

CLASIFICATION SYSTEM BASED ON COMPUTER VISION

José Porras

jose_music86@hotmail.com

Miguel De la Cruz (†)

Asesor:

Profesor Antonio Morán

amoran@mail.urp.edu.pe

Curso: CE 1002 Taller de Electrónica IV

Escuela Profesional de Ingeniería Electrónica
Universidad Ricardo Palma

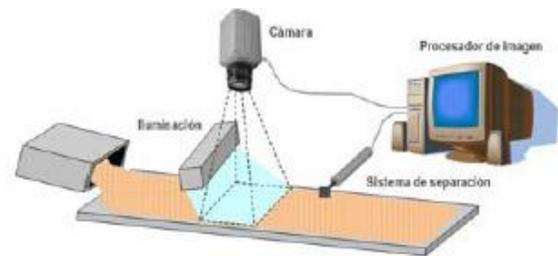
RESUMEN: El presente proyecto trata de un sistema automatizado controlado por visión artificial, utilizando hardware (sensor de luz, control de motor a pasos) y software (control por MATLAB). El principio básico de funcionamiento es transportar objetos de color (rojo, verde y azul) por una faja transportadora, detectarlos con un sensor, capturar la imagen con una cámara web común y procesarla por MATLAB para clasificarlos en tres distintos contenedores. Con este proceso se logró una clasificación de objetos eficaz tanto en color y forma.

SUMMARY: This project is an automated system controlled by computer vision, using hardware (light sensor, stepper motor control) and software (control by MATLAB). The basic principle of operation is to transport objects (red, green and blue) by a conveyor belt, a detecting sensor, image capture web camera and processed by MATLAB common to classify them in three different containers. This process achieved an efficient classification of objects in color and shape.

1 INTRODUCCIÓN

La visión artificial trata de simular el proceso del sentido de la visión de los seres vivos, según el cual un objeto es captado mediante los receptores de la retina y es transformado en impulsos nerviosos que son procesados por el cerebro.

La visión artificial consiste en la captación de imágenes en línea mediante cámaras basadas en matrices de sensores sensibles a la luz (CCD o CMOS), el posterior tratamiento de las mismas mediante técnicas de análisis de imagen y la actuación sobre el proceso (control de producto) o el producto (control de calidad).



Las principales ventajas son la reducción de los costes del control de calidad, el aumento de la producción, la inspección del 100% de la producción, la reproducibilidad y el funcionamiento 24 horas al día sin "cansancio".

1.1. Elementos de un Sistema de Visión Artificial

Los elementos básicos son los siguientes:

- El sistema de **iluminación** (fluorescente, LED, polarizada, backligth, láser)
- El sensor o **cámara** de captura de imagen (CMOS), CCD, InGaAs)
- **Tarjeta de captura** de elevado ancho de banda (Firewire, CameraLINK, GbEthernet)
- **Procesamiento de imagen** (Procesador y algoritmos de pretratamiento y filtrado de la imagen, de segmentación y reconocimiento de formas, de extracción de descriptores y de clasificación)
- **Sincronía** con el proceso, para adquirir la imagen en el momento adecuado y para actuar con el proceso o separar unidades defectuosas.

1.2. Principio de Funcionamiento:

La visión artificial, también conocida como visión por computador (del inglés computer vision) o visión técnica, es un subcampo de la

inteligencia artificial. El propósito de la visión artificial es programar un computador para que "entienda" una escena o las características de una imagen.

Generalmente la visión artificial se divide en seis áreas:

- **Captación:** Proceso de obtener una imagen. El proceso de captura está centrado en tres aspectos importantes, necesarios para obtener una imagen de calidad que no altere las condiciones reales agregando errores considerables al sistema.
- **Pre procesamiento:** Métodos para quitar o reducir características no deseadas en la imagen como el ruido. El pre-procesamiento consiste en la aplicación de técnicas que permitan el realce o mejoramiento de algunas características importantes en las imágenes originales para facilitar el proceso de segmentación.
- **Segmentación:** Dividir una imagen en objetos que sean de nuestro interés. La técnica de segmentación más conocida dentro del procesamiento de imágenes es quizás la umbralización. Este método permite separar dos o más regiones de una imagen a partir de un análisis del histograma.
- **Descripción:** Obtención de características de un objeto como su forma, tamaño, color con el fin de diferenciarlo de otros objetos.
- **Reconocimiento:** Proceso a través del cual se identifica un objeto dentro de una escena.
- **Interpretación:** Asociar un significado a un conjunto de objetos reconocidos.

1.3. Aplicaciones

- Automatizar procesos de control de calidad: detección de defectos y fallos en materia prima producto final o en el envase.
- Automatizar procesos de clasificación en base a la forma, tamaño o color en materias primas muy variables.
- Automatizar procesos de producción: ajustar las condiciones de proceso en base a variables medidas mediante análisis de imagen.
- Automatización de procesos de manipulación y envasado: extracción de coordenadas para su posterior manipulación mediante robots.
- Automatizar procesos de detección de cuerpos extraños.

- En la medicina aparece el procesamiento de imágenes, a menudo orientadas hacia el diagnóstico de dolencias o enfermedades, entre las que se incluyen radiografías, resonancias magnéticas, tomografías etc.
- Navegación en Robótica, la visión es un elemento de un sistema multisensorial. La información procedente de la visión es validada, comparada y finalmente integrada con el resto de la información proporcionada por otro tipo de sensores. El resultado es la reconstrucción de la escena 3-D, que permite la navegación autónoma del sistema

2 PRESENTACIÓN DEL PROBLEMA

El problema del que se ocupa este trabajo, es el de ubicar objetos según sus características en diversos contenedores (clasificación), con el fin de minimizar costos de operación en una planta de producción industrial.

