

Visión para Robots. Proyecto Final:

Clasificador RRR de Formas (04/12/2017)

Miguel Ángel Guzmán A01327078, Juan Bosco Cervantes Quintana A01327406, Guillermo López Ruiz A01326965, Raúl del Saz González A01676735

Resumen— Para este proyecto se aplicarán los conceptos aprendidos durante el curso de Visión para Robots. Se tomará como punto de partida la conversión a escala de grises de imágenes digitales en tiempo real, técnicas de binarización y algoritmos de detección de formas implementados con las librerías ofrecidas por Labview Vision y myRIO.

Esto con el fin de desarrollar un robot clasificador de formas utilizando un brazo robótico de tres grados de libertad con uniones de revoluta con un electroimán en el efector final. Una cámara USB se conectará a myRIO donde se realizará el procesamiento digital en tiempo real y le informará al brazo mediante Arduino como debe colocar los servomotores para imantar la figura deseada para su posterior depósito.

Index Terms— Matlab, Arduino Board, LabVIEW, NI myRIO, Binarización

I. INTRODUCCIÓN

En el área de visión maquina, la detección de figuras o formas simples representa un problema muy común dentro del nivel intermedio. Dentro de este tipo de procesamientos los algoritmos de detección de bordes como lo puede ser el ya implementado algoritmo de Canny, son requerimientos previos imperativos para determinar los puntos dentro de la imagen que pertenecen a la frontera de la figura deseada. Este proyecto requiere de una binarización limpia de una imagen de espacio de trabajo proporcionada en tiempo real por una cámara web para realizar un posterior algoritmo de detección de formas basado en el entrenamiento con templates.

Manuscrito recibido el 04 de Diciembre de 2017. Esta práctica fue propuesta por Walther Carballo Hernández, Profesor de Visión para Robots en Tecnológico de Monterrey Campus Puebla.

Autor M.G, ITESM Campus Puebla, Ingeniería en Sistemas Digitales y Robótica, Puebla (e-mail: A01327078@itesm.mx).

Autor J.C, ITESM Campus Puebla, Ingeniería en Mecatrónica, Puebla (e-mail: A01327406@itesm.mx).

Autor G.L, ITESM Campus Puebla, Ingeniería en Tecnologías Electrónicas, Puebla (e-mail: A01326965@itesm.mx).

Autor R.S, ITESM Campus Puebla, Ingeniería en Tecnologías Electrónicas, Puebla

II. MARCO TEÓRICO

Para el desarrollo de esta práctica ciertos aspectos teóricos y consideraciones acerca del material y entorno de trabajo deben ser considerados:

- Binarización

Técnica que consiste en la realización de un barrido en la matriz de la imagen digital, por medio de bucles o recursividad, con el fin de que el proceso produzca la reducción de la escala de grises a dos únicos valores. Negro (= 0) y blanco (= 255), o lo que es lo mismo, un sistema binario de ausencia y presencia de color 0-1 [1].

- Matlab

MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux [2].

- Arduino Board

Placa de circuito que integra un microcontrolador Atmega, puertos analógicos y digitales, y un entorno de desarrollo (IDE) que se utiliza para programar, compilar y cargar código en la placa.

Arduino recibe información del dispositivo NI myRIO para ubicar al objeto dentro del campo de trabajo, de tal manera que el brazo logre posicionarse en el objeto para su posterior manipulación [3].



2.1 Arduino Board

- LabVIEW

LabVIEW es un entorno de desarrollo integrado diseñado específicamente para ingenieros y científicos que construyen sistemas de medición y control. Con un lenguaje de programación gráfica nativo, IP incorporada para análisis de datos y procesamiento de señales, y una arquitectura abierta que permite la integración de cualquier dispositivo de hardware y cualquier enfoque de software, LabVIEW es el software que necesita para construir la solución óptima que satisfaga sus necesidades requisitos y resolver los desafíos a la mano [4].



2.2 LabVIEW software logo

- NI myRIO

El myRIO - Dispositivo embebido para estudiantes presenta E / S en ambos lados del dispositivo en forma de conectores MXP y MSP. Incluye entradas analógicas, salidas analógicas, líneas de E / S digitales, LED, un botón pulsador, un acelerómetro incorporado, un FPGA Xilinx y un procesador ARM Cortex-A9 de doble núcleo. Algunos modelos también incluyen soporte WiFi. Puede programar el myRIO - Dispositivo embebido para estudiante con LabVIEW o C[4].



