

Control de un Mini Robot de 5 grados de Libertad usando Visión Artificial.

Control of a 5 Degree of Freedom Robot using Computer Vision.

José Manuel Torres Zavala, jm.torreszavala@ugto.mx

Departamento de Estudios Multidisciplinarios, División de ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato,

José Emiliano Reyes Lemus, je.reyeslemus@ugto.mx

Departamento de Estudios Multidisciplinarios, División de ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato,

Jesús Rafael Tapia Camargo, jr.tapiacamargo@ugto.mx

Departamento de Estudios Multidisciplinarios, División de ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato,

Juan Manuel Jiménez Rosiles, jm.jimenezrosiles@ugto.mx

Departamento de Estudios Multidisciplinarios, División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato,

Fabian García Chávez, f.garciachavez@ugto.mx

Departamento de Estudios Multidisciplinarios, División de ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato,

Emiliano Ortega García, e.ortegagarcia@ugto.mx

Departamento de Estudios Multidisciplinarios, División de ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato,

Juan Manuel López Hernández, jmlopez@ugto.mx

Departamento de Estudios Multidisciplinarios, División de ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato,

Resumen

El uso de Robots en la industria es cada vez más frecuente sobre todo en procesos repetitivos como la toma de objetos para llevarlos de un proceso a otro en la línea de producción o tomar un objeto que viene en la banda transportadora y posicionarlo en un lugar específico como puede ser una caja para su empaquetado. Estos procesos repetitivos los puede realizar un robot con ayuda de actuadores que se activan para moverse de una forma específica. En la actualidad estos robots son programados con líneas de código que activan actuadores para que el robot realice las diferentes secuencias de movimiento. En este proyecto, se propone la utilización de algoritmos de visión artificial para proporcionar a un mini robot de 5 grados de libertad, la capacidad de visión y por medio de algoritmos de adquisición y procesamiento de imágenes, la identificación y localización de objetos específicos con ayuda de una webcam USB como sistema de visión. Una vez realizada la detección del objeto, el MATLAB enviara los códigos de control necesarios a un sistema con microcontrolador ATMEGA328P para que los actuadores se activen de tal forma que el robot alcance el objeto o producto a manipular y posicionarlo en una pequeña bandeja de plástico.

Palabras clave: Visión por computadora, Segmentación de imágenes, contornos activos, control por visión.

Introducción

Este proyecto está posicionado dentro del área de robótica y control de procesos industriales, donde la construcción de máquinas que sean capaces de simular actividades propias de una persona, como visualizar objetos es de suma importancia. La visión artificial como la visión de un robot, entra en un proceso que se llama en los humanos "percepción visual" que es la habilidad para procesar, interpretar y analizar información recibida mediante nuestros ojos. Una de las tendencias actuales desde la aparición de las computadoras y más recientemente la aparición de la Inteligencia Artificial (IA), es el de poder desarrollar sistemas autónomos que no solo operen en ambientes restringidos y conocidos, sino que sean capaces de operar en ambientes abiertos y desconocidos como una persona lo puede hacer. El área que involucra la percepción visual en los robots es la de Procesamiento de imágenes digitales en la cual podemos desarrollar ciertos algoritmos de procesamiento en un ambiente de programación para identificar ciertas características de una escena o imagen que este visualizando el sistema robótico. La identificación o separación de estas características a partir de una imagen, se le conoce como segmentación de imágenes. Existen diferentes tipos de algoritmos de procesamiento de imágenes (Gonzalez, Woods, 2008) (Pratt, 2007) donde cada uno se utiliza dependiendo de lo que se desea obtener, por ejemplo, identificar rostros, colores y formas dentro de una imagen adquirida por medio de un sistema de visión.

En la figura 1 se muestra un diagrama a bloques de las diferentes etapas del proyecto para llevar a buen fin el trabajo de investigación, en estas etapas se realiza uso de Software como MATLAB™ y Arduino IDE así como Hardware como una WEBCAM y una tarjeta con microcontrolador ATMEGA328p (Mosquera, 2019).