3 DESCRIPCIÓN DE LA SOLUCIÓN

El proyecto busca implementar una pequeña planta industrial para el proceso de control de calidad.

Hoy en día se ha facilitado este trabajo mediante sistemas inteligentes que hagan esto por el hombre, así evitar la fatiga humana de verificar cada elemento uno por uno lo que sería muy tedioso.

La metodología utilizada para encontrar la solución se basa en dos bloques que son hardware y software.

En esta investigación, se utilizan como herramientas de creación sensores, motores y programación en el software de ingeniería MATLAB, estas partes serán descritas detalladamente en este documento.

Con este proyecto solo bastará con una persona la cual tendrá como función accionar el sistema para que opere automáticamente, claro está que este sistema puede modificar algunos de sus algoritmos internos (software) para así hacerlo trabajar al modo del operario.

Principalmente se hará que este sistema implementado clasifique objetos según color y forma.

El proyecto constará de dos partes básicas para su funcionamiento:

3.1. Hardware:

a) Faja transportadora:

La faja servirá de transporte para los objetos que posteriormente serán seleccionados, esta constará de un motor DC, Pequeños rodillos en los extremos, una banda. La velocidad de la faja tendrá que ser controlada a una velocidad adecuada que de tiempo al procesador detectar el objeto y seleccionarlo. Véase Figura 1.

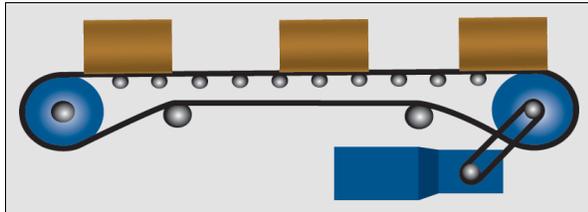


Figura 1. Faja Transportadora

b) Circuito Sensor de Luz:

Consta un sensor LDR, un PIC16F628, MAX232N:

Sensor LDR:

Si bien los valores que puede tomar una LDR en total oscuridad y a plena luz puede variar un poco de un modelo a otro, en general oscilan entre unos 50 a 1000 ohmios (1K) cuando están iluminadas (por ejemplo, con luz solar) y valores comprendidos entre 50K (50,000 Ohms) y varios megohmios (millones de ohms) cuando está a oscuras.

Desde el punto de vista constructivo, las LDR están fabricadas con materiales de estructura cristalina, siendo los más utilizados el sulfuro de cadmio y el seleniuro de cadmio, aprovechando sus propiedades fotoconductoras. Véase Figura 2.

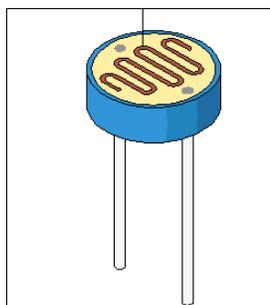


Figura 2. Sensor LDR

Integrado HEF40106BP:

La función de este circuito integrado es la de convertir la señal analógica del LDR a señal digital para el PIC16F628.

Comunicación PIC16F628 con MAX232:

El MAX232 es el circuito integrado estándar para convertir señales TTL/CMOS a señales RS232. Las señales en RS232 tienen 1's y 0's estos son +12V y -12V respectivamente. Las señales de salida del PIC16F628 son entre 0 y 5V.

Lo que el MAX232 hace es poner 12V en el pin T1OUT cuando se alimenta con 5V el pin T1IN. De esta forma se pueden pasar datos hacia tu computador. Si presionas una tecla en el hyperterminal una señal será enviada a través del cable hasta el R1IN del MAX232 donde la señal de 12V proveniente del PC es convertida a 0-5V. Esta señal que sale por el R1OUT es perfectamente manejable por el PIC. Todo OK. La magia está en cómo el MAX232, el cual es alimentado con 5V y puede producir señales de +/-12V. Bueno, esto se hace con condensadores externos. Véase Figura 3.

El diagrama de flujo de esta parte del hardware se muestra en la Figura 4.

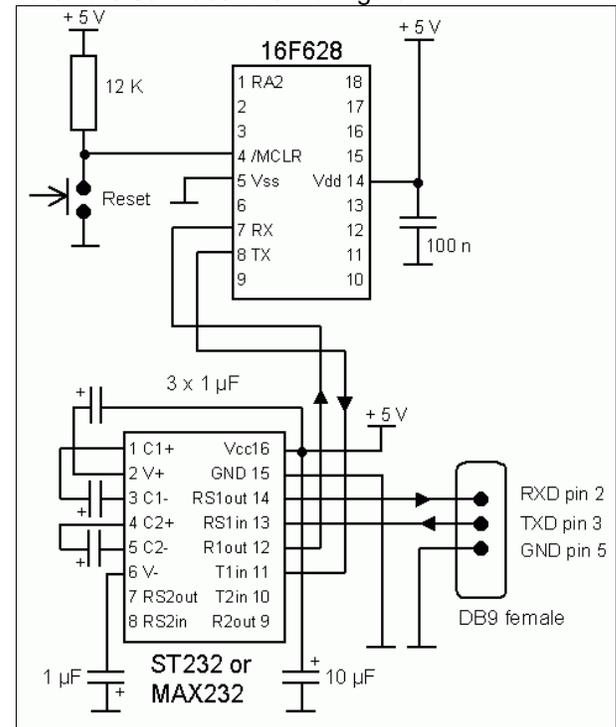


Figura 3. Comunicación PIC16F628 con MAX232

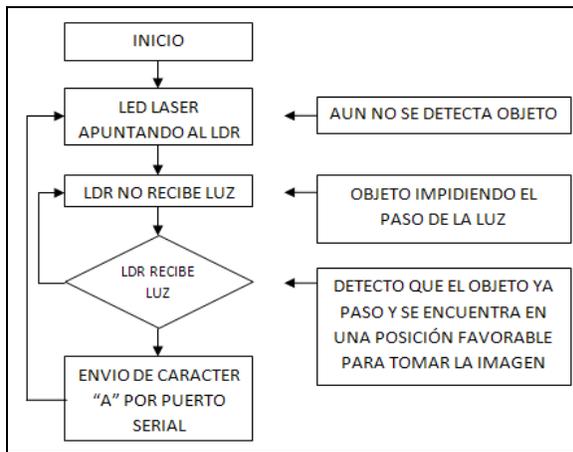


Figura 4. Diagrama de Flujo Sensor de Luz

La programación del PIC se mostrará en la parte de anexos en la sección 1.

