Apellidos, Nombre: Muñoz Navarro, Alejandro
Apellidos, Nombre: Solodilova, María

Introducción

Adjuntad código y e imagen resultante.

```
% Reseteamos
clear;
clc;

%Cargamos imagenes contenidas en ./retratos
lee_jpgs;

%Guardamos el número de imagenes
NUM = L;

%Guardamos el tamaño de las imagenes
ALTO_IMG = N;
ANCHO_IMG = M;

%creamos el mosaico
ALTO_MOSAICO = 2880;
ANCHO_MOSAICO = 4800;
mosaico=uint8(zeros(ALTO_MOSAICO,ANCHO_MOSAICO));

%rellenamos con las imagenes
n=1;
ry=(1:ALTO_IMG);
for k=1:(ALTO_MOSAICO/ALTO_IMG)
    rx=(1:ANCHO_IMG);
    for j=1:(ANCHO_MOSAICO/ANCHO_IMG)
        mosaico(ry,rx)=imags(:,:,n);
        rx=rx+ANCHO_IMG;
        n=n+1;
    end
    ry=ry+ALTO_IMG;
end

% mostramos el resultado
imshow(mosaico);
```

Adjuntad código y e imagen resultante.

```
% Reseteamos
clear;
clc;

%Cargamos imagenes contenidas en ./retratos
lee_jpgs;

%Guardamos el número de imagenes
NUM = L;

%Guardamos el tamaño de las imagenes
ALTO_IMG = N;
ANCHO_IMG = M;

%creamos el mosaico
ALTO_MOSAICO = 2880;
ANCHO_MOSAICO = 4800;
mosaico=uint8(zeros(ALTO_MOSAICO,ANCHO_MOSAICO));

%rellenamos con las imagenes
n=1;
ry=(1:ALTO_IMG);
for k=1:(ALTO_MOSAICO/ALTO_IMG)
    rx=(1:ANCHO_IMG);
    for j=1:(ANCHO_MOSAICO/ANCHO_IMG)
        mosaico(ry,rx)=imags(:,:,n);
        rx=rx+ANCHO_IMG;
        n=n+1;
    end
    ry=ry+ALTO_IMG;
end
```

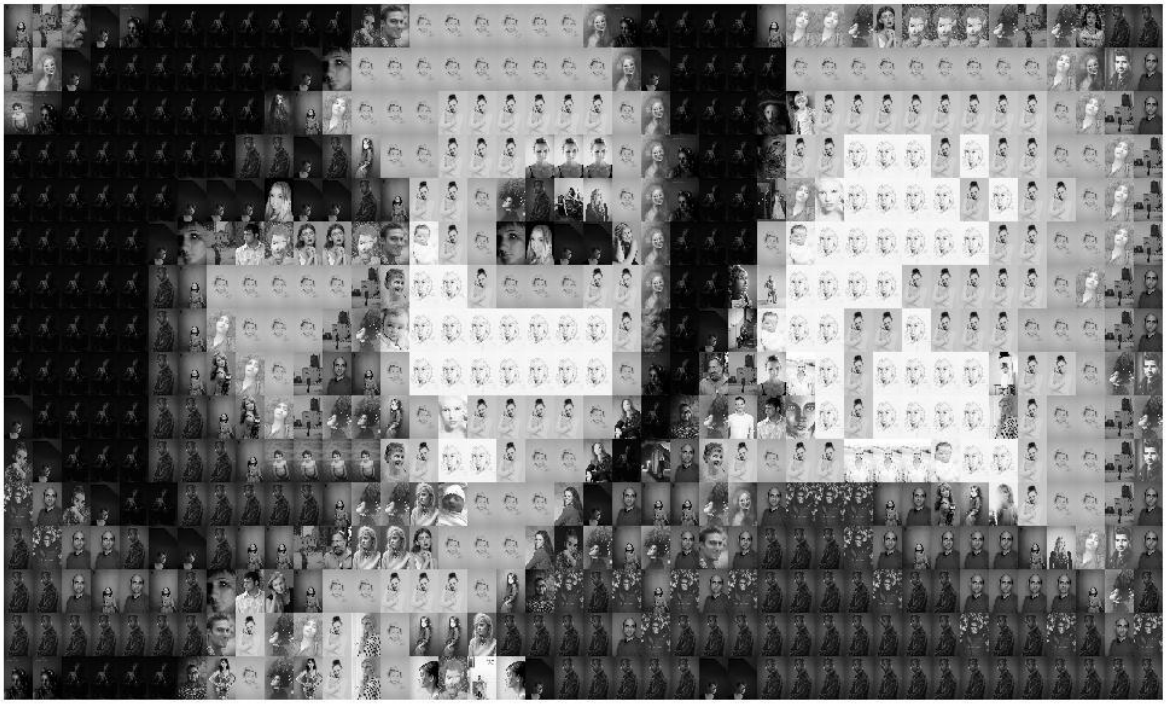
```
% mostramos el resultado
%imshow(mosaico);

%borramos el contenido del mosaico
mosaico(:,:)=0;

%Insertamos imagenes aleatorias

for n=1:L
    yo=floor(rand(1)*(ALTO_MOSAICO-ALTO_IMG));
    xo=floor(rand(1)*(ANCHO_MOSAICO-ANCHO_IMG));
    ry=yo+(1:ALTO_IMG);
    rx=xo+(1:ANCHO_IMG);
    mosaico(ry,rx)=imags(:,:,n);
end

% mostramos el resultado
imshow(mosaico);
```

