Articulo **Revista de Prototipos Tecnológicos**
Marzo, 2017 Vol. 3 No.7 1-6

Detección de Objetos a Color en Tiempo Real con Técnicas de Visión Artifical y Arduino

FLORES-MONTES, Miguel Ángel†*, MEDINA-MUÑOZ, Luis Arturo, MAYORQUÍN-ROBLES, Jesús, GARCÍA-MUÑOZ, Omar Adrián

Instituto Tecnológico de Nogales, Departamento de Posgrado e Investigación
Instituto Tecnológico de Ensenada

Recibido Agosto 03, 2016; Aceptado Febrero 15, 2017

Resumen

Este articulo muestra información de la aplicación de visión artificial utilizando la plataforma Arduino Uno R3 y MATLAB 2011, describiendo el procedimiento a seguir para utilizar el hardware y software necesario para el procesamiento de imágenes utilizando dispositivos comunes como una cámara web y una PC. Trabajos futuros implica aplicar estos conocimientos en drones para realizar reconocimiento de patrones.

Visión artificial, detección de objetos, reconocimiento de patrones

Abstract

This article provide information on artificial vision and MATLAB 2011, describing procedures to use hardware and software necessary for process images with common devices like webcam and PC. Futures works are to apply this knowledge in drones to do patterns recognition.

Artificial vision, object detection, pattern recognition

Citación: FLORES-MONTES, Miguel Ángel, MEDINA-MUÑOZ, Luis Arturo, MAYORQUÍN-ROBLES, Jesús, GARCÍA-MUÑOZ, Omar Adrián. Detección de Objetos a Color en Tiempo Real con Técnicas de Visión Artifical y Arduino. Revista de Prototipos Tecnológicos. 2017 3-7: 1-6

† Investigador contribuyendo como primer autor.
*Correspondencia al autor (Correo electrónico: angel.m@depiitn.edu.mx)

Introducción

Este trabajo consiste en la detección de objetos a color (Rojo, Verde y Azul) utilizando las imágenes obtenidas en tiempo real con una cámara web, permitiendo realizar el procesamiento de imágenes para marcar cada una de los objetos con su respectivo centroide, coordenadas, y mostrar también en tiempo real la cantidad de objetos detectadas de cada uno de los colores RGB (Rojo, Verde y Azul), al mismo tiempo que encuentra los objetos se desplegarán señales en 3 diferentes leds de colores RGB (Rojo, Verde y Azul), con el objetivo de indicar de manera visual y física que ha detectado algún color especificado.

La implementación de este sistema esta sujeta a ambientes controlados en donde los objetos a detectar posean matices con nitidez alta. La sección 2 habla de las plataformas utilizadas y sus características al igual del circuito. En la sección 3 se describen los pasos para llevar acabo el procesamiento de imagen para la detección de colores RGB dentro de la captura de video. La sección 4 se muestra los resultados obtenidos en la detección de objetos a color con un conteo individual y total.

La sección 5 cuenta con agradecimientos, en la sección 6 se habla de la conclusión de este trabajo y por ultimo en la sección 7 se encuentran las referencias utilizadas.

Sincronización entre diferentes plataformas

Plataformas utilizadas

Arduino [2], una plataforma electrónica de código abierto tanto de software como hardware permite de una forma sencilla realizar proyectos interactivos. Existe una gran cantidad de placas, sensores y actuadores compatibles.
Una tendencia tecnológica es utilizar la placa de Arduino como tarjeta de adquisición de datos desarrollando interfaces con diferentes entornos tecnológicos.

MATLAB [1], una herramienta de software matemático con un lenguaje de programación propio, permite la comunicación con programas en otros lenguajes y diferentes dispositivos hardware como es el caso de Arduino.

Arduino Support from MATLAB [1] permite conectar y controlar la placa Arduino con MATLAB, facilitándonos el desarrollo del proyecto, como lo es en este caso utilizar visión artificial para la detección de objetos de color en tiempo real y desplegar señales digitales de salida en Arduino.

Circuito

El circuito mostrado en la figura 1, está compuesto por 3 leds de color Rojo, Verde y Azul, con 3 resistencias de 250 ohm para cada uno de los leds, conectados a la placa de Arduino Uno R3.