c) Cámara Web:

Se utilizará una cámara web común y corriente para la etapa de captura de imagen de los objetos a ser clasificados.

En la Figura 5 se muestra el diagrama de flujo de la captura de imagen.

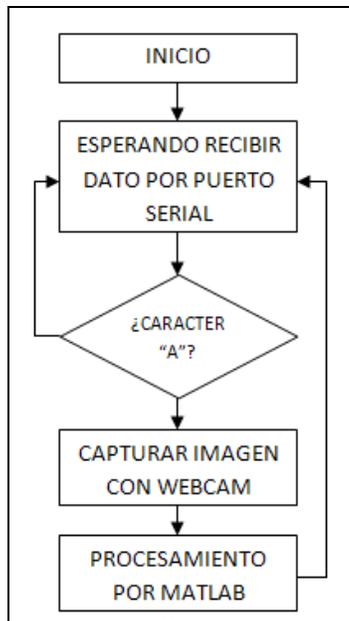


Figura 5. Diagrama de Flujo Captura de la Imagen

c) Circuito del Motor a Pasos:

Constará de las siguientes partes:

Comunicación MAX232 con PIC 16F628:

Esta parte circuital (Figura 3) es la que se encargará de convertir los datos que vienen del

puerto serial (RS232) a niveles TTL para el PIC 16F628.

La programación del PIC se muestra en la parte de anexos en la sección 2.

A continuación en la Figura 6 se presenta el diagrama de flujo para el control del giro del motor en tres posiciones distintas.

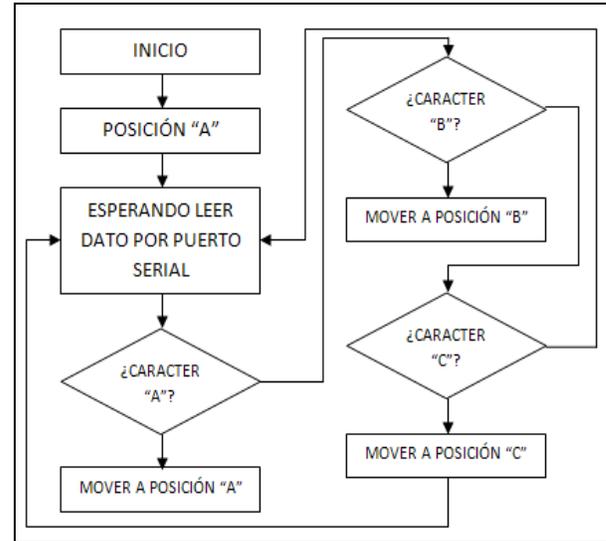


Figura 6. Diagrama de Flujo Control del Giro del Motor a Pasos

Driver para el motor a pasos:

Se utilizó el L297 junto con transistores de potencia Darlington NPN (TIP122), la salida del PIC 16F628 va conectada al L297, luego las salidas de este irán hacia los transistores para conectarse al motor a pasos. Véase Figura 7.

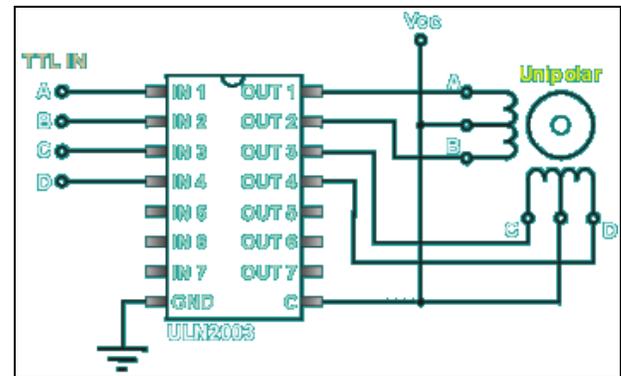


Figura 7. Circuito para Motor a Pasos

3.1. Software:

Su misión consiste en aplicar las transformaciones necesarias y extracciones de información de las imágenes capturadas.

Utilizaremos el software de ingeniería "MATLAB" junto con la librería "vfm.dll" para captura y procesamiento de imágenes.

Herramienta "vfm.dll":

Esta herramienta permite utilizar una cámara web dentro del Matlab.

Mientras la faja guía los objetos llegará un momento en que el sensor de luz detecte uno, para la captura de imagen mediante la cámara web.

Se sabe que una cámara web toma alrededor de 12 cuadros por segundo, si se tiene un sensor de luz que avise cuando se tiene el objeto en una posición adecuada bastará con tomar como imagen solamente un cuadro para que se procese dentro del computador.

Se diseñó dos programas, uno para detectar colores y otro para las formas de los objetos.

En la Figura 8 se mostrará el diagrama de flujo para el proceso de selección según el color; mientras que en la Figura 9 se mostrará el diagrama de flujo para la clasificación del objeto según sea su forma.

En la parte de anexos en la sección 3 y 4 se mostrará la programación correspondiente.