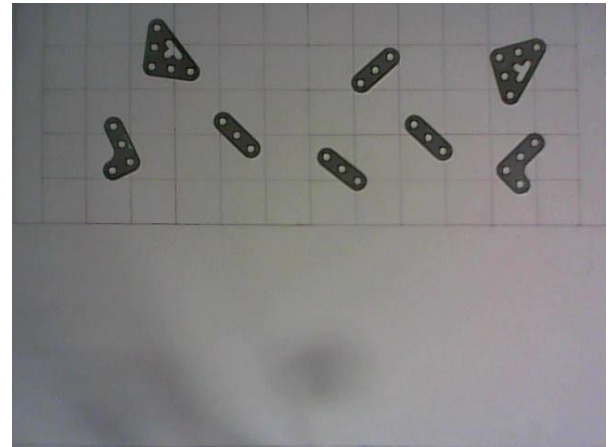
2.3 myRIO board

- LabVIEW Vision

Software de visión artificial que proporciona funciones de procesamiento de imágenes y visión artificial para Labview, el cual adquiere, guarda y presenta imágenes compatibles para diferentes hardware de visión[4].

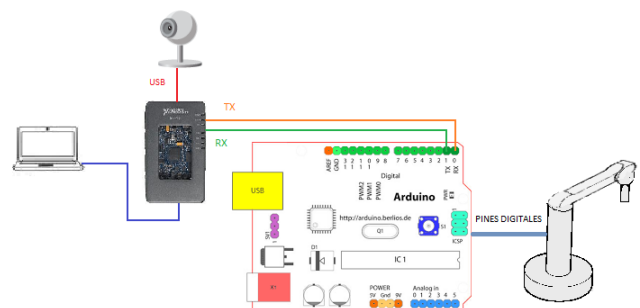
III. DESARROLLO

El sistema consta de una cámara web que está todo el tiempo tomando captura de un espacio de trabajo donde se colocan tres tipos de formas metálicas: triángulos, 'L's, y rectas.

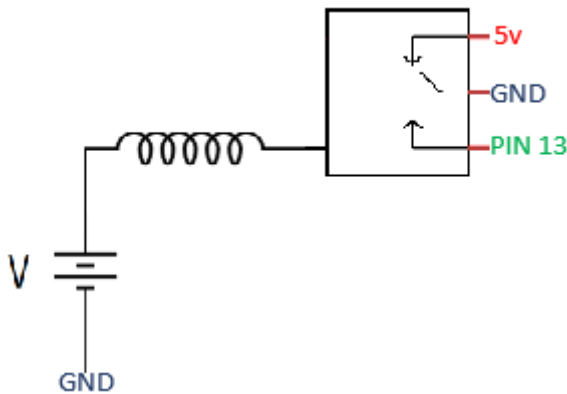


3.1 Imagen del espacio de trabajo vista por la cámara USB.

La cámara está conectada a su vez vía USB a una tarjeta myRIO que realiza el procesamiento requerido de la imagen para nuestra aplicación. La tarjeta myRIO se encuentra conectada a una laptop para mostrar la interfaz gráfica de usuario que muestra entre otras cosas la captura original de la cámara en tiempo real junto a su versión binarizada. Asimismo, la tarjeta myRIO se comunica por medio de UART con una placa Arduino que se conecta con el actuador del proyecto, un brazo robótico de tres grados de libertad de uniones tipo revoluta y con un electroimán como su efector final que permitirá agarrar los objetos metálicos del espacio de trabajo. Los servomotores se conectan a los pines digitales del Arduino así como un relevador que permitirá activar el electroimán.