Creación de mosaicos

Adjuntad script con código. Usando imshow, adjuntad la imagen mosaico resultante.

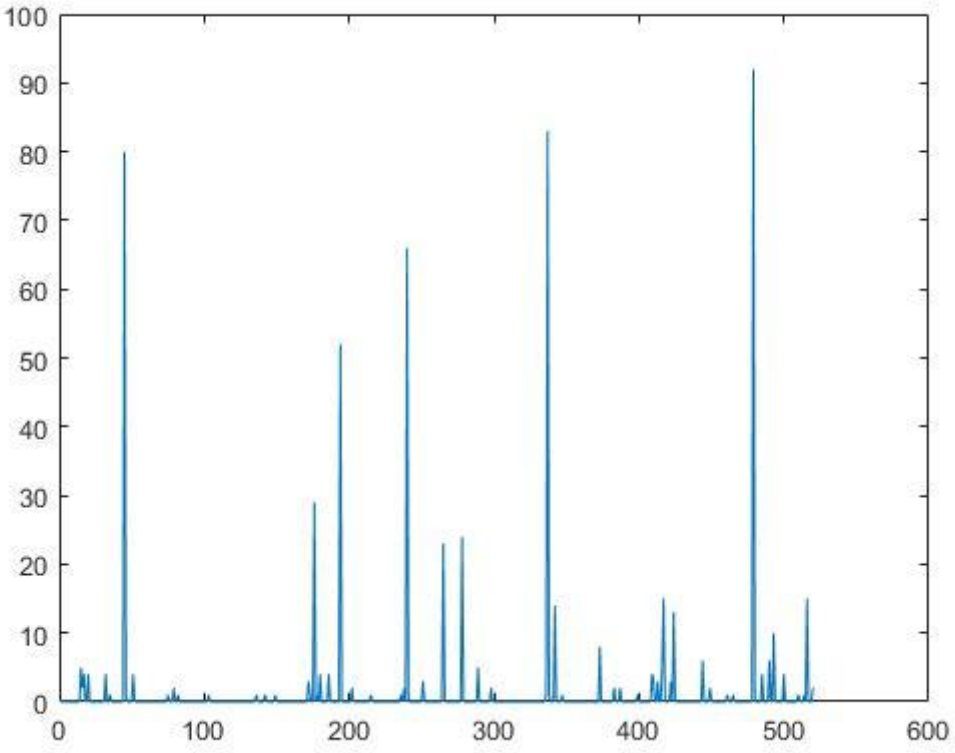
```
% Reseteamos
clear;
clc;

%Cargamos imagenes contenidas en ./retratos
lee_jpgs;

%Guardamos el número de imagenes
```

```
NUM = L;

%Guardamos el tamaño de las imagenes
ALTO_IMG = N;
ANCHO_IMG = M;

%Cargamos imagen target
target = imread('target2.jpg');

%creamos el mosaico
ALTO_MOSAICO = size(target,1);
ANCHO_MOSAICO = size(target,2);
mosaico=uint8(zeros(ALTO_MOSAICO,ANCHO_MOSAICO));

%Reducimos imagenes
F=4;
reds=imresize(imags,1/F);

%rellenamos con las imagenes
ry=(1:ALTO_IMG);
for k=1:(ALTO_MOSAICO/ALTO_IMG)
    rx=(1:ANCHO_IMG);
    for j=1:(ANCHO_MOSAICO/ANCHO_IMG)

        %Extraemos sub-imagen objetivo
        sub_img = imresize(target(ry,rx),1/F);

        %Comparamos con las otras imágenes
        MIN = -1;
        INDICE = -1;

        for i=1:NUM

            %diferencia reducidas
            dif = double(reds(:,:,i))- double(sub_img(:,:));
            %abs
            absoluto = abs(dif);
            %mean2
            media = mean2(absoluto);
            if INDICE < 0 || MIN > media
                INDICE = i;
                MIN = media;
            end
        end

        %introducimos la imagen
        mosaico(ry,rx)=imags(:,:,INDICE);
        rx=rx+ANCHO_IMG;

    end
    ry=ry+ALTO_IMG;
end

% mostramos el resultado
imshow(mosaico);
```