Para la selección de formas se utilizó una relación matemática la cual es: $[\text{perímetro}^2]/[\text{area}]$; esta relación dará como resultado un cociente (número) el cual se le asociará a cada forma (cuadrado, triángulo, círculo).

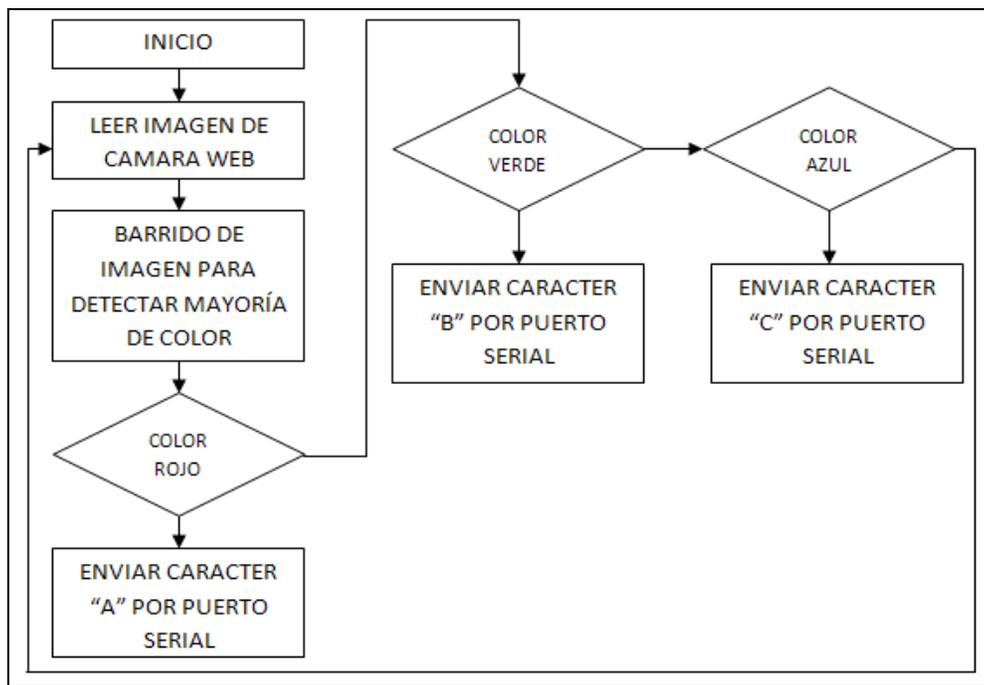


Figura 8. Diagrama de Flujo para Clasificación por Colores

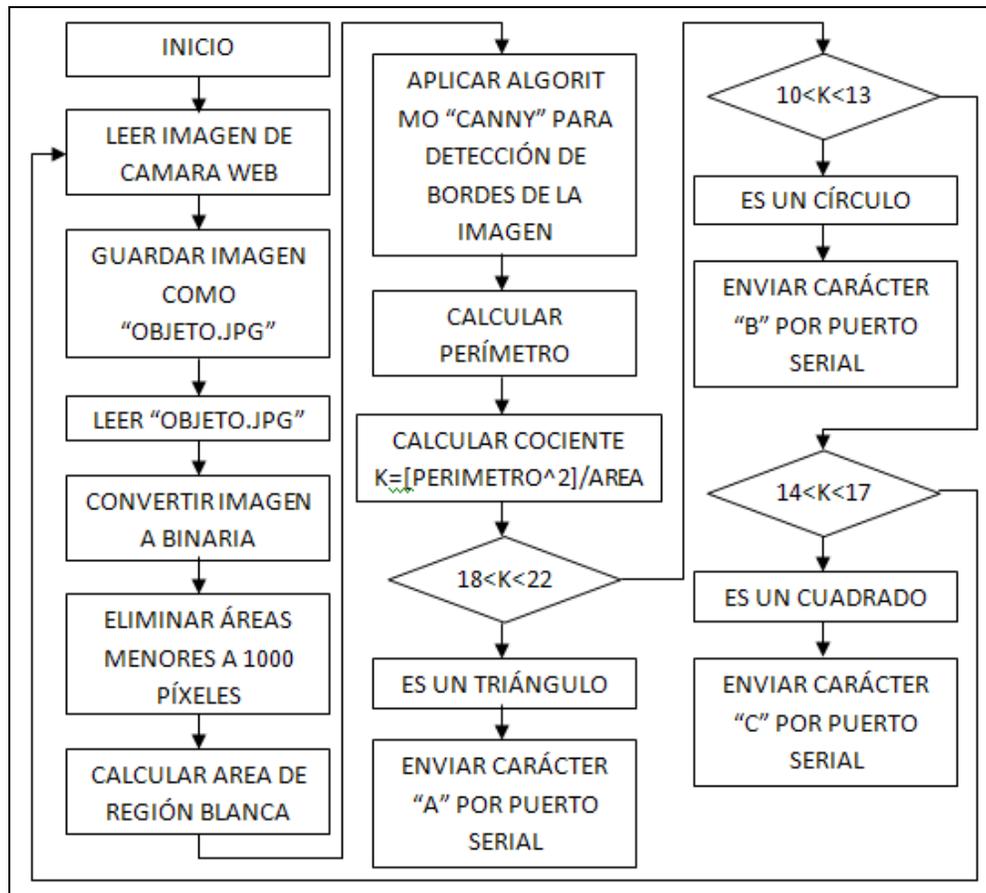


Figura 9. Diagrama de Flujo para Clasificación por Formas

En el anexo sección 5 se muestra el desarrollo matemático para hallar el cociente “K” ya mencionado.

4 RESULTADOS

La faja transportadora fue diseñada para el paso de los objetos, tiene montada encima la cámara web que capturaré las imágenes.