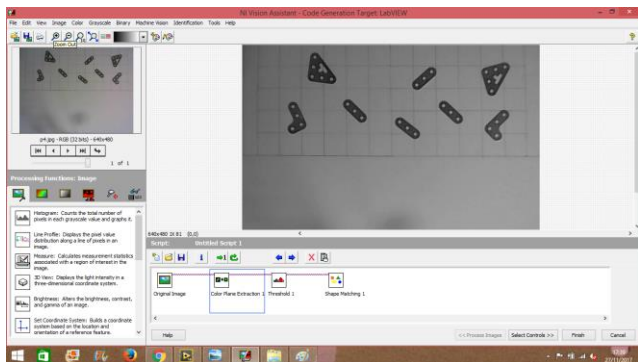


3.2 Diagrama de conexión.



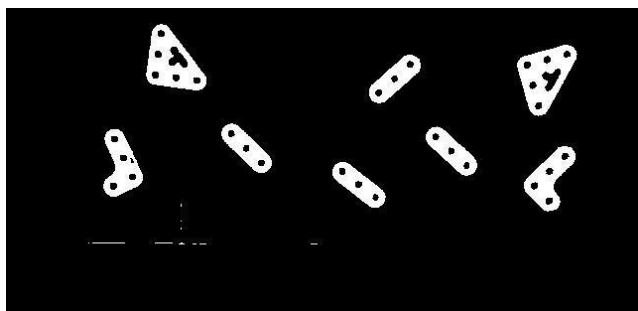
3.3 Diagrama de conexión del electroimán con el relevador

En cuanto al procesamiento de la imagen tomada por la cámara, se hace uso de la herramienta conocida como Vision Assistant proporcionada por Labview que incluye muchas herramientas y librerías que implementan algoritmos de visión. Lo primero que se hace con Vision Assistant es convertir la imagen RGB de 32 bits proporcionada por la webcam a una imagen de 8 bits en escala de grises. Esto es implementado con la extracción del plano HSL de la imagen.



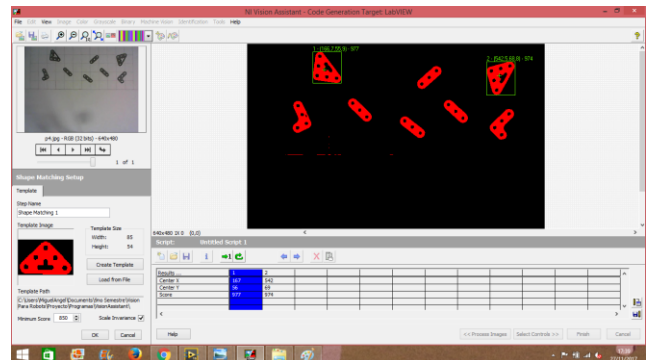
3.4 Extracción del plano HSL y conversión a imagen en escala de grises de 8 bits.

Una vez obtenida la escala de grises, se procede a aplicarle un proceso de binarización en el cual se hace un ajuste del ancho del kernel y del factor de desviación hasta tener resultados óptimos dada la iluminación del lugar donde se colocó el espacio de trabajo.



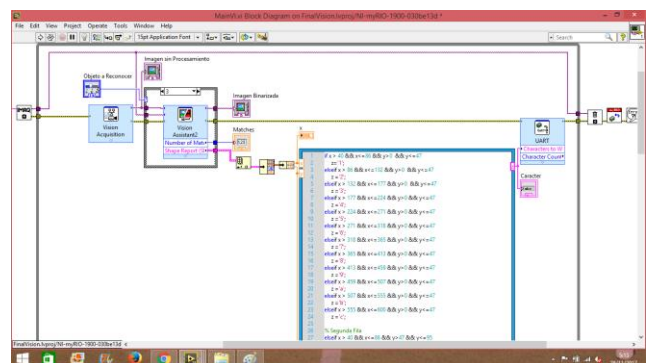
3.5 Imagen Binarizada.

Con una imagen binarizada adecuada, se puede aplicar un algoritmo de detección de formas el cual es entrenado con 'templates' de cada figura particular a detectar y cuyo 'score' de detección mínimo es ajustado para que detecte la figura requerida no importando cuantas muestras con 'match' de esta se encuentren dentro del espacio de trabajo e independientemente de que puedan estar rotadas. El conjunto de estos algoritmos nos da como producto las coordenadas en pixeles de la imagen de los centroides de cada figura a detectar encontrada.