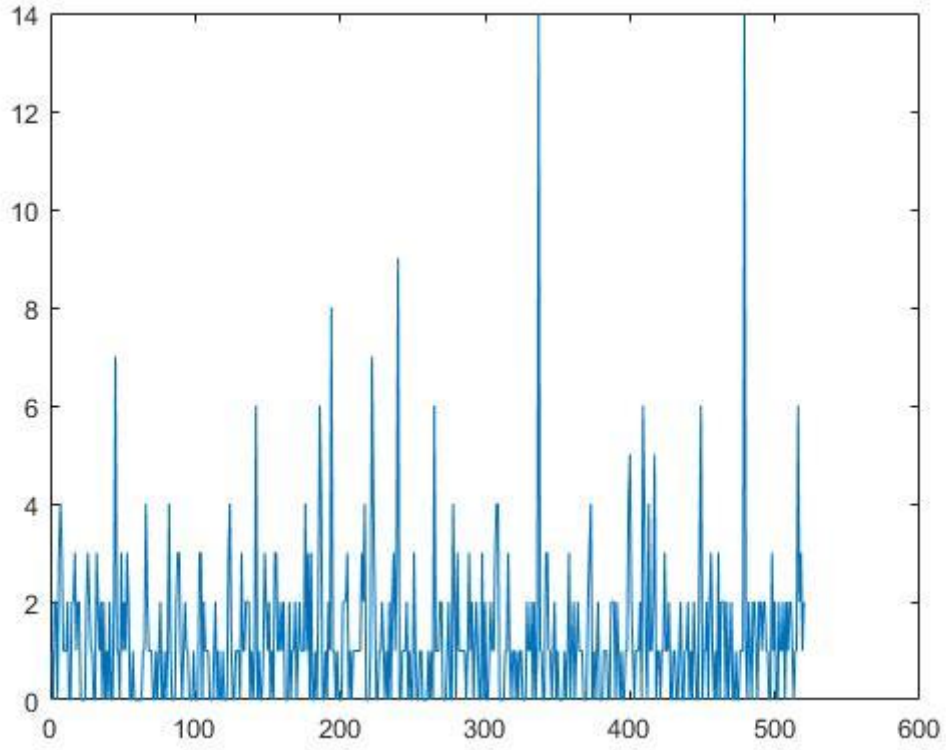
---------- Hasta aquí 25% de la nota ----------

Tras terminar, haced un plot del vector veces y adjuntadlo.



Adjuntad las 2 imágenes más usadas. ¿Cuántas veces se usan cada una de ellas?
Se usan 92 y 83 veces cada una.

Adjuntad nuevo mosaico y la gráfica de veces.



¿Qué imagen (#?) es ahora la más usada? ¿Cuántas veces?

Ambas son usadas 14 veces.

