



Benemérita Universidad
Autónoma de Puebla



FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

PRACTICA 5: Aplicación de visión artificial con Matlab

Nombre de la Asignatura:

CONTROL DE PROCESOS POR COMPUTADORA

Profesor:

DR. JAIME CID MONJARAZ

Alumno:

ISLAS FUENTES JESUS ALBERTO

Objetivos:

Esta practica tiene como objetivo usar la herramienta de Matlab llamada “Visión System Toolbox” que nos permite diseñar y simular aplicaciones de visión artificial y procesamiento de imágenes, se puede realizar aplicaciones como detección de objetos, seguimiento de objetos, visión estéreo y tareas de detección de movimiento, en esta practica se usara la capacidad de detectar movimiento para rastrear la trayectoria de un objeto móvil, la aplicación primero realiza la captura del movimiento en 50 Frames y posteriormente se procesara la información mostrando en una nueva la trayectoria realizada por el objeto móvil.

Desarrollo:

Lo primero que se realizo fue revisar los ejemplos brindados en la ayuda de Matlab sobre procesamiento de imágenes usando Simulink y el lenguaje nativo de Matlab, al analizar los ejemplos se determino que para realizar la aplicación descrita en los objetivos seria más versátil realizarla usando el lenguaje nativo de Matlab.

Por lo cual se procedió a realizar una investigación de las funciones requeridas para implementar esta aplicación en base a ejemplos que se tomaron como referencia.

Finalmente se después de varias pruebas el código completo se describe detalladamente a continuación.

En el siguiente bloque de código se realiza la inicialización de la cámara web y se define el formato de entrada Ycbcr422 a 1280x720px que es el único formato nativo disponible para esta webcam, posteriormente se indica que se tienen que capturar 50 Frames, después se realiza la conversión a RGB para su posterior procesamiento, inmediatamente se inicia la captura de video y se cierra la vista previa, los datos de imagen quedan almacenados en la variable “videocaptura”

```
1 - video = videoinput('macvideo', 1, 'YCbCr422_1280x720');
2 - set(video, 'FramesPerTrigger', 50);
3 - set(video, 'ReturnedColorSpace', 'RGB');
4 - video.ReturnedColorSpace
5 - start(video)
6 - preview(video)
7 - wait(video)
8 - videocaptura = getdata(video);
9 - closepreview
```

A continuación se realiza una conversión de las imágenes a escala de grises para facilitar y agilizar su procesamiento, también para optimizar el procesamiento considerando que los movimientos a rastrear no serán tan rápidos se toma un muestreo del video cada 2 cuadros, el resultado se guarda en la variable “datavideo1”

```

%%conversion escala grises
N=2;
[H,W,C,F] = size(videocaptura);
for f = (F/N):-1:1
    datos1(:,:,f) = rgb2gray(videocaptura(:,:,N*f));
end
s = size(datos1);

datavideo1 = datos1;

```

A continuación se realiza un proceso llamado promedio de Frames que nos sirve para eliminar el ruido que puede tener cada imagen debido a condiciones de iluminación.

Posteriormente se realiza una diferenciación usando la función `imabsdiff` que devuelve la diferencia absoluta entre cada Frame y de esta forma detectar las partes móviles.

```

datavideo12 = datavideo1;
for g = s(3)-1:-1:2
    datavideo12(:,:,g) = (datavideo1(:,:,g) + 0.5*datavideo1(:,:,g+1))/1
end

videodatos2 = zeros(s);
for a = s(3)-1:-1:1
    diffvid(:,:,a) = imabsdiff(datavideo1(:,:,a),datavideo1(:,:,a+1));
    gt(a) = graythresh(diffvid(:,:,a));
    if gt(a) > 0.01
        videodatos2(:,:,a) = (diffvid(:,:,a) >= (gt(a) * 255));
    else
        videodatos2(:,:,a) = 0;
    end
end

```

Después se eliminan todos los objetos pequeños con la función `bwareaopen`. Esta función elimina todos los objetos en una imagen binaria con tamaño menor al especificado, como se puede ver se eliminan los objetos pequeños con tamaño menor a 20 pixeles de área, de esta forma eliminamos perturbaciones de objetos que no son de nuestro interés.

```

for g = s(3):-1:1
    videodatos22(:,:,g) = bwareaopen(videodatos2(:,:,g), 20,8);
end

```

En este punto del procesamiento se tienen ya una imagen en color negro con solo 2 objetos blancos que representa el objeto móvil en la posición actual y su posición en el frame anterior, ahora se procede a realizar el cálculo de sus centroides de ambos objetos con la función `regionprops`.

Los datos obtenidos se guardan en una matriz de mayor a menor, considerando que las coordenadas del centroide del primero objeto represente el objeto con mayor área.



```

centroides = zeros(s(3),2);
for n = 1:1:s(3)
    if sum(sum(videodatos22(:,:,n))) > 0
        reg = regionprops(logical(videodatos22(:,:,n)), 'area', 'centroid');
        av = [reg.Area];
        [value, index] = max(av);
        centroides(n,:) = reg(index(1)).Centroid;
    end
end

```

Una vez teniendo nuestra matriz de centroides se procede a realizar una técnica de optimización para descartar posibles ruidos debido a la iluminación, con esto se descartarán todos los conjuntos de centroides que no se muestren en por lo menos 5 frames consecutivos.

```

for p = 3:1:s(3)-2
    if centroides(p-2,1) && centroides(p+2,1) == 0
        centroides(p,1) = 0;
    end
    if centroides(p-2,2) && centroides(p+2,2) == 0
        centroides(p,2) = 0;
    end
end