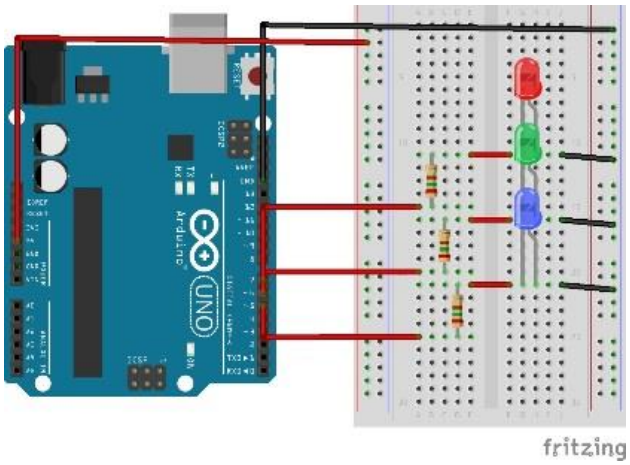

Figura 1 Circuito con Arduino

Este circuito nos permitirá ver brillar a los leds correspondientes cada vez que la imagen obtenida en tiempo real detecte un objeto de color Rojo, Verde o Azul.

Procesamiento de imágenes

En la siguiente figura se muestra el diagrama de flujo representativo del proceso de captura de imágenes del video en tiempo real y su correspondiente tratado de fotogramas y despliegue de señal digital en arduino.

ISSN 2444-4995

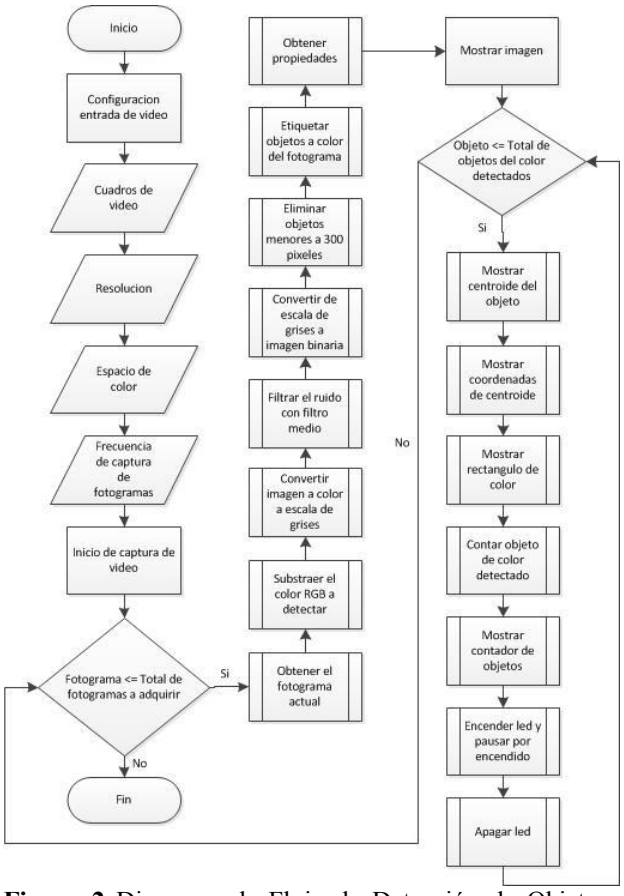

Figura 2 Diagrama de Flujo de Detección de Objetos a Color RGB

Captura de video en tiempo real

El primer paso en MATLAB para utilizar una cámara web en la detección de objetos en tiempo real, es, asignar el formato de la cámara, al igual que la captura de los cuadros que corren en tiempo real y su resolución (en este caso 640x480 pixeles); es necesario también especificar en la configuración el espacio de color que se desea utilizar (utilizaremos RGB, para obtener imágenes a color), asignamos la propiedad de frecuencia de captura de fotogramas (un intervalo de 5 fotogramas como entrada de video).

Después de asignar las especificaciones anteriormente mencionadas, comenzamos con la captura de video (para nuestro caso los fotogramas adquiridos serán 1000).