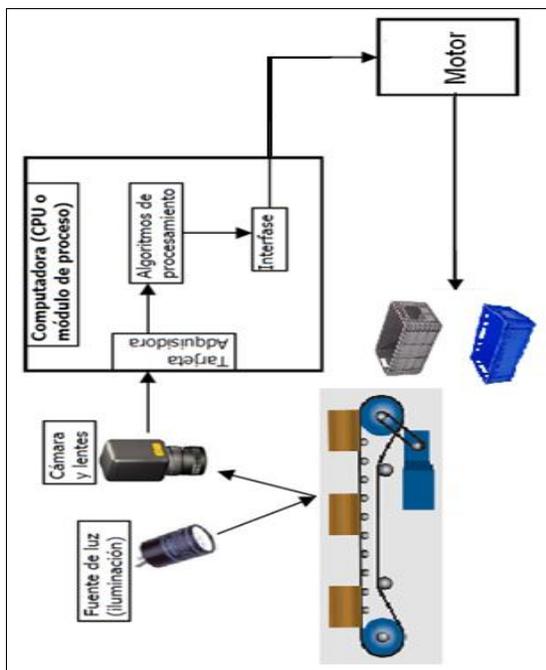


Figura 10. Esquema



Figura 11. Diseño de la faja transportadora

El circuito sensor se muestra a continuación, el cual su explicación ya se mencionó anteriormente.

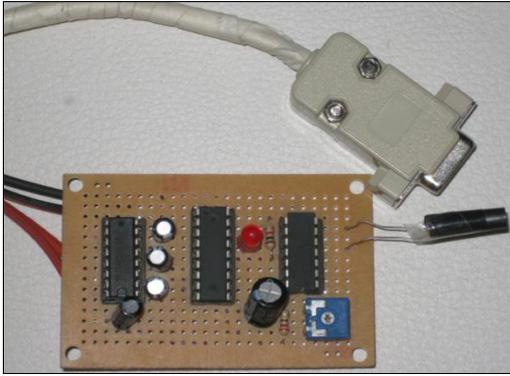


Figura 12. Diseño del circuito sensor

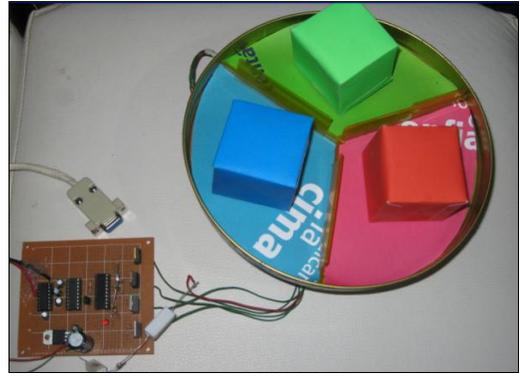


Figura 15. Contenedores con objetos

El circuito para el motor a pasos tendrá como función la de controlar el giro de los contenedores.

Se muestra como se hará el proceso de sensado y captura del objeto, como se ve en la siguiente foto, en el momento que el objeto atraviese por completo el sensor de luz la cámara web tomará la imagen del objeto para enviarla a la PC.

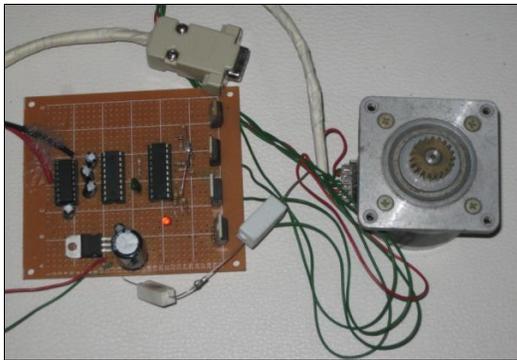


Figura 13. Diseño circuito motor a pasos

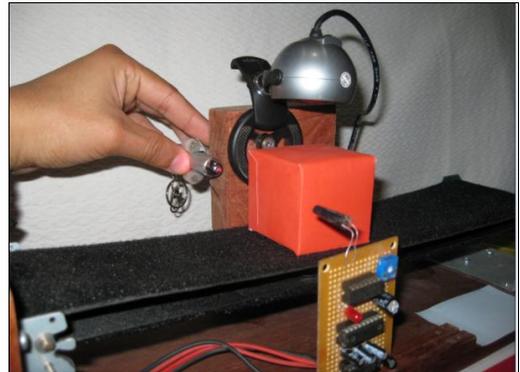


Figura 16. Censado y Captura del Objeto

A continuación se muestran las imágenes del contenedor puesto encima del motor a pasos.

En las siguientes imágenes se muestra la unión de cada parte diseñada, este esquema en su totalidad cumple el objetivo por el que fue construido, el llegar a controlar y clasificar de forma automática los objetos que pasan por la faja.

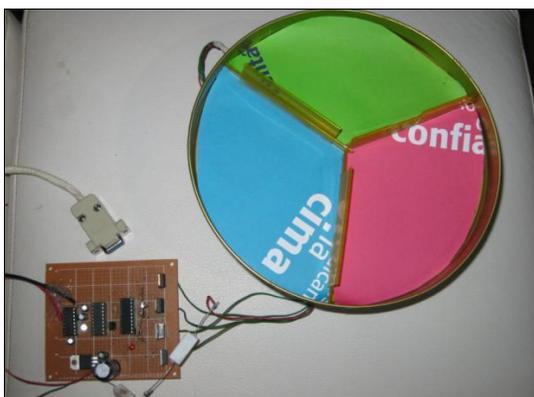


Figura 14. Contenedores



Figura 17. Esquema completo

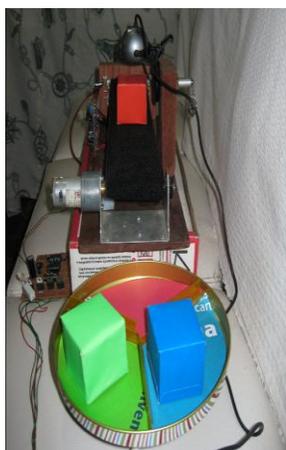


Figura 18. Esquema completo

Resultados para la detección de formas:

