyecto. Se diseñó y construyó un brazo robot de 5 DOF con materiales de bajo costo, realizando también el estudio de su cinemática inversa.

Se ha logrado el diseño e implementación de una plataforma de visión artificial. Su buen funcionamiento depende de que la iluminación del ambiente en donde se encuentre sea uniforme generando el menor contenido de ruido en la imagen capturada.

Como trabajos futuros se proponen, entre otros:

- Determinación de posiciones de objetos en movimiento.
- Utilizar cámaras digitales de mayor calidad.
- Incorporar a la plataforma un autómata de mayor tamaño y capacidad.
- Independizarse del uso de una PC desarrollando un sistema embebido.
- Agregar sensores al efector final del brazo robot.

Referencias

- Gary Bradski, Adrian Kaehler, Learning OpenCV, O'Reilly Media, 2008
- José Jaime Esqueda Elizondo, Luis Enrique Palafox Maestre, Fundamentos de procesamiento de imágenes, Universidad Autónoma de Baja California, 2005.
- OpenCV v2.4.0 Documentation, <http://docs.opencv.org/>, 2012.
- Python v2.7.3 documentation, <http://docs.python.org/2/>, 2010.
- Aníbal Ollero Baturone, Robótica: manipuladores y robots móviles, Marcombo, 2001.
- Jorge Rodríguez Araújo, Introducción a los sistemas de visión artificial, 2009.