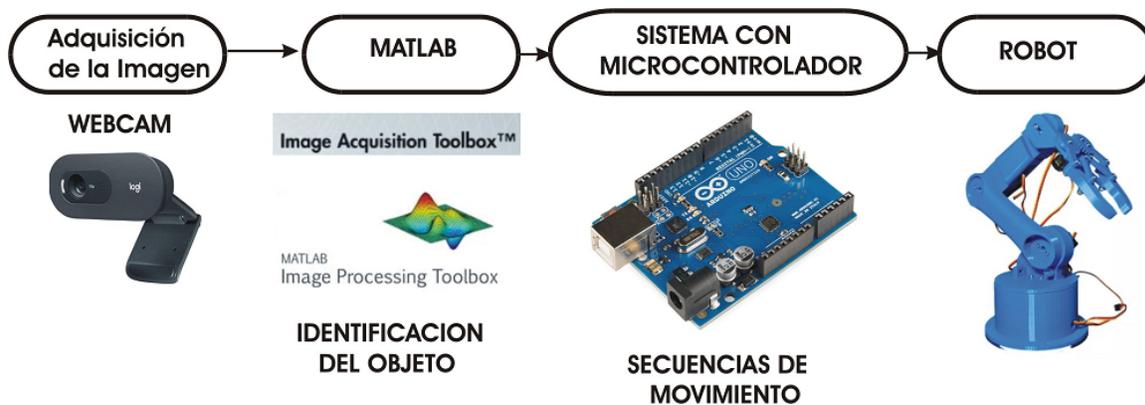


Figura 1. Diagrama a bloques del proyecto.

Bloque 1: Uso de una webcam USB2.0 como sistema de visión configurada a una resolución de 640x480 color.

Bloque 2: Uso de las cajas de Herramientas (ToolBox) de adquisición de imágenes y de procesamiento de imágenes para adquirir y procesar la imagen adquirida para la detección de objetos en la escena o imagen.

Bloque 3: Dentro del sistema de MATLAB se establece una comunicación serial USB entre el MATLAB y el sistema con microcontrolador para el envío de señales de control para el robot.

Bloque 4: Estructura del Robot de 5 grados de libertad con actuadores que en nuestro caso son servomotores que proporcionaran los movimientos y la fuerza necesarias para la realización de secuencias de movimiento.

El robot fue adquirido por el laboratorio de procesamiento de señales del DEM perteneciente al Campus Irapuato-Salamanca para su uso en prácticas de mini robótica y se cuenta con 4 unidades, cada unidad consta solo de la estructura impresa en 3D y de los actuadores que en nuestro caso son servomotores que pueden ser controlados por cualquier sistema con microcontrolador. En este proyecto usamos el sistema con microcontrolador ATmega328p (Mosquera, 2019) por su uso amigable con el programador y por su bajo costo.

En la figura 2, se muestra el robot con las dimensiones físicas, así como la estructura que se construyó para diversas aplicaciones de mini robótica.

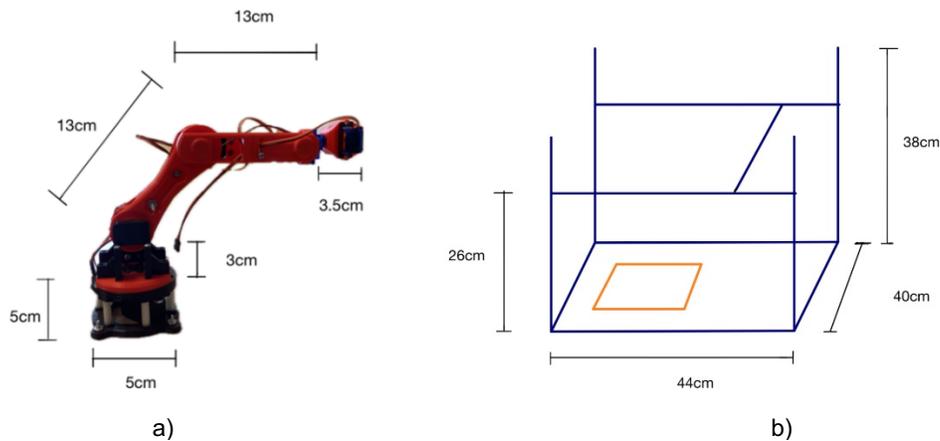


Figura 2. (a) Dimensiones del robot a utilizar, (b) estructura realizada en vigueta de aluminio donde se alojará el robot, el espacio en color naranja corresponde al espacio donde se encuentra el robot.