Adjuntad código de vuestro script.

```
% Reseteamos
clear;
clc;

%Cargamos imagenes contenidas en ./retratos
lee_jpgs;
```

```
%Guardamos el número de imagenes
NUM = L;

%Guardamos el tamaño de las imagenes
ALTO_IMG = N;
ANCHO_IMG = M;

%Cargamos imagen target
target = imread('target2.jpg');

%creamos el mosaico
ALTO_MOSAICO = size(target,1);
ANCHO_MOSAICO = size(target,2);
mosaico=uint8(zeros(ALTO_MOSAICO,ANCHO_MOSAICO));

%Reducimos imagenes
F=4;
reds=imresize(imags,1/F);

%rellenamos con las imagenes
veces = zeros(1,NUM);
ry=(1:ALTO_IMG);
for k=1:(ALTO_MOSAICO/ALTO_IMG)
    rx=(1:ANCHO_IMG);
    for j=1:(ANCHO_MOSAICO/ANCHO_IMG)

        %Extraemos sub-imagen objetivo
        sub_img = imresize(target(ry,rx),1/F);

        %Comparamos con las otras imágenes
        MIN = -1;
        INDICE = -1;

        for i=1:NUM

            %diferencia reducidas
            dif = double(reds(:,:,i))- double(sub_img(:,:));

            %abs
            absoluto = abs(dif);

            %mean2
            media = mean2(absoluto);

            %Calculamos factor
            factor = 1 + ((F*veces(i))/(max(veces)+1));
            media = media*factor;

            if INDICE < 0 || MIN > media
                INDICE = i;
                MIN = media;
            end
        end

        %introducimos la imagen
        mosaico(ry,rx)=imags(:,:,INDICE);
        veces(INDICE) = veces(INDICE)+1;
        rx=rx+ANCHO_IMG;

    end
```

```
    ry=ry+ALTO_IMG;
end


% mostramos el resultado
figure();
plot(veces);
figure();
imshow(mosaico);

%buscamos las X más usadas
X = 2;
for n=1:X
    [maximo,ind] = max(veces);
    figure();
    imshow(imags(:,:,ind));
    veces(ind) = 0;
end
```

---------- Hasta aquí 40% de la nota ----------

Adjuntad imagen resultado. ¿Cuántas tiradas de 200 habéis hecho antes de que tanto por ciento de substituciones aceptados baje del 5%? ¿Qué tanto por ciento de aceptación tuviste en la última tirada de 200 pruebas?

Se han realizado 6 tiradas y 8 aciertos (4%).

Adjuntad el mosaico final resultante (para 144x96) ¿Cuántas tiradas de 200 hay que hacer ahora para que el tanto por ciento de substituciones aceptados baje del 5%?

Se han realizado 13 tiradas y 7 aciertos (3.5%).

Adjuntad código de vuestro script para la colocación aleatoria.

```
% Reiniciamos veces
veces = zeros(1,NUM);

contador = 0;
ITERACIONES = 200;
aciertos = ITERACIONES*0.05;
FI = 2;
reds=imresize(imags,1/FI);
%bucle de iteraciones
while aciertos >= ITERACIONES*0.05
    contador=contador+1;
    aciertos = 0;
    for x = 1:ITERACIONES
        %escoger una posición aleatoria
        yo=floor(rand(1)*(ALTO_MOSAICO-(ALTO_IMG/FI)));
        xo=floor(rand(1)*(ANCHO_MOSAICO-(ANCHO_IMG/FI)));
        ry=yo+(1:ALTO_IMG/FI);
        rx=xo+(1:ANCHO_IMG/FI);

        %selecionamos imagen target
        sub_img = target(ry,rx);

        %selecionamos imagen mosaico
        %sub_mos = imresize(cpy_mosaico(ry,rx),1/FI);
        sub_mos = mosaico(ry,rx);

        %diferencia reducidas
        dif = double(sub_mos(:,:)) - double(sub_img(:,:));

        %abs
```

```
        absoluto = abs(dif);

        %mean2
        E_min = mean2(absoluto);

        INDICE = 0;
        %recorremos imagenes
        for i=1:NUM
            %diferencia reducidas
            dif = double(reds(:,:,i)) - double(sub_img(:,:));

            %abs
            absoluto = abs(dif);

            %mean2
            media = mean2(absoluto);

            %Calculamos factor
            factor = 1 + ((4*veces(i))/(max(veces)+1));
            media = media*factor;

            if E_min > media
                INDICE = i;
                E_min = media;
            end
        end
        if INDICE ~= 0
            %introducimos la imagen
            mosaico(ry,rx)=reds(:,:,INDICE);
            veces(INDICE) = veces(INDICE)+1;
            aciertos = aciertos+1;
        end
    end
end

% mostramos el resultado
imshow(mosaico);
```

Adjuntad mosaico final después de ejecutar el script para (72x48) y (36x24).