En la figura 3 se muestra la sección de código en MATLAB correspondiente a la configuración de captura del video en tiempo real.

```
%Comenzamos configurando la entrada de video
a = imaqhwinfo; % Formato de camara
vid =videoinput('winvideo',1,'YUY2_640x480');% Captura los cuadros del video
% Propiedades del objeto y la forma en que vamos a obtener las imagenes
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 5; %%Frecuencia de entrada de vídeo
start(vid) % Inicia video

while(vid.FramesAcquired<=1000)% Inicializa para que se detenga adquisición
```

Figura 3 Configuración de captura de video en tiempo real

Detección de objetos a color

Una vez ya comenzada la captura de video en tiempo real, procedemos a substraer el color que deseamos detectar (Rojo, Verde o Azul, por sus siglas en ingles RGB) en el fotograma capturado no sin antes haber sido convertido a escala de grises. Eliminamos ruido utilizando el filtro medio (medfilt2 método de MATLAB), el resultado obtenido del filtro pasamos a convertirlo a binario (utilizando un umbral de 0.18 con el método im2bw en MATLAB), descartamos objetos en la imagen que llevamos procesada hasta el momento menores a 300 pixeles (utilizando el método bwareopen), seguimos con el etiquetado de cada uno de los objetos encontrados de acuerdo al color a detectar establecido al inicio del procesamiento del fotograma (utilizando el método bwlabel).

Regionprops nos permitirá obtener un conjunto de propiedades para cada uno de los objetos detectados en el procesamiento del fotograma.

La siguiente sección de código correspondiente a la figura 4 muestra el procesamiento del fotograma para la detección de objetos de color rojo.

```
data = getsnapshot(vid); % Obtiene un fotograma del cuadro actual

 %Deteccion de objetos en rojo en tiempo real
red_im = imsubtract(data(:,:,1), rgb2gray(data)); %Resta del color
red_im = medfilt2(red_im, [3 3]); %Filtro medio para filtrar el ruido
red_im = im2bw(red_im,0.18); %Convirtiendo de escala de grises a una imagen binaria
red_im = bwareaopen(red_im,300); %Quita pixels menores de 300 pixels
bw_red = bwlabel(red_im, 8); %Etiquetar imagen
stats_red = regionprops(logical(bw_red), 'BoundingBox', 'Centroid'); % Propiedades para cada región
```

Figura 4 Detección de objetos de color rojo

ISSN 2444-4995

Marcado de objeto detectado y conteo

En este punto ya contamos con una imagen (fotograma) procesada en el cual se encuentran detectados solamente los objetos del color que fue configurado a detectar (RGB). Continuamos con el marcado de cada uno de los objetos detectados anteriormente (proporcionados por el método de regionprops), donde localizaremos su centroide, las coordenadas del objeto (base a su centroide) en la imagen y lo encerraremos en un rectángulo del mismo color en el cual fue configurado a detectar.

Procedemos a llevar un contador por cada uno de los diferentes colores a detectar (en este caso 3 contadores diferentes por el RGB), para posteriormente cada vez que es detectado cada uno de los objetos mandar una señal digital a Arduino en el cual configuramos el pin 5, 6, y 7 como salidas digitales, permitiéndonos observar mediante los led de color RGB cada vez que un objeto es detectado respectivamente.

La figura 5 muestra la sección de código en MATLAB para realizar el marcado de los objetos de color rojo detectados en el fotograma procesado.

```
% Ciclo para encerrar objetos rojos
for object_red = 1:length(stats_red)
    bb_r = stats_red(object_red).BoundingBox;
    bc_r = stats_red(object_red).Centroid;
    rectangle('Position',bb_r,'EdgeColor','r','LineWidth',5) %%Selección del color y grosor de lineas del rectángulo
    plot(bc_r(1),bc_r(2), '-m+') %%Grafico de los referenciado previamente

    count_r=count_r+1;
    a=text(bc_r(1)+15,bc_r(2), strcat('X: ', num2str(round(bc_r(1))), ' Y: ', num2str(round(bc_r(2)))));
    set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'yellow');
    title({'Objetos Rojos: ',num2str(count_r), ', ', 'Objetos Verdes: ',num2str(count_g), ', ',
        'Objetos Azules: ',num2str(count_b) , ', Objetos Totales R-V-A: ' ,num2str(count_r + count_g + count_b)});

    %%Arduino uno rojo
    b.pinMode(07,'output'); %Configura el pin 07 como salida
    b.digitalWrite(07,1); %El pin 07 se pone en alto
    pause(0.1); % 0.1 segundo de espera
    b.digitalWrite(07,0); %El pin 07 se pone en bajo
    pause(0.1); % 0.1 segundo de espera
end
```

Figura 5 Marcado de objetos rojos detectados

Resultados

La Figura 6 muestra los objetos que se detectaron con las características anteriormente mencionadas (Seccion 3 Procesamiento de imágenes).