Métodos Comunes de Segmentación

El término de segmentación significa partición de una imagen en píxeles que contengan las mismas características como color, textura, etc., dentro de los métodos básicos de segmentación podemos mencionar (Gonzalez, Woods, 2008) (pratt, 2007):

- 1.- Segmentación por Umbral (Thresholding) en la cual divide la imagen en regiones basadas en valores de intensidad de los píxeles.
- 2.- Segmentación Basada en Regiones en la cual agrupa píxeles en regiones basadas en criterios de homogeneidad.
- 3.- Segmentación Basada en Bordes en la cual se aplican detectores para mostrar los bordes de los objetos que se encuentran en la escena como el algoritmo por contornos activos[] utilizado en este proyecto.
- 4.- Segmentación por Agrupamiento (Clustering) dentro de este método, se aplican algoritmos de agrupamiento como K-means para agrupar píxeles en diferentes regiones

Segmentación de objetos por contornos activos.

Los contornos activos (*Kass, Witkin, Terzopoulos, 1987*) es una técnica de segmentación que se basa en el movimiento de una curva cerrada compuesta de un conjunto de puntos unidos entre si como si fueran una curva elástica la cual definiremos como $c(q)$ definida en el espacio R^2 . A este conjunto de puntos individuales con coordenadas (x,y) se les conoce en la literatura como Snaxels (como analogía podemos mencionar que los píxeles son componentes individuales de una imagen). Esta curva se moverá tendiendo a cerrarse en un punto si no existiera objeto a localizar y si el componente individual o punto de la curva encuentra en su camino a un pixel con una cierta característica como color, etc., el punto se detendrá (se pone ese snaxel con energía cero) atrayendo a los otros puntos de la curva (como si tuvieran un resorte entre cada uno de los puntos de la curva) y de esta forma localizando en la imagen el contorno del objeto que se encuentra en la imagen del sistema de visión. Esta evolución o movimiento de la curva hasta llegar al contorno del objeto de interés, se debe a un proceso por iteraciones que son definidas por el programador.

El algoritmo de segmentación por contorno activo se basa en la ecuación 1 y que es la Energía total de la curva (Esnake) como la suma de la energía interna y la energía externa.

$$E_{snake} = \int_0^1 [E_{int}(c(q)) + E_{ext}(c(q))] dq \quad (1)$$

Se le llama energía, pero en realidad es un valor numérico que se hace cero cuando el Snaxel llega al contorno del objeto.

La energía interna E_{int} es un parámetro intrínseco de la curva y a grandes rasgos es modelada como una lamina (segunda derivada de la curva) que se encuentra entre cada 3 puntos de la curva. Estos modelos proporcionan a la curva un comportamiento como una liga que atrae a los otros puntos que todavía no llegan a la energía cero (que todavía no llegan al contorno del objeto). Cuando existe un objeto dentro de la curva, el comportamiento de liga provoca que los puntos o snaxels tiendan a moverse y cerrarse en el centroide de la curva o centro de masa de los puntos.

La energía externa E_{ext} es definida por las características externas de la curva como el gradiente de la imagen $\nabla I(c(q))$ el cual corresponde a la localización del contorno del objeto. Este gradiente significa la detección en un cambio de intensidad entre pixeles adjuntos que en imágenes se le conoce como el contorno de un objeto. Entonces la ecuación de la energía total se transforma en la segunda derivada de la curva en conjunto con el gradiente de la imagen que son los bordes del objeto.

$$E(c) = \beta \int_0^1 |c''(q)|^2 dq + \lambda \int_0^1 |\nabla I(c(q))| dq \quad (2)$$

Los parámetros beta y lambda son las contribuciones de cada termino y son seleccionados por el usuario programador (ya que depende del comportamiento de la curva, estos valores están entre 0 y 1) dependiendo de los resultados obtenidos.