Cuadrado:

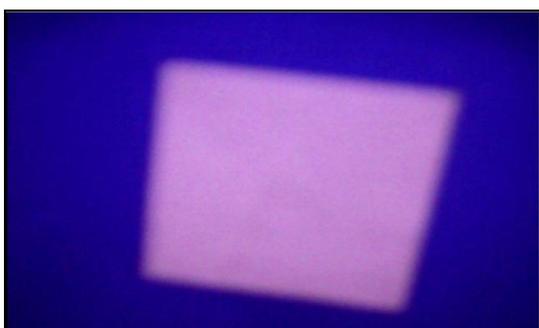
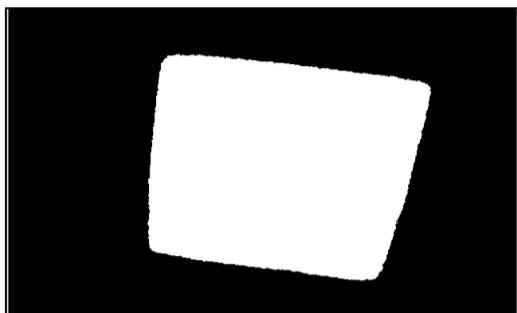
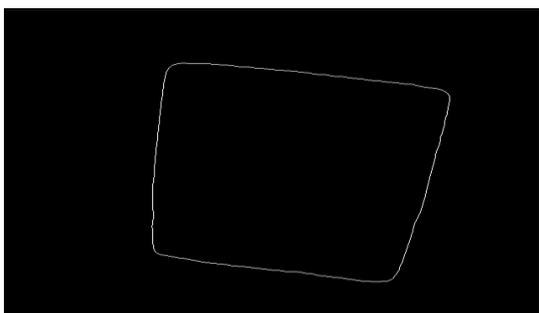


Imagen original tomada con la WebCam



Convirtiendo la imagen a binaria



Aplicando Filtro "Canny" para detección de bordes

Cociente hallado con MATLAB:

```
cociente =
15.4958
```

Triángulo:

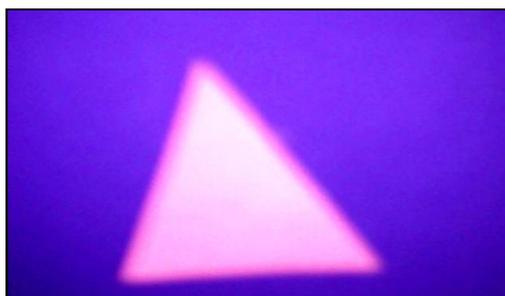
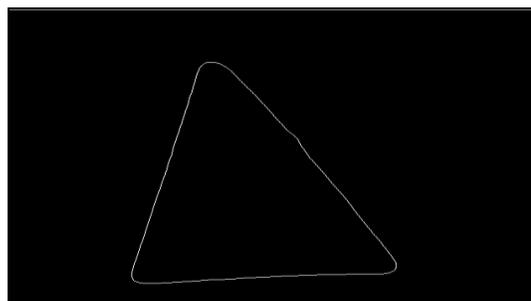


Imagen original tomada con la WebCam



Convirtiendo la imagen a binaria



Aplicando Filtro "Canny" para detección de bordes

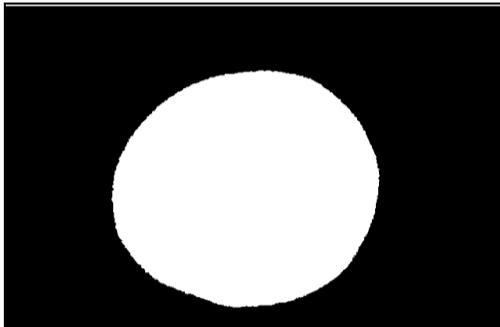
Cociente hallado con MATLAB:

```
cociente =
18.9302
```

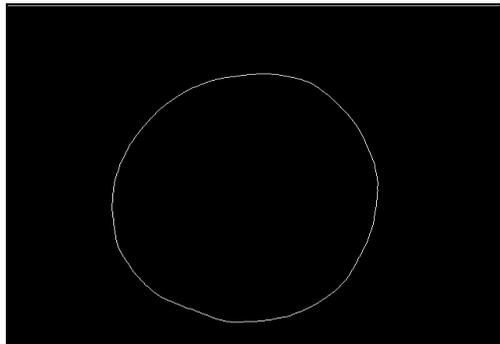
Círculo:



Imagen original tomada con la WebCam



Convirtiendo la imagen a binaria



Aplicando Filtro "Canny" para detección de bordes

Cociente hallado con MATLAB:

`cociente =`

`12.5137`

5 ANEXOS

Sección 1: Programación del PIC16F628 para el Sensor de Luz

```
-----  
;----PROGRAMA PRINCIPAL-----  
-----  
INICIO
```

```
call RS232_Inicializa
```

```
LEE_LDR  
btfsc PORTA,5  
goto $-1  
call Retardo_100ms  
btfss PORTA,5  
goto $-1  
movlw 'A'  
call RS232_EnviaDato  
goto LEE_LDR
```

```
INCLUDE RS232_12F.INC  
INCLUDE RETARDOS.INC  
END
```

