



Detector de ropa Blanca

Profesor:

Elizabeth Guevara

Materia:

Procesamiento Digital de Señales

Alumno:

Hans Alexander Luna Eisermann

Id:

00012332

Introducción

A lo largo del curso hemos estudiado y analizado el comportamiento de una señal analógica en donde necesitamos un convertidor analógico digital para convertir esta señal a digital para posteriormente analizar y poder manipular la señal, en donde esta señal puede ser datos, imágenes, voz, audio o video.

En esta materia hemos analizado estas señales desde su comportamiento cuando son datos, esto es necesario para entender el funcionamiento y los procesos que realizamos posteriormente a las señales en donde se almacenan imágenes, audio o video.

En el curso hemos aprendido a manipular todas estas señales tanto manualmente, como utilizando un software llamado Matlab que en este caso es de mucha ayuda para poder ver gráficamente todos los procesos que estamos aplicando a nuestra señal.

Al igual que en el curso para este proyecto utilizaremos el software Matlab, ya que este software nos ayudara ya que existen ciertas ecuaciones que ya están establecidas en Matlab las cuales nos ayudaran a hacer nuestras operaciones y a la vez nos permitirá ver gráficamente todos los proceso y también nos ayudara para hacer un programa en el cual le podemos dar una entrada y definiremos cual será nuestra respuesta-

En este proyecto nuestra entrada será una foto y nuestra respuesta que a través la foto nos dirá si la persona viene vestido de blanco.

La foto será tomada al detectar un sonido alto, de lo contrario esta foto nunca se tomara.

ENTRADAS	RESPUESTA
SONIDO EXTERNO	TOMAR FOTO
FOTO	SABER SI VIENE VESTIDO DE BLANCO

Objetivo:

Que el desarrollador del programa en este caso el alumno, aplique lo aprendido en el curso en un solo programa que haga una función en específica. Al aplicar lo aprendido en el curso se podrá entender más fácil los funcionamientos de las funciones aprendidas.

La realización de este proyecto es meramente de aprendizaje ya que no se piensa comercializar con esta aplicación y es por eso que la implementación de este proyecto se realiza en un ambiente controlado.

Aplicaciones:

Este proyecto se pensó en el sistema de seguridad a la hora de entrar a un laboratorio, en donde específicamente se tiene que portar vestimenta “blanca”.

Otras aplicaciones:

- ENTRADA A UN LABORATORIO
- ENTRADA A UN HOSPITAL
- ENTRADA A UNA COCINA DE UN RESTAURANTE
- ENTRADA A UNA OFICINA EN DONDE LA VESTIMENTA SEA BLANCO
- EN UN SISTEMA DE SEGURIDAD EN DONDE EN UN CUARTO BLANCO SE TOME UNA FOTO CADA DETERMINADO TIEMPO Y SI EL PROGRAMA DETECTA QUE NO HAY BLANCO, HUBO ALGO QUE CAMBIO ESTE UMBRAL DE COLOR.

Al realizar este proyecto también estamos implementando sonido en el cual al obtener un sonido muy alto tome una foto, en otros casos lo podríamos tener como:

- ALARMAS (AL DETECTAR UN SONIDO FUERTE REPRODUCIR UNA ALARMA)
- BARK OFF (EN DONDE SI UN PERRO LADRA SE ACTIVE UN SONIDO EN DONDE SEA TAN MOLESTO PARA EL PERRO QUE LO DEJE DE HACER)

Desarrollo:

En la figura 1.1 se muestran los pasos a seguir para la implementación de este proyecto.