En la figura 3 se muestra el funcionamiento del contorno activo para la detección del contorno del objeto. La curva inicial (a) son los snaxels que rodean al objeto a detectar su contorno y por consiguiente al obtener el centroide de los puntos de la curva final (b) obtendremos la localización del objeto.

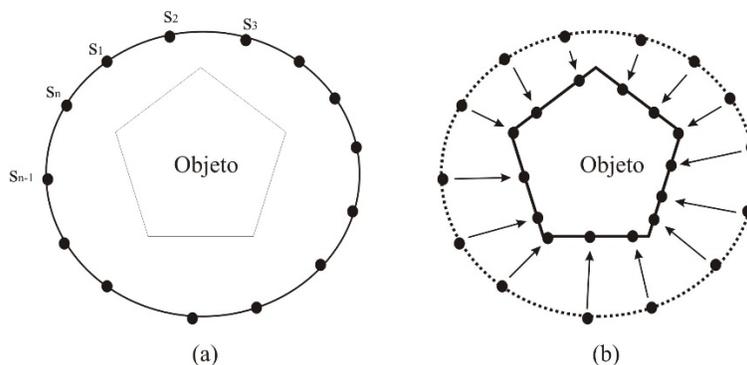


Figura 3. (a) Contorno activo formado por puntos individuales llamados snaxels, (b) evolución de la Curva hasta localizar los bordes del objeto.

El sistema Arduino UNO está basado en el microcontrolador ATMEGA328p de 8 bits con arquitectura AVR RISC de alto rendimiento con una memoria flash de 32kb con capacidades de lectura y escritura, 23 líneas de entrada salida, 32 registros de trabajo de propósito general. Su utilización se ha extendido en los últimos años por su bajo costo y software libre. Para este proyecto, este sistema con microprocesador es el adecuado para controlar los servomotores además nos permite realizar la comunicación con el MATLAB para el envío de comandos de control.



Figura 4. a) Sistema con microcontrolador ATmega328p y b) designación de terminales.

Usamos este sistema para controlar de forma digital los actuadores o servomotores que moverán los eslabones del robot. El sistema Arduino posee las librerías necesarias para controlar los actuadores por medio de señales PWM que controlan el ángulo de giro del servomotor. Además posee los circuitos y librerías necesarias para establecer la comunicación serial con la PC.

Entorno MATLAB.

El MATLAB es un ambiente de programación desarrollado en la década de los 80's, en un principio en un ambiente MSDOS por comandos, para después en los 90's con la llegada de WINDOWS en un ambiente de visualización grafica a color. En su desarrollo se han ido desarrollando librerías que contienen un conjunto muy amplio de rutinas para el procesamiento y análisis de datos. Estas librerías también llamadas TOOLBOX son un conjunto de funciones desarrolladas para ciertos fines como el de procesamiento de imágenes (Image Processing Toolbox) y el de Image Acquisition que son los que utilizamos para desarrollar este proyecto. El Toolbox de Procesamiento de imágenes, posee las funciones necesarias para realizar el procesamiento de la imagen adquirida por la WEBCAM en la cual podemos configurar el tamaño a 640x480 y encontrar el contorno del objeto visualizado en la imagen. El Toolbox de image acquisition nos proporciona las funciones necesarias para interactuar con la WebCam y adquirir una imagen o un conjunto de imágenes. En la figura 5 podemos ver la WebCam utilizada en el proyecto.



Figura 5. WebCam utilizada para la adquisición de imágenes.

Comunicación serial entre el MATLAB y el microcontrolador ATMEGA328p.

El MATLAB tiene la capacidad para establecer una comunicación serial USB2.0 con cualquier sistema de adquisición o de control, específicamente el ATMEga328p posee la comunicación USART (Universal Synchronous Serial Receiver and Transmitter) para transferencia y recepción de datos seriales. Como lo indica la siguiente figura, además podemos establecer una WEBCAM como nuestro sistema de visión para la adquisición de las imágenes a procesar para obtener la identificación y localización física del objeto a manipular.

Control de servomotores.