Para un F=4 (72x48) se ha obtenido:

Para un F=8 (36x24) se ha obtenido:

¿Cuántas iteraciones habéis necesitado en cada paso?

Se han necesitado 51 iteraciones y 8 aciertos (4%) para un F=4, y 153 iteraciones y 9 aciertos (4.5%) para un F=8.

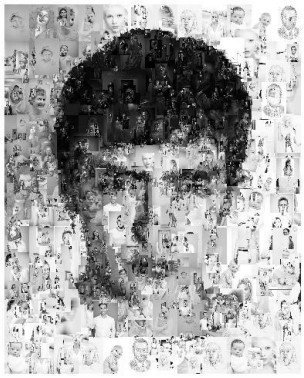
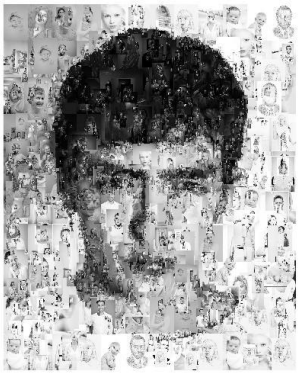
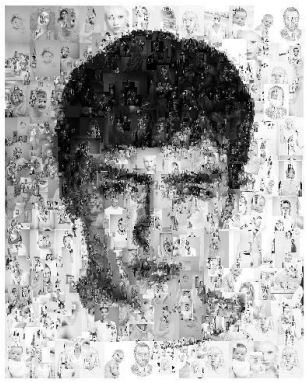
De la última imagen seleccionar la zona del ojo y la ceja de la derecha de la foto y mostrad un detalle de ella.

Haceros una foto vuestra en primer plano, pasarla a B/W usando rgb2gray() y hacer un mosaico a partir de ella usando vuestro programa. Adjuntad la imagen original y el mosaico resultante.

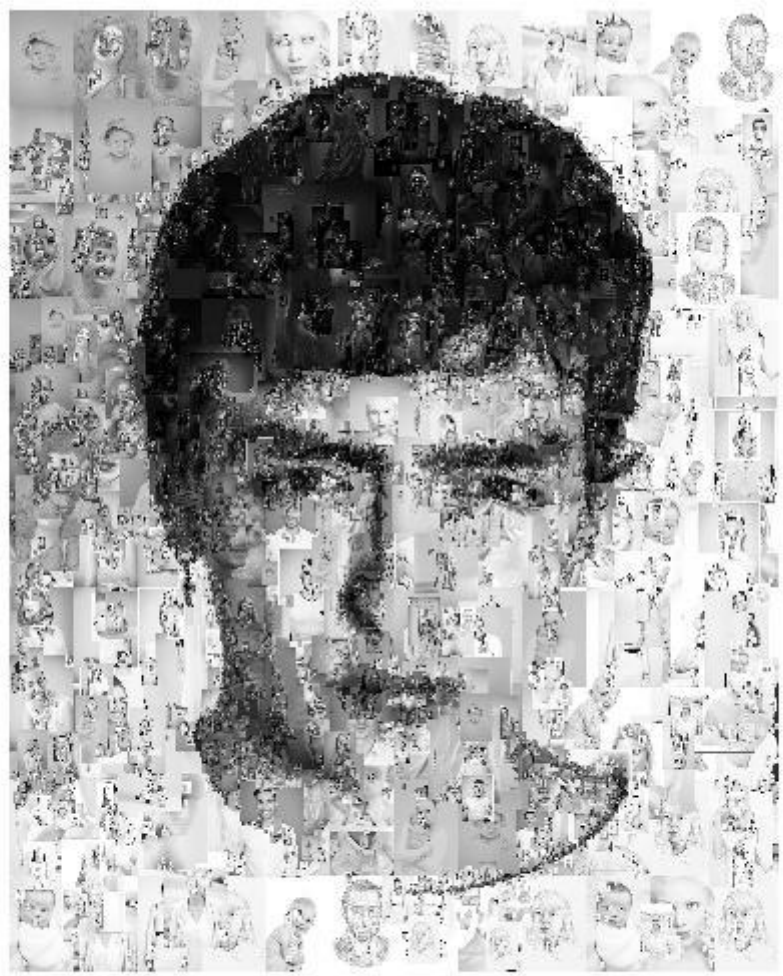
Se parte de la siguiente imagen tirada por una réflex APS-C por uno de los participantes. Se recorta la imagen para que los N retratos puedan completar el mosaico debido al tamaño predeterminado de estos. Finalmente la imagen original queda tal que así:

Antonio Tabernero, 2019

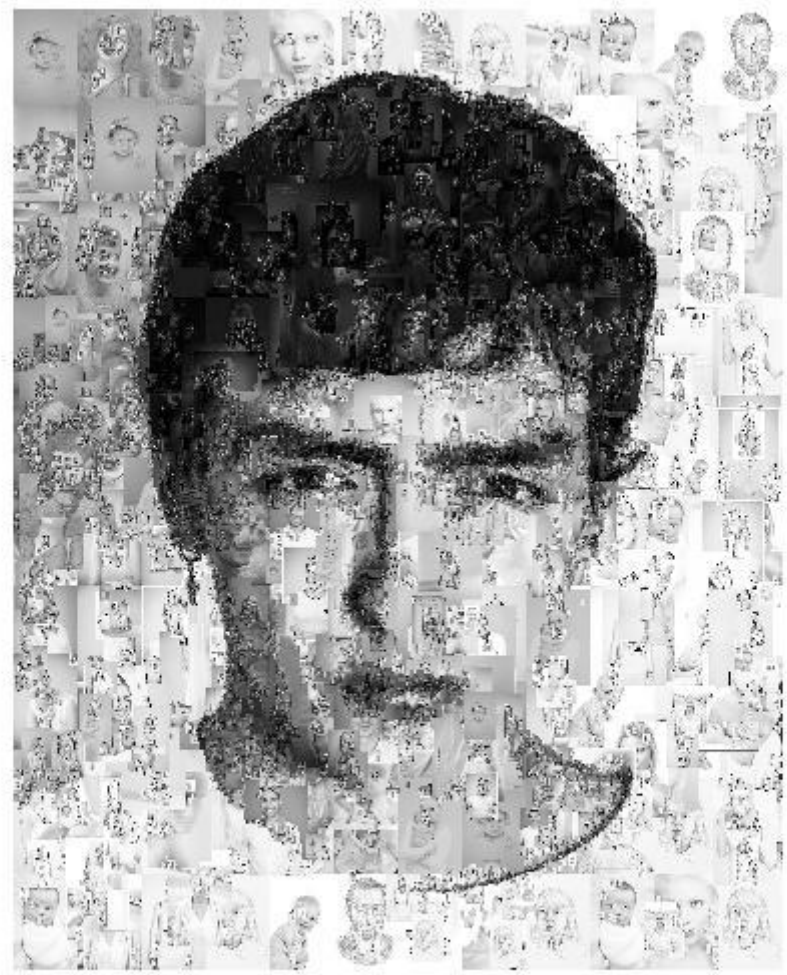
Realizamos todos los pasos como en el ejercicio anterior y repetimos F=8 dos veces más:

Cambiamos F=16, ITERACIONES = 500 y un acierto del 3% de tal forma que podamos obtener el siguiente resultado:



Para obtener nuestro resultado final, cambiamos acierto a 1% para perfeccionar nuestra imagen aún más.



---------- Hasta aquí 75% de la nota ----------

MANEJO de VIDEOS en MATLAB

Info	punto negro	Punto rojo	Punto verde	Punto cyan
X	490	754	862	591
Y	320	220	485	591
R	13	133	32	7
G	10	30	86	90
B	12	16	31	104

Adjuntad código del bucle en j con la actualización de las posiciones X,Y

```
% Calculamos coordenadas
x=round(X(j));
y=round(Y(j));
rx=x+(-RAD:RAD);
ry=y+(-RAD:RAD);
% Extraemos subimagen
sub_img=frame(ry,rx,:);
```

```
% Sacamos los componentes
R=double(sub_img(:,:,1));
G=double(sub_img(:,:,2));
B=double(sub_img(:,:,3));
% Calculamos la diferencia en canales
T=10;
dr=(R-col(1,j))./T;
dg=(G-col(2,j))./T;
db=(B-col(3,j))./T;
% Calculamos w
w=exp(-(dr.^2 + dg.^2 + db.^2));
% Normalizamos w
w=w./sum(w(:));
% Calculamos estimacion y actualizamos
Ax=(x+dx).*w;
X(j) = sum(Ax(:));
Ay=(y+dy).*w;
Y(j) = sum(Ay(:));
```

¿En qué rango de frames es más aparente este comportamiento?