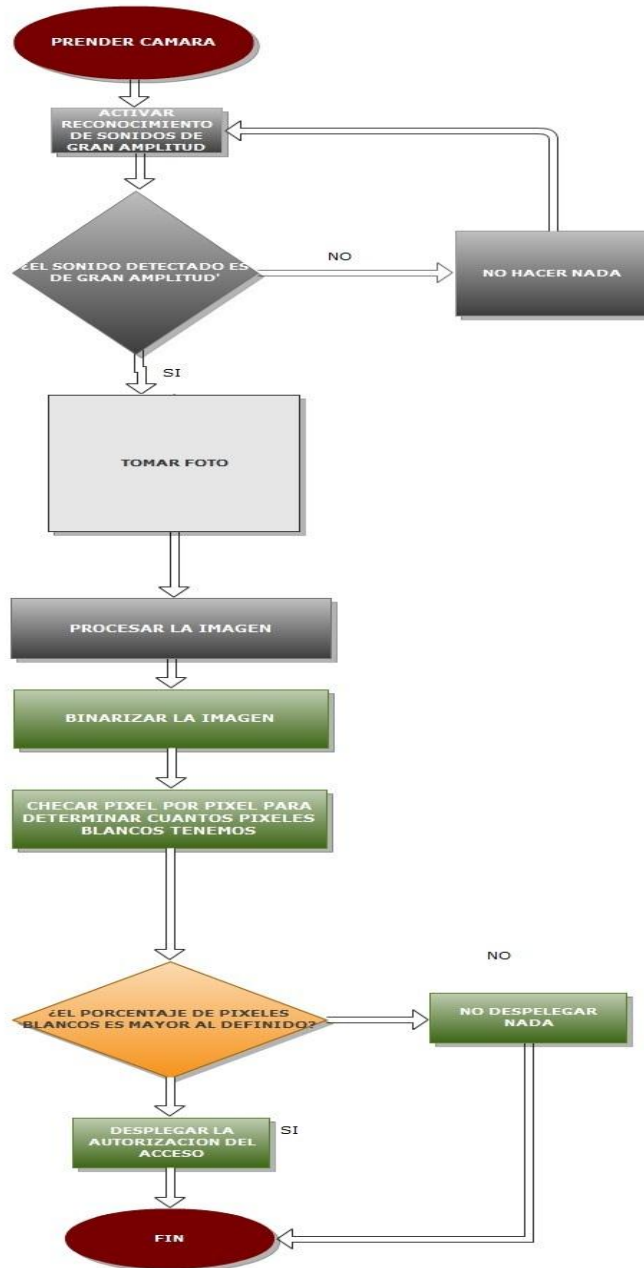


Figura 1.1

Para el desarrollo de este proyecto se utilizó una interfaz GUI, esta interfaz es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objeto gráficos para representar la información y acciones disponibles en la interfaz.

Gracias a la interfaz gráfica podemos añadir unos botones en donde cada botón se encuentra un algoritmo para realizar las operaciones deseadas. En la figura 1.2 se muestra el GUI que fue utilizado para este proyecto.

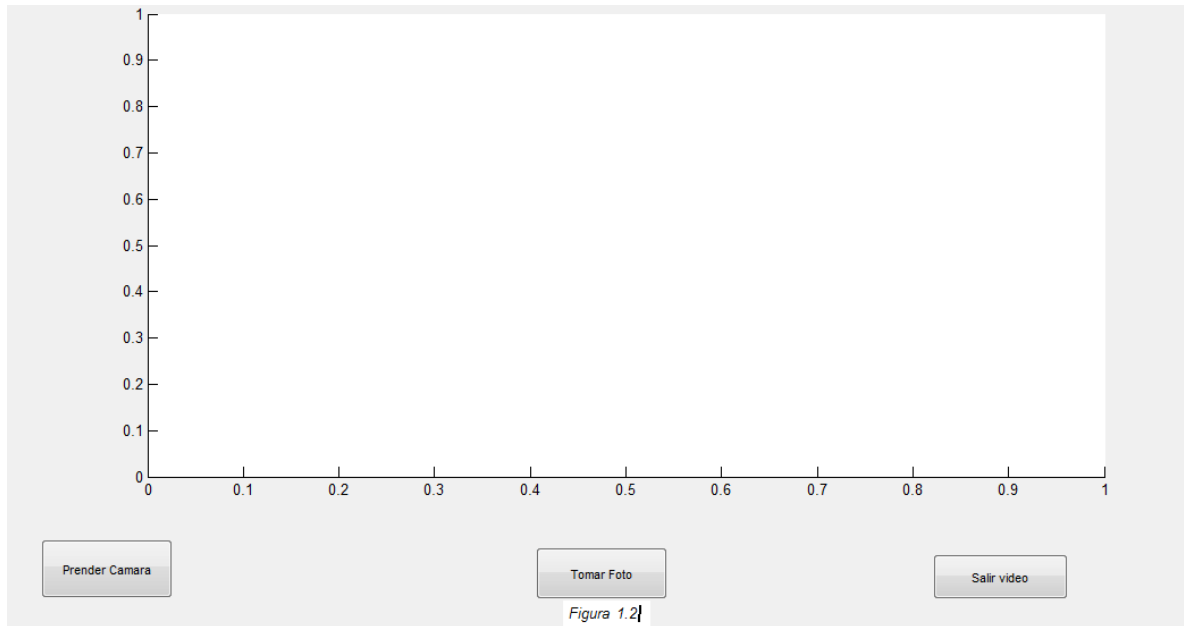


Figura 1.2

En donde en Prender cámara (Botón UNO) se encuentra el siguiente código:

```
global frame;
global canalvideo;
canalvideo=videoinput('winvideo',1);
preview(canalvideo);
```

En donde este código se utiliza para prender la cámara web.