Los servomotores (Mosquera, 2019) son dispositivos electromecánicos utilizados para controlar el movimiento con precisión en una variedad de aplicaciones, desde robótica y automatización industrial hasta modelos de radiocontrol y sistemas de dirección en vehículos. Estos motores son conocidos por su capacidad para mantener una posición específica con precisión como el ángulo de rotación del eje que puede variar de 0 a 180. Aquí hay algunos aspectos clave sobre los servomotores:

Control de posición: Los servomotores son capaces de mantener una posición específica o moverse a una posición deseada con precisión. Esto se logra mediante el envío de señales de control precisas, como pulsos PWM (modulación por ancho de pulso) o señales de posición, que indican al motor la posición exacta que debe alcanzar.

Retroalimentación: La mayoría de los servomotores están equipados con algún tipo de retroalimentación, como un potenciómetro o un codificador. Esta retroalimentación permite al controlador del servomotor saber la posición actual del eje del motor, lo que permite un control preciso de la posición y la velocidad.

Torque: Los servomotores están diseñados para proporcionar un alto torque o fuerza de giro, lo que los hace adecuados para aplicaciones donde se requiere un control preciso del movimiento incluso bajo carga.

En la figura 6 se pueden observar los 2 tipos de servomotores que utiliza el robot, 3 de especificaciones de alto torque que son los que se sitúan en la parte baja del robot para poder soportar y mover el peso de las articulaciones superiores y 2 que son de bajo torque más pequeños. El hecho de que los servomotores sean más pequeños en las articulaciones superiores es para disminuir el peso de las articulaciones superiores como el que abre y cierra la pinza del robot para sujetar objetos.



Figura 6. a) Servomotor de alto torque 1.2Kg.cm, b) Servomotor de bajo torque de 800gr.cm.

Cada grado de libertad significa un servomotor que realiza un movimiento de giro entre 0 grados y 180 grados, los hemos designado como M1 que es el que realiza la rotación de todo el robot (se encuentra en la base del robot), M2 es el que realiza el movimiento de subir o bajar el primer eslabón (mueve toda la parte superior del robot). El servomotor M3 mueve el segundo eslabón, M4 sube o baja el mecanismo donde se encuentra la pinza para tomar objetos y por último M5 se encarga de abrir o cerrar la pinza para tomar o soltar los objetos. En la figura 7 se ilustra la designación de los actuadores del robot.

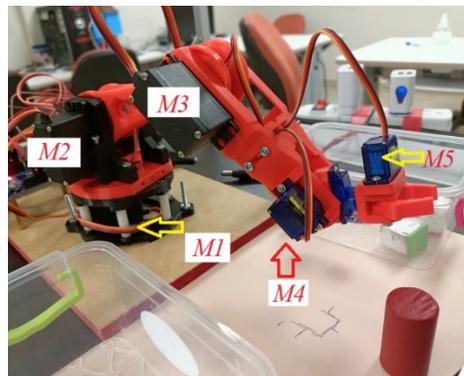


Figura 7. Designación de etiquetas de cada uno de los servomotores para una fácil identificación.

Robot de 5 grados de libertad

En la figura 8 se muestra el robot desde diferentes ángulos, la estructura del robot está hecha con impresora 3D fabricada por la empresa, los actuadores o servos para proporcionar al robot el movimiento para realizar las secuencias de desplazamiento, en color Negro son servos de 1.2 Kgf.cm y en azul tenemos los servos de 800 grf.cm.