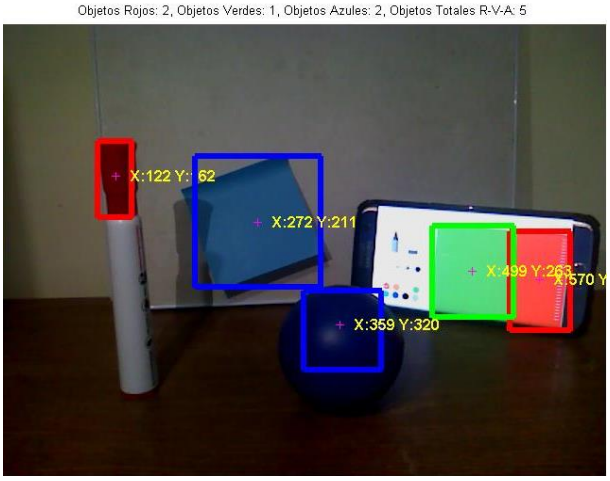

Figura 6 Detección de 5 objetos

En la figura 7 se muestra diferentes cantidades de objetos detectadas con diferente iluminación.

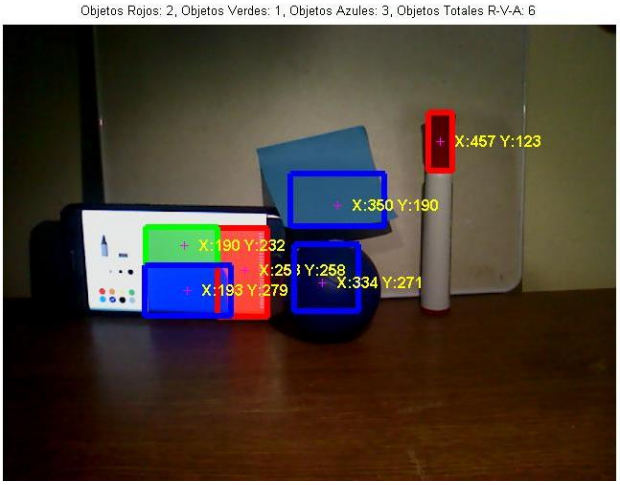

Figura 7 Detección de 6 objetos

En la siguiente tabla 1 se encuentra una comparación entre la cantidad de objetos detectados en las diferentes figuras utilizando un umbral del 18% (0.18 al momento de convertir de escala de grises a binario).

Imagen	Objetos Rojos	Objetos Verdes	Objetos Azules	Total de objetos
Fig 2	2	1	2	5
Fig 3	2	1	3	6

Tabla 1 Objetos detectados por figura

ISSN 2444-4995

La figura 8 con un umbral al 10% y la figura 9 con un umbral al 20% fueron las que presentaron un mejor resultado al momento de detectar objetos de color RGB. La tabla 2 que se muestra a continuación contiene resultados de la misma figura sometidas a otros porcentajes de umbral.