En donde:

videoinput('winvideo',1): Es el comando que utiliza Matlab para prender la cámara en donde 'winvideo' es el dispositivo de Windows y el 1 corresponde el número de cámara a prender y como en este caso solo tenemos una cámara se utiliza el valor 1.

Preview(); Es el comando para visualizar el contenido de la cámara web que estamos obteniendo con el comando anterior.

En donde en Tomar Foto (Botón DOS) se encuentra el siguiente código:

```
Fs = 11024;%DEFINIMOS NUESTRA FRECUENCIA A LA QUE SERA GRABADO NUESTRO SONIDO
x = wavrecord(2*Fs, Fs, 'int16');% NOS PERMITE GRABAR EL SONIDO EN DONDE EL PRIMER PARAMETRO SON LOS NUMEROS DE SEGUNDOS
%%% A GRABAR MULTIPLICADOS POR LA FRECUENCIA, EL SEGUNDO PARAMETRO ES LA FRECUENCIA A LA CUAL SERA GRABADA Y EL TERCER Y ULTIMO PARAMETRO ES EL
```

```

%%% # DE BITS QUE TENDRA EL AUDIO
wavplay(x, Fs); %% REPRODUCIMOS EL SONIDO GRABADO A LA FRECUENCIA
ESTABLECIDA PARA VERIFICAR QUE HUBO UN SONIDO
m2=max(x);%%%NOS ARROJA EL VALOR MAXIMO DE AMPLITUD DE LA SEÑAL QUE SE
OBTUVO DE LA MATRIZ
if m2 >= .50 %%% DEFINIMOS QUE SI EN LA MATRIZ HAY UN VALOR MAYORMAYOR AL
50 PORCIENTO DE LA AMPLITUD VA A SER LO NECESARIO PARA
    %%% TOMARNOS LA FOTO
frame=getsnapshot(canalvideo); %%% ESTA COMANDO NOS VA A CAPTURAR UNA
IMAGEN DEL VIDEO QUE ESTA RECIBIENDO
IM =frame; %%% LA GUARDAMOS EN UNA VARIABLE PARA QUE POSTERIORMENTE LA
PODAMOS MANIPULAR
subplot(221);
imshow(IM); %%% COMANDO QUE NOS MUESTRA LA FOTO OBTENIDA
title('Imagen Original');
[nf nc c]=size(IM); %%% ESTE COMANDO NOS VA A PERMITIR SACAR EL TAMAÑO DE
LA IMAGEN PARA QUE POSTERIORMENTE PODAMOS
    %%% ANALIZAR LA MATRIZ PIXEL POR PIXEL

ROI=[152    203    255 %%% ESTA MATRIZ SON LOS COLORES QUE DEFINIMOS
ANTERIORMENTE
    202    232    215 %%% YA QUE QUEREMOS FILTRAR BLANCO (255) PERO EL
BLANCO PURO NO LO ENCONTRAMOS
    189    222    215 %%% DEFINIMOS COLORES QUE SEAN MUY CERCANOS AL
BLANCO Y QUE EN EL MEDIO AMBIENTE
    189    222    215 %%% NOS PERCATAMOS QUE ES BLANCO
    189    222    215
    196    227    215
    198    230    215
    202    234    215
    202    228    213
    202    228    213
];
ROI=double(ROI); % me convierte de una matriz a un numero
MC=cov(ROI); %%Para obtener la matriz de covarianza en Matlab se aplica
el comando MC=cov(ROI)
%%% esta matriz es necesaria para poder aplicarl el concepto de la
%%% distancia de MAHALANOBIS
MCi=inv(MC); %La distancia de Mahalanobis utiliza la inversa de la
matriz de covarianza, por lo tanto se debe
%%%aplicar el comando de Matlab "inv" para invertir la matriz:
MCi=inv(MC);
MCi=MCi/max(max(MCi)); %%Generalmente se utiliza el valor normalizado de
la matriz inversa de covarianza. Para ello se divide
%%%la matriz MCi por el máximo valor presente: MCi=MCi/max(max(MCi));
colormedio2=[mean(ROI(:,1));
mean(ROI(:,2));
mean(ROI(:,3))];
%%% EL COMANDO MEAN NO ARROJA LOS VALORES MEDIOS DE CADA COLUMNA DE LA
%%% MATRIZ ROI
IM3=IM; %%% GUARDAMOS EN UNA VARIABLE NUEVA NUESTRA IMAGEN ORIGINAL
for f=1:nf %%% UTILIZAMOS UN FOR PARA HACER LA SUMATORIA QUE NOS IRA
GENERANDO UNA MATRIZ NUEVA DE VARIANZA COMPARANDO ELEMENTO A ELEMENTO
    %%% DE NUESTRA MATRIZ DE LA IMAGEN
    for c=1:nc
        z=[IM(f,c,1);IM(f,c,2); IM(f,c,3)];
        z=double(z)/255;
    end
end