3.6 Matches con coordenadas de sus centroides en base al template proporcionado.

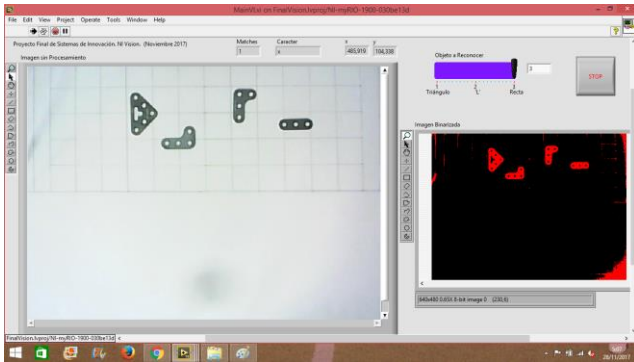
A partir de este algoritmo de procesamiento se genera un Express VI que es incluido dentro de un proyecto para Labview myRIO. El proyecto de Labview prácticamente recibe la imagen de la cámara y en base a una barra de selección de figura a detectar mostrada en la interfaz de usuario, decide que figura detectará el procesamiento implementado en la Express VI generada por Vision Assistant. Una vez obtenidas las coordenadas de los matches obtenidos, entrará a un Mathscript Node que determinará dentro de qué cuadrante se encuentra el centroide de las figuras detectadas y en base a esa información le enviará un dato por UART al Arduino que será interpretado como los valores que deberá darle a los ángulos de los servomotores para colocarse en el centro de ese cuadrante.



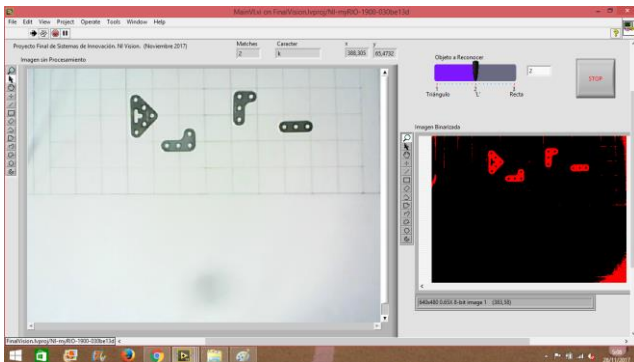
3.7 Código 'G' de Labview

IV. RESULTADOS

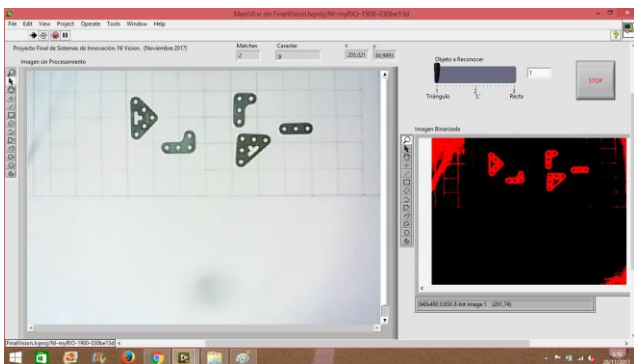
En las figuras 4.1 - 4.3 se puede apreciar la interfaz de usuario en tiempo de ejecución mostrando la imagen captada por la cámara y la imagen binarizada, así como etiquetas con las coordenadas del centroide del match con mayor score detectado y el número actual de matches. La barra morada sirve para especificar el objeto a detectar.