Se pude dar en el que el cuadrado se transforma en un rombo debido a la traslación de la cámara o por temas de iluminación del color seleccionado.

Indicad si éste es o no vuestro caso. Si es así adjuntad una foto de un frame donde se haya perdido el tracking de alguno de los puntos.

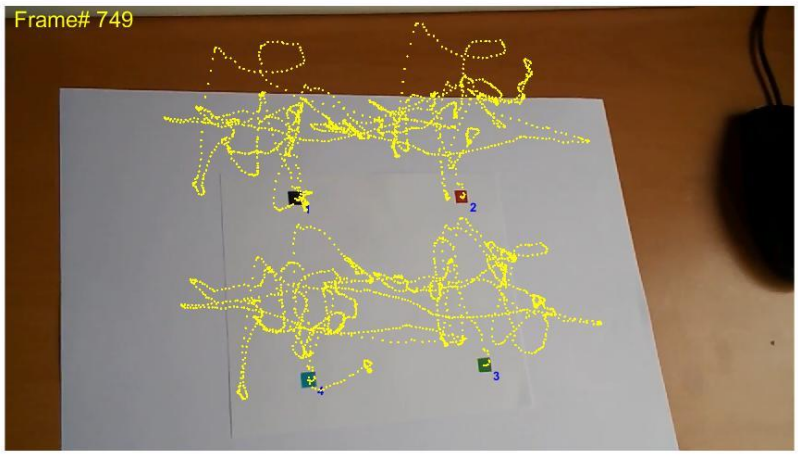
En nuestro caso han continuado en el cuadrado, pero la estabilización de estos se ha perdido.

---------- Hasta aquí 90 % de la nota ----------

Adjuntad vuestro código con la actualización del color de referencia. Si ejecutáis de nuevo el script, el tracking debería verse más estable.

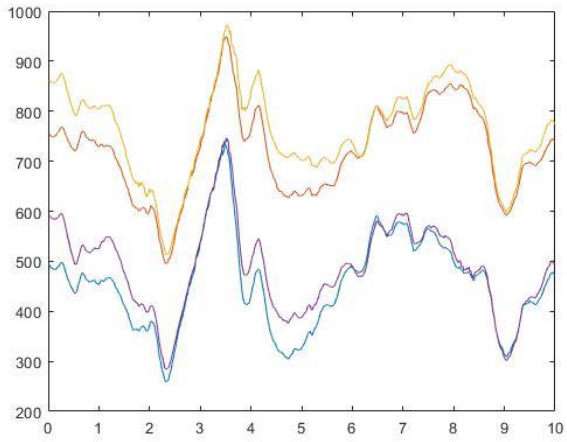
```
Ar = R.*w;
col(1,j)=mean2(sum(Ar(:)));
Ag = G.*w;
col(2,j)=mean2(sum(Ag(:)));
Ab = B.*w;
col(3,j)=mean2(sum(Ab(:)));
```

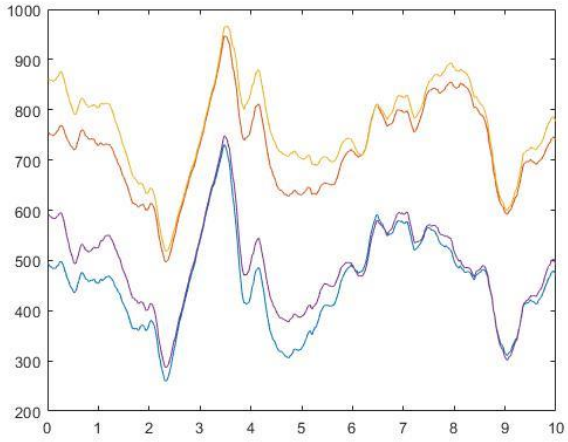
Adjuntad foto final.