```

```

    a=double(colormedio2)/255;
    d2=(z-a)'*MCi*(z-a); %%% SE APLICA LA FORMULA DE LA DISTANCIA DE
    MAHALANOBIS CON NUESTRAS 2 NUEVAS MATRICES DE VARIANZA
    if d2>0.12, %%% se define este porcentaje ya que es el adecuado
    para que los colores que no estan dentro de la matriz ROI los haga negros
        IM3(f,c,:)= [0 0 0];
    end;
end;
end;

subplot(222);
imshow(IM3);
IM31=rgb2gray(IM3); %%% CONVERTIMOS LA IMAGEN OBTENIDA CON MAHALANOBIS A
GRIS
figure(2)
subplot(211);
imshow(IM31)
subplot(212);
imhist(IM31)

u=170 %%% DEFINIMOS UN UMBRAL
IM31=IM31>u %%% DECIMOS QUE SI PASA ESTE UMBRAL TODO VALOR MAYOR LO HARA
BLANCO = 1 Y TODO LO MENOR LO HARA NEGRO =0
[y1 x1]= size(IM31) %%% SACAMOS EL TAMAÑO DE LA IMAGEN BINARIZADA
pixW=0; %%% DEFINIMOS UNA VARIABLE QUE SE INICIALIZA EN CERO
for y=1:y1 %%% HACEMOS DOS FOR PARA RECORRER TODOS LOS PÍXELES DE LA
MATRIZ
    for x=1:x1
        if double(IM31(y,x))==1 %%% AL RECORRER LE PONEMOS CONDICION QUE
SI EL PIXEL ES IGUAL A UNO ENTONCES NOS SUME
            %%%UNA UNIDAD A NUESTRA VARIABLE
            pixW=pixW+1;
        end
    end
end
if pixW/(x1*y1)> 0.5 %%% PONEMOS CONDICION QUE SI EL PORCENTAJE DE 1 QUE
APARECEN EN LA MATRIZ ES MAYOR AL PORCENTAJE DEFINIDO
    %%% ENTONCES NOS DEJE PASAR
    beep
    figure(3)
    title('ES BLANCO ADELANTE')
else %%% DE LO CONTRARIO NO NOS DEJARA PASAR
    figure(3)
    title('LO SIENTO NO PUEDE PASAR')
end
else %%%RECORDEMOS QUE ESTE IF ES EL QUE COMPARABA LA AMPLITUD DEL AUDIO
SI NO SE CUMPLE
    %%%NO NOS ACTIVA LA CAMARA
    figure(3)
    title('LO SIENTO NO SE PUDO ACTIVAR LA CAMARA')
end
end

```

En donde en SALIR VIDEO (Botón TRES) se encuentra el siguiente código:

```
global canalvideo;  
delete (canalvideo);
```

En donde el comando delete(): nos permite cerrar la visualización del video.

RESULTADOS:

Se obtuvo un programa en donde por medio de audio se activa el algoritmo y este algoritmo nos sirve para identificar si alguien viene vestido de blanco.

CONCLUSIÓN:

Debido a que el proyecto está diseñado para hacerse en un ambiente controlado, este proyecto funciona pero debe ser un ambiente controlado, por lo cual concluimos que este programa será implementado a la hora de entrar a un cuarto en donde este cuarto sea todo negro para que al momento de tomar la foto detecte que sea blanco y dependiendo de la iluminación y del cuarto en donde se pruebe tendremos dos variables a controlar, que serían el umbral de la binarización y el porcentaje de blancos que se encuentra en una imagen después de la binarización.