4.1 Detección de Rectas.

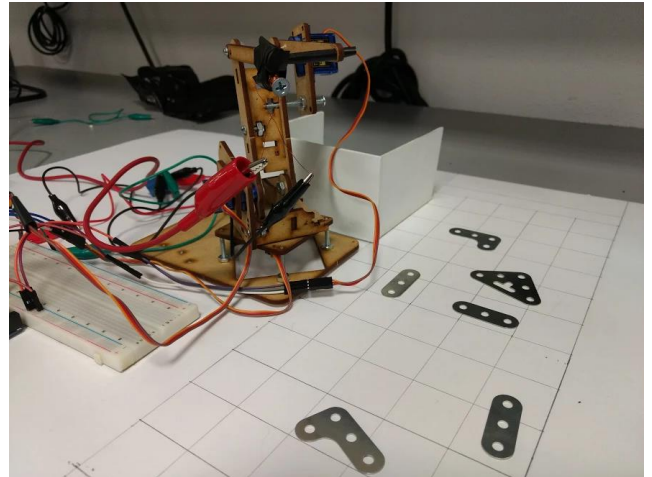


4.2 Detección de 'L's.

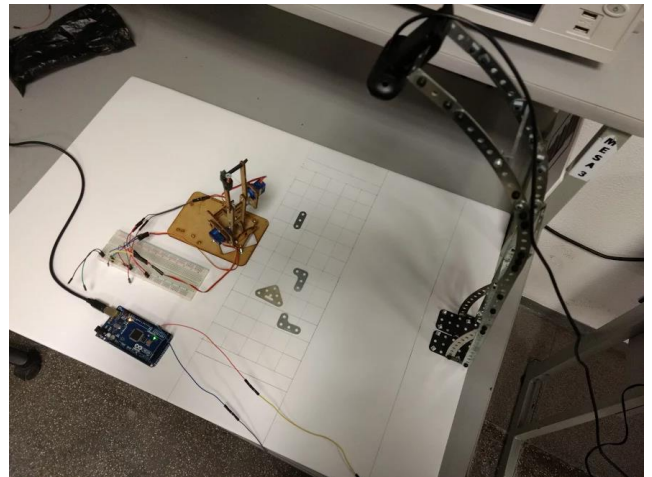


4.3 Detección de Triángulos.

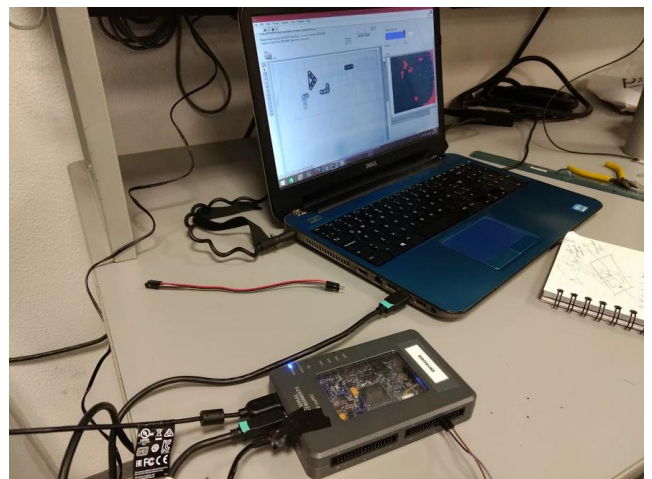
A continuación se muestran imágenes del prototipo con todos sus componentes:



4.4 Brazo robótico.



4.5 Espacio de trabajo.



4.6 MyRIO e Interfaz.

Link de Youtube del Video Demostrativo:

<https://www.youtube.com/watch?v=Ry5U19dfIC0>

V. CONCLUSIÓN

En este proyecto se le pudo dar un uso práctico a las técnicas de procesamiento de imagen digital aprendidas durante el curso de Visión enfocado en este caso a solucionar un problema industrial como lo puede ser la clasificación de componentes en base a un criterio de detección de formas para su posterior depósito.

Para lograr esto se utilizó la conversión de una imagen a escala de grises, técnicas de binarización y algoritmos de detección de formas en base a entrenamiento con templates y ajustes de los 'scores'. La conversión a grayscale no resultó ningún problema ya que consistía solamente en la extracción del plano HSL. En cuanto a la binarización es necesario tener en cuenta los factores de iluminación ya que afecta mucho en el resultado final de la imagen de salida, para esto se realiza una pequeña 'calibración' cada que se monta el prototipo en algún lugar haciendo ajustes al factor de desviación, ancho de kernel y tipo de binarización hasta tener resultados óptimos. Para la detección de formas simplemente se hicieron modificaciones de templates y 'scores'.

Los principales problemas a los que nos enfrentamos en el desarrollo de este proyecto fueron las calibraciones en el algoritmo de binarización para ajustarse a distintas iluminaciones cada que montábamos el dispositivo, el ajuste de la cámara para que nos diera la imagen deseada del espacio de trabajo y el crear un electroimán que nos diera la capacidad de carga y rápida descarga adecuada para la funcionalidad que se buscaba.

VI. REFERENCIAS

- [1]García Morato. (n.d.). BINARIZACIÓN DE IMÁGENES DIGITALES Y SU ALGORITMIA COMO HERRAMIENTA APLICADA A LA ILUSTRACIÓN ENTOMOLÓGICA. Retrieved from <http://sea-entomologia.org/PDF/Boletin53/443464BSEA53BinarizacionRMagro.pdf>
- [2] MATLAB. (n.d.). Retrieved August 26, 2017, from <https://www.mathworks.com/products/matlab.html>
- [3] ARDUINO. (2017). WHAT IS ARDUINO? Retrieved February 19, 2017, from <https://www.arduino.cc/>
- [4]NI myRIO and LabVIEW. (2017). Retrieved March 31, 2017, from <http://www.ni.com/en-us.html>