```

Ya que tenemos detectados los puntos del movimiento del objeto que están en la matriz centroides, procedemos a mostrar los resultados, primero se crea una nueva figura en donde desplegara el ultimo frame capturado.

```

clear figure(1)
figure(1)
imshow(videocaptura(:,:,:,s(3)))
hold on

```

Finalmente se grafican los centroides sobre la figura y se unen los puntos para mostrar la trayectoria.

```

q = centroides(:,1);
q(~q) = NaN;
v = centroides(:,2);
v(~v) = NaN;
plot(q,v,'y')

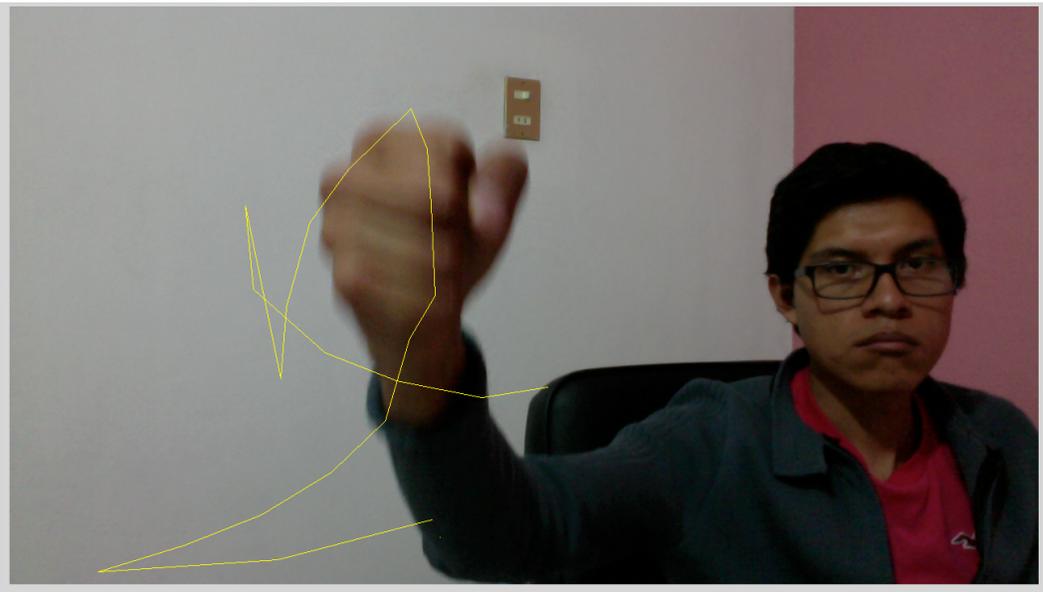
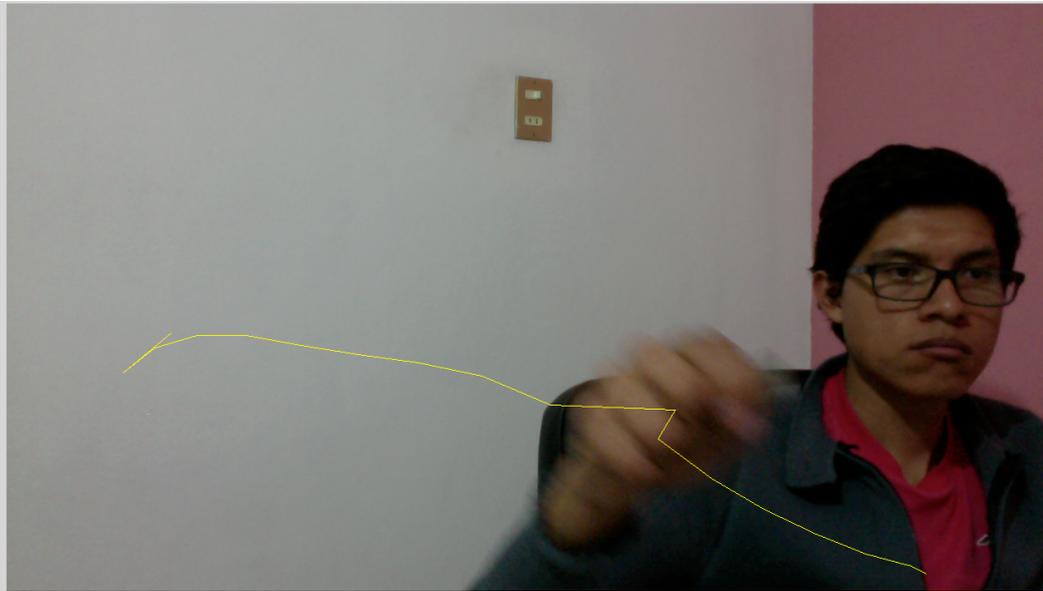
clear all

```

Pruebas:

A continuación se muestran un par de pruebas en donde la trayectoria de movimiento esta representada por la línea color amarillo.





Conclusiones:

Como se puede observar se cumplió con el objetivo propuesto de trazar la trayectoria de un objeto en movimiento, sin embargo en base a los resultados obtenidos se puede notar que el rastreo de la trayectoria aun es muy básica con aplicaciones limitadas por lo cual se puede mejorar aumentando el numero de frames y agregando un filtro extra que nos permita definir el objeto móvil de nuestro interés que deseemos analizar por ejemplo discriminándolo por color, de esta forma mejorar el rastreo de la trayectoria.