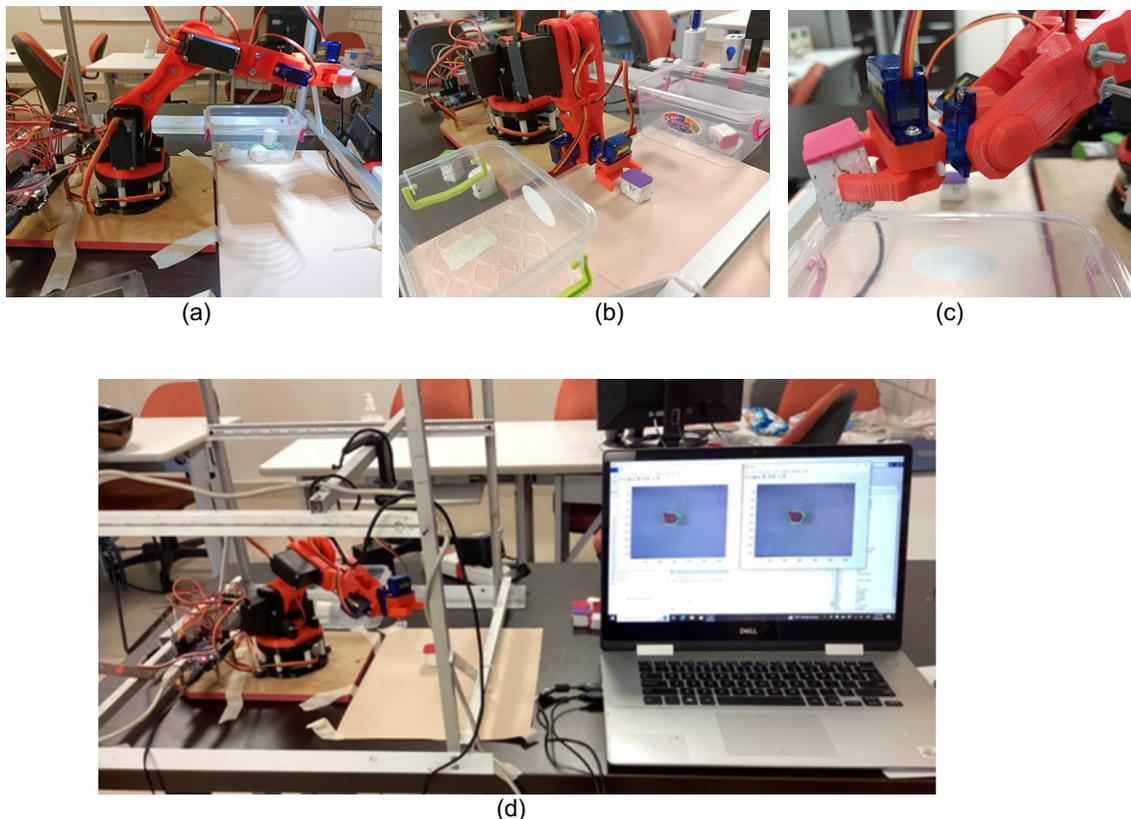


Figura 8. Diferentes vistas del robot realizando diferentes secuencias de movimiento, a) prueba inicial de funcionamiento, b) tomando un objeto para posicionarlo en la bandeja, c) antes de posicionar el objeto en la bandeja y d) vista panorámica del robot y la computadora donde se muestra en la pantalla la identificación del objeto.

En la figura 9 podemos ver el preview de la webcam donde se puede observar el objeto a manipular, estos objetos a manipular son cubos de unicel de aproximadamente 1cm por lado y con un cuadro de foami de

color rojo o morado para que el sistema de visión lo pueda identificar, podemos ver también el fondo de nuestro campo de visión que es una hoja de color café claro ya que entre el objeto a localizar y el fondo debe de haber diferencia remarcable para que el objeto pueda ser identificado.

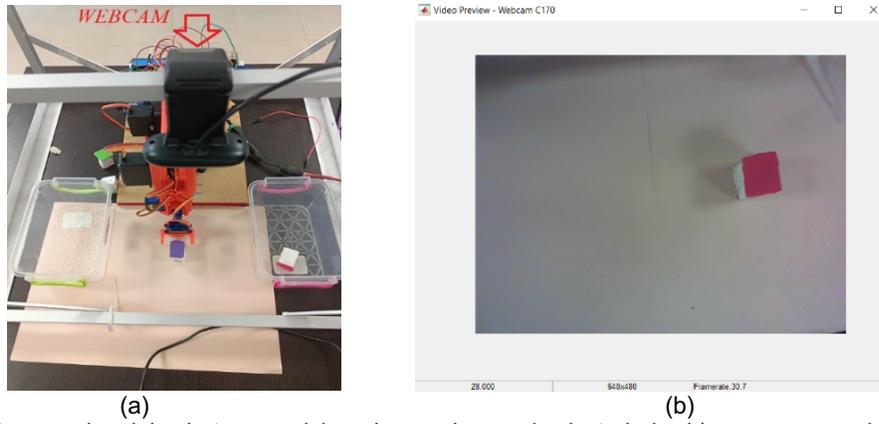


Figura 9. a) vista superior del robot ya posicionado en el espacio de trabajo, b) escena que visualiza la webcam en formato color 640x480.