Sección 2: Programación del PIC16F628 para el Control de movimiento del Motor a Pasos

```
INICIO  
-----  
POSICION_A  
call LEE_DATO ;ESCAÑA EL DATO SERIAL  
Y REGRESA CON EL NUMERO D PASOS  
A DAR  
movwf VAR_DATO  
movf VAR_DATO,0  
sublw 'B'  
btfsc STATUS,2  
goto P_B_A  
movf VAR_DATO,0  
sublw 'C'  
btfsc STATUS,2  
goto P_C_A  
goto POSICION_A  
P_B_A  
A_HOR  
call GIRA_MOTOR  
goto POSICION_B  
P_C_A  
HOR  
call GIRA_MOTOR  
goto POSICION_C  
-----
```

```

POSICION_B
    call LEE_DATO ;ESCANEA EL DATO SERIAL
    Y REGRESA CON EL NUMERO D PASOS
    A DAR
    movwf VAR_DATO
    movf  VAR_DATO,0
    sublw 'A'
    btfsc STATUS,2
    goto  P_A_B
    movf  VAR_DATO,0
    sublw 'C'
    btfsc STATUS,2
    goto  P_C_B
    goto  POSICION_B
P_C_B
    A_HOR
    call  GIRA_MOTOR
    goto  POSICION_C
P_A_B
    HOR
    call  GIRA_MOTOR
    goto  POSICION_A
.....
POSICION_C
    call LEE_DATO ;ESCANEA EL DATO SERIAL
    Y REGRESA CON EL NUMERO D PASOS
    A DAR
    movwf VAR_DATO
    movf  VAR_DATO,0
    sublw 'B'
    btfsc STATUS,2
    goto  P_B_C

```

```

    movf  VAR_DATO,0
    sublw 'A'
    btfsc STATUS,2
    goto  P_A_C
    goto  POSICION_C
P_A_C
    A_HOR
    call  GIRA_MOTOR
    goto  POSICION_A
    P_B_C
    HOR
    call  GIRA_MOTOR
    goto  POSICION_B
    .....
    GIRA_MOTOR
    movlw .67
    movwf VAR_PULSO
    GIRA_MOTOR_1
    bsf          PORTB,1
    ;CLK
    call  Retardo_20ms ;PAUSA
    bcf          PORTB,1
    ;CLK
    call  Retardo_20ms ;PAUSA
    decfsz VAR_PULSO,1
    goto  GIRA_MOTOR_1
    return
    LEE_DATO
    call  RS232_LeeDato
    return
    INCLUDE     RS232_12F.INC
    INCLUDE     RETARDOS.INC
END

```

Sección 3: Programación en MATLAB para Clasificación por Colores

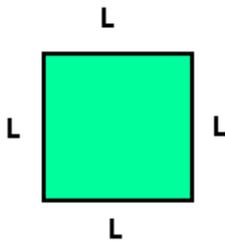
```
Editor - C:\MATLAB7\work\colores.m*
File Edit Text Cell Tools Debug Desktop Window Help
Stack Base
1 % cerrar puerto serial si ya está abierto
2 - warning off MATLAB:serial:fread:unsuccessfulRead
3 s = instrfind;
4 if ~isempty( s )
5     fclose( s );
6     delete( s );
7     clear s
8 end
9 % crear puerto serial
10 s = serial( 'COM1','BaudRate', 9600 );
11 s.InputBufferSize = 5000;
12 % abrir puerto serial
13 fopen( s );
14 for cc = 1:10
15     pause( 1 );% definir tiempo de captura en segundos
16     fprintf(s,'G')
17     out = fread(s,1);
18     if ( out == 65 )
19         disp ( 'Objeto detectado' )
20         %PRUEBA CON VFM
21         NUMCUADROS = 1;
22         K = VFM('grab',NUMCUADROS);
23         size(K);
24         image(K);
25         subplot(221); image(K);
26         R = K(:, :,1);
27         G = K(:, :,2);
28         B = K(:, :,3);
29         colormap(gray(256)), subplot(222), image( K(:, :,1) );
30         colormap(gray(256)), subplot(223), image( K(:, :,2) );
31         colormap(gray(256)), subplot(224), image( K(:, :,3) );
32         mR = sum(sum(R));
33         mG = sum(sum(G));
34         mB = sum(sum(B));
35         if( (mR > mG) & (mR > mB) )
36             disp('El Color es Rojo');
37             fprintf(s,'A');
38         elseif( (mG > mR) & (mG > mB) )
39             disp('El Color es Verde');
40             fprintf(s,'B');
41         elseif( (mB > mR) & (mB > mG) )
42             disp('El Color es Azul');
43             fprintf(s,'C');
44         end
45     end
46 end
47 fclose( s );
48 delete( s );
49 clear s
50
objetos.m* x colores.m* x
script Ln 43 Col 24 OVR
```