Adjuntad vuestros propios "plots" correspondientes a las coordenadas V (y's') para el caso SIN y CON estabilización. Indicad sobre los plots en qué tiempo del video se notan más los efectos de la estabilización en la trayectoria.

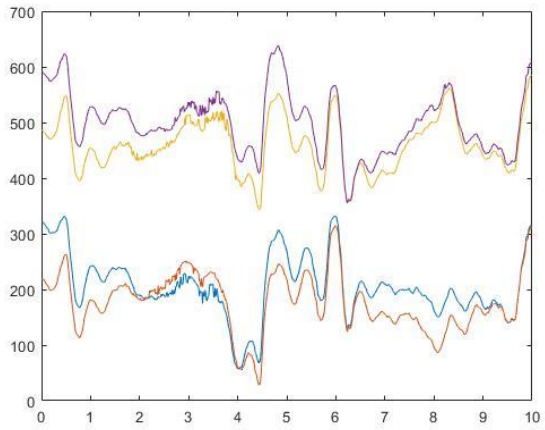
Plot de U sin estabilización y con estabilización:

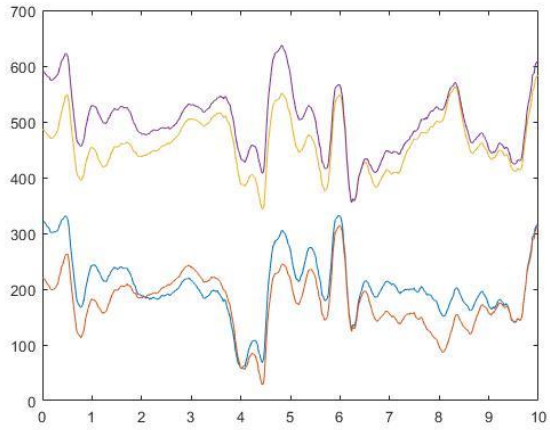




En este caso se pueden apreciar los efectos de la estabilización al girar en el eje X. Sin embargo estos no son tan significativos como veremos más adelante en las y's.

Plot de V sin estabilización y con estabilización:





En este caso los efectos de la estabilización son más significativos. Este efecto se puede apreciar entre el segundo 2 y 4. Esto se debe al girar sobre el eje y.

Adjuntar código completo de vuestro script track_video con todas las modificaciones indicadas

```
clear

 X=[490 754 862 591];
 Y=[320 220 485 591];

 col = [  13  133 32 7
          10  30 86 90
          12  16 31 104 ];

 obj=VideoReader('./color.mp4');
```

```
 NF = get(obj, 'NumberOfFrames');

 figure(1);
 frame=read(obj,1); im_obj=imshow(frame); hold on;
 pp_obj=plot(X,Y,'yo','MarkerFaceCol','y','MarkerSize',4);
 tt=text(X+20,Y+20,['1';'2';'3';'4']);
 t_frame=text(15,20,'Frame# 001','Color','y','FontSize',18);
 set(tt,'FontWeight','Bold','Color',[0 0 1]);
 hold off

 pause

RAD=25; % Zona de exploracion
dx=ones(2*RAD+1,1)*(-RAD:RAD);
dy=(-RAD:RAD)'*ones(1,2*RAD+1);

%Creamos U y V para guardar la posicion de las X's y las Y's
U = zeros(NF,4);
V = zeros(NF,4);
% Calculamos variables para el futuro calculo del tiempo en cada frame
fps = get(obj,'FrameRate');
T = NF/fps; %tiempo total del video (segundos)
% inicializamos variable para guardar el tiempo de cada frame
tiempo=zeros(NF,4);

for k=1:NF

  frame=read(obj,k); set(im_obj,'Cdata',frame);

 % Bucle actualizando posiciones X(j),Y(j) de las 4 esquinas
  for j=1:4
     % Calculamos coordenadas
     x=round(X(j));
     y=round(Y(j));
     rx=x+(-RAD:RAD);
     ry=y+(-RAD:RAD);
     % Extraemos subimagen
     sub_img=frame(ry,rx,:);
     % Sacamos los componentes
     R=double(sub_img(:,:,1));
     G=double(sub_img(:,:,2));