Identificando el objeto a manipular

Se realiza el código necesario para que la webcam adquiera una imagen de 640x480 donde se observa el objeto a manipular, el MATLAB la recibe y la procesa obteniendo primero el contorno del objeto para posteriormente usar los contornos activos para encontrar las coordenadas del centroide de la curva final y así obtener la identificación espacial del objeto.

En la figura 10 se muestra la identificación por el algoritmo por contornos activos

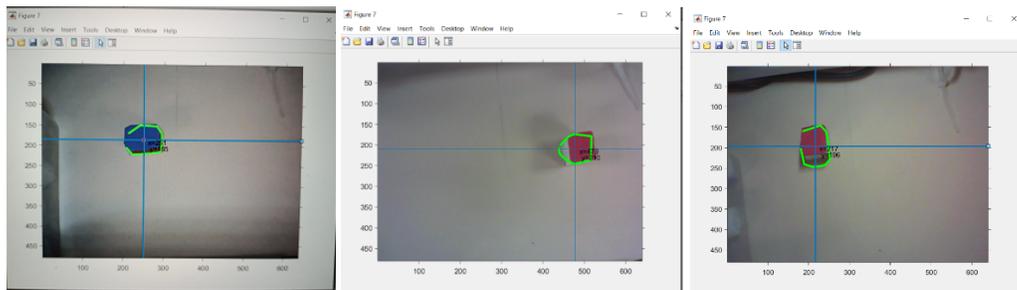


Figura 10. Objeto localizado en diferentes posiciones del campo de visión de la WebCam.

Se posiciono un contenedor de plástico para alojar los objetos manipulados, en la primera imagen el objeto es localizado en las coordenadas (251,185) mientras que en la segunda imagen el objeto se encuentra en las coordenadas (479,210) y en la figura 12 c) se muestra el objeto localizado en las coordenadas (217,196)

Conclusiones

La robótica industrial y el procesamiento de imágenes han transformado la manufactura y el análisis visual en diversas industrias. Los avances en robótica, como los brazos manipuladores tipo PUMA y los robots paralelos, han mejorado significativamente la precisión y la eficiencia en la producción. Por otro lado, técnicas avanzadas de procesamiento de imágenes como la segmentación y los contornos activos permiten una identificación y análisis más precisos de objetos y regiones en imágenes digitales. La integración de estas tecnologías continúa impulsando la innovación y la productividad en múltiples sectores. En este proyecto de investigación se usaron técnicas modernas de visión artificial como los contornos activos que como su nombre lo indica, a partir de una curva inicial de puntos posicionados por el usuario cercas de los límites de la imagen estos se mueven por llamadas energías internas que tienen que ver con los modelos intrínsecos de la curva que en nuestro caso es por modelo de lámina rígida lo que ocasiona que a cada iteración se vayan acercando los puntos convergiendo al centroide de la curva cerrada. Otro modelo que se puede utilizar es el de globo en el cual el contorno activo se expande en la misma proporción como si fuera un globo hasta encontrar los bordes del objeto y por otro lado tenemos las energías externas como son el contorno del objeto que atrae al contorno activo y de esta forma los puntos del contorno activo finales estarán sobre los bordes del objeto a localizar. Este tipo de segmentación por contornos es muy poderoso ya que podemos segmentar objetos de cierto color u objetos con cierta textura, para esto debemos de programarle al contorno activo los modelos necesarios para detectar lo deseado.

Referencias

- R. Gonzalez, R. E. Woods, (2008), *Digital Image Processing*, (3ª ed.), Prentice Hall.
- William K. Pratt, (2007), *Digital Image Processing*, (4ª ed.), Wiley interscience.
- M. Kass, A. Witkin, D. Terzopoulos, (1987), *Snakes, active contours models*, Int. J. on Computer Vision, 1(4):321-331.
- Ruben Beiroa Mosquera, (2019), *Aprender Arduino, electrónica y programación*, (1ª ed.), Edit. Alfaomega Marcombo.