Sección 4: Programación en MATLAB para Clasificación por Formas

```
Editor - C:\MATLAB7\work\objetos.m*
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base
1 % cerrar puerto serial si ya está abierto
2 - warning off MATLAB:serial:fread:unsuccessfulRead
3 s = instrfind;
4 if ~isempty( s )
5     fclose( s );
6     delete( s );
7     clear s
8 end
9 s = serial( 'COM1','BaudRate', 9600 );% crear puerto serial
10 s.InputBufferSize = 5000;
11 fopen( s );% abrir puerto serial
12 for cc = 1:10
13     pause( 1 );% definir tiempo de captura en segundos
14     fprintf(s,'G')
15     out = fread(s,1);
16     if ( out == 65 )
17         disp ( 'Objeto detectado' )
18         k=vfm('grab',1);
19         imshow(k);
20         imwrite(k,'objeto.jpg'); %% Guarda el primer cuadro de la webcam como objeto.jpg
21         a=imread('objeto.jpg'); %% Lectura de objeto.jpg
22         bn=im2bw(a);%% Convertir a blanco y negro la imagen
23         imshow(bn);%% Mostrar imagen
24         bn=bwareaopen(bn,1000); %% Eliminar áreas menores a 1000 pixels
25         imshow(bn);
26         area = bwarea(bn);%% Area total del objeto
27         [g,tg] = edge(bn,'canny',[0.09 0.12],0.3);%% Asignando bordes a la imagen
28         imshow(g);
29         perimetro = bwarea(g);%% Calculando el perimetro
30         k=perimetro^2;
31         cociente=k/area%% Cociente
32         %% Determinando que figura es
33         if ((18<cociente) & (cociente<22))
34             disp('El objeto es un triangulo');
35             fprintf(s,'A');%Envio de caracter "A" por puerto serial
36         elseif ((10<cociente) & (cociente<13))
37             disp('El objeto es un circulo');
38             fprintf(s,'B');%Envio de caracter "B" por puerto serial
39         elseif ((14<cociente) & (cociente<17))
40             disp('El objeto es un cuadrado');
41             fprintf(s,'C');%Envio de caracter "C" por puerto serial
42         end
43     end
44 end
45 fclose( s );
46 delete( s );
47 clear s
48
objetos.m* x colores.m* x
script Ln 41 Col 68 OVR
```

Sección 5: Desarrollo Matemático para Clasificación por Formas

Analizando para un cuadrado:

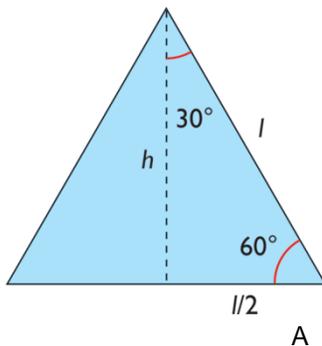


Área del cuadrado: $A = L^2$

Perímetro: $P = 4L$

Cociente (K): $\frac{P^2}{A} = 16$

Analizando para un triángulo equilátero:



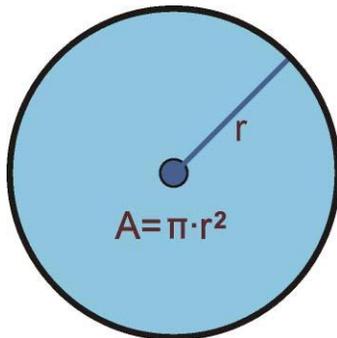
$$h = \frac{\sqrt{3}}{2} l$$

Área del triángulo: $A = \frac{\sqrt{3}}{4} l^2$

Perímetro: $P = 3l$

Cociente (K): $\frac{P^2}{A} = 20.78460969$

Analizando para un círculo:



Área del círculo: $A = \pi r^2$

Perímetro: $P = 2\pi r$

Cociente (K): $\frac{P^2}{A} = 12.56637061$

Ahora definimos intervalos como se explica en la programación de MATLAB para que el computador tenga conocimiento de qué forma está procesando:

Para un cuadrado:

$$14 < \text{cociente} < 17$$

Para un triángulo:

$$18 < \text{cociente} < 22$$

Para un círculo:

6 CONCLUSIONES

- La técnica de Control de calidad con visión artificial permite una inspección rápida y una automatización del 100 % de la producción.
- Con esta técnica se obtiene una reducción de costes en control de calidad y a su vez minimiza el envío de productos defectuosos a los clientes.
- En la actualidad los costes de los sistemas de visión artificial han disminuido enormemente mientras que las prestaciones de esta técnica han aumentada espectacularmente.
- La visión artificial es una herramienta para automatizar procesos de control, supervisión, manipulación y robotización.
- El sistema construido cumplió con las expectativas requeridas, lográndose así un proceso automatizado eficaz.

7 BIBLIOGRAFÍA

- Raúl Pino Díez, Alberto Gómez Gómez, Nicolás de Abajo Martínez, " Introducción a la inteligencia artificial: Sistemas expertos, redes neuronales artificiales y computación evolutiva", Universidad de Oviedo, 2001 - 106 páginas.
- Universidad Rey Juan Carlos (Madrid), "Aplicaciones de la visión artificial y la biometría informática", Librería-Editorial Dykinson, 2005 - 128 páginas.

Enlaces Web Consultados:

- <http://gavab.escet.urjc.es/master/visionartificial/material/apuntes/tema1.ppt>
- <http://serdis.dis.ulpgc.es/~a013775/asignaturas/itis-fia/T1bn.pdf>
- <http://www.logicelectronic.com/vision/quey porque.htm>
- <http://www.jasvisio.com/control-de-calidad-industrial-con-vision-artificial.html>

- <http://redalyc.uaemex.mx/pdf/477/47711103.pdf>
- <http://www.depi.itchihuahua.edu.mx/electro/archivo/electro2001/mem2001/articulos/img2.pdf>
- http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/garcia_b_ci/capitulo2.pdf
- http://es.wikipedia.org/wiki/Visi%C3%B3n_artificial
- <http://www.tupperbot.es/wp-content/uploads/2007/01/visionartificial.pdf>
- <http://mx.reocities.com/CapeCanaveral/8482/docu003.pdf>
- http://dmi.uib.es/~ygonzalez/VI/Material_de_L_Curso/Teoria/Aplicaciones_VC.PDF
- <http://www2.unalmed.edu.co/dyna2005/133/vision.pdf>
- <http://www.neoteo.com/ldr-fotorresistencias>
- http://robots-argentina.com.ar/Prueba_PIC628-RS232.htm
- <http://www.olimex.cl/tutorial/tutorial2.pdf>