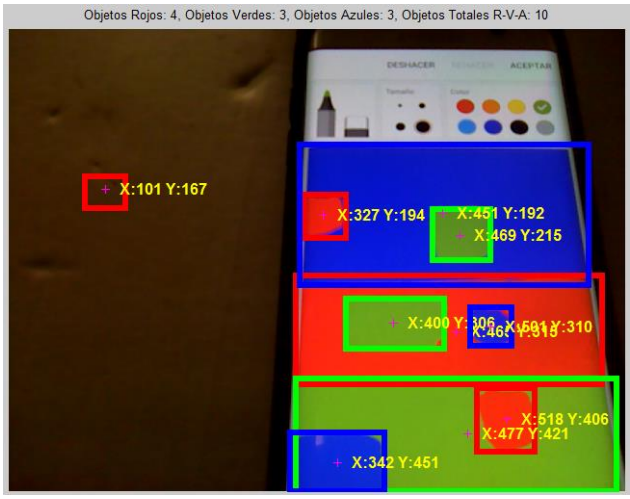

Figura 8 Umbral al 10% detección de 10 objetos en total

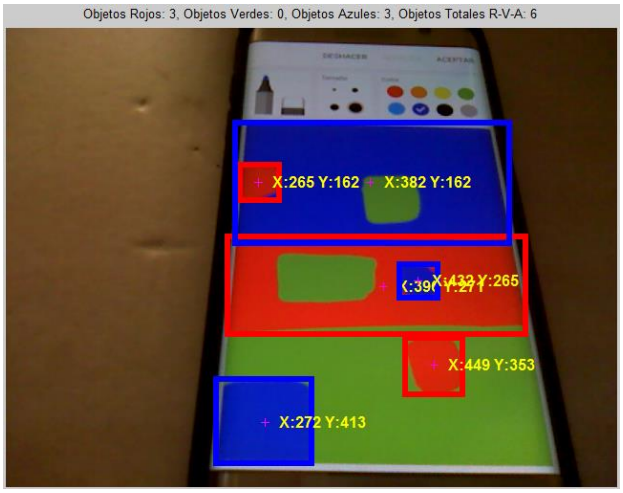

Figura 9 Umbral 20% detección de 6 objetos en total

Imagen	Objetos Rojos	Objetos Verdes	Objetos Azules	Total de objetos
Fig 8 umbral 10%	4	3	3	10
Fig 9 umbral 20%	3	0	3	6
Umbral 18%	3	0	3	6
Umbral 50%	0	1	1	2
Umbral 80%	0	0	0	0

Tabla 2 Objetos detectados con diferentes umbrales

Conclusiones

La sincronización de las dos plataformas (Arduino Uno R3 y MATLAB 2011) facilita sin gran cantidad de esfuerzo el procesamiento de imágenes para aplicar visión artificial y al mismo tiempo de una manera más sencilla permite crear aplicaciones de automatización; existe ventaja al momento de utilizar estas dos plataformas ya que Arduino siendo código abierto tiene una gran comunidad que respalda y ayuda el aceleramiento de crear diferentes aplicaciones, MATLAB también cuenta con una gran comunidad que lo respalda ya que lleva mucho tiempo evolucionando; realizar este proyecto con la característica presentadas como lo son: toma de video en tiempo real y detección de objetos de color en ese mismo instante. Se nos permite en relativamente pocos pasos obtener un producto final satisfactorio, presentando buenos resultados.

Por otra parte en este tipo de proyecto también se tienen que tener en cuenta las siguientes consideraciones: en ocasiones un objeto de color Rojo, Verde o Azul, para nosotros como seres humanos es fácil identificarlos en diferentes situaciones, pero para un sistema con procesamiento de imagenes, existen diferentes tipos de variables que no siempre pueden estar a nuestro favor, como lo es el reflejo de los objetos, las sombras, la luz del ambiente principalmente, provocando que en muchas situaciones se tenga que intentar ajustar lo mejor posible estas variables, como lo es en este caso el valor del umbral que se trabaja al convertir de escala de grises a binario.

Otro punto a tomar en cuenta son las pausas al momento de mandar la señal a Arduino, las cuales son las que permiten que el led tenga una duración de prendido y de apagado, si estas pausas se aumentan, la adquisición de imágenes en el video se volverá más lenta.

El umbral en la conversión de escala de grises a imagen binaria es una parte importante en la detección de objetos y como se muestra en la tabla 2 los mejores resultados se muestran en un rango mayor al 10% y menor al 50%.

ISSN 2444-4995

Referencias

Dean Seal, "Install the MATLAB and Simulink Support Packages for Arduino", organization MATLAB, disponible: https://www.mathworks.com/videos/install-the-matlab-and-simulink-support-packages-for-arduino-106497.html , martes 27 de Diciembre del 2016.

Arduino, "Blink", organización Arduino, disponible: https://www.arduino.cc/en/Tutorial/BlinkWithoutDelay , martes 27 de Diciembre del 2016.

Apuntes de Visión Artificial. Procesamiento morfológico. Dpto. Electrónica, Automática e Informática Industrial. www.elai.upm.es.

Funciones para cambio de espacio de color. Funciones MATLAB. profesores.fi-b.unam.mx. http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm

ISSN 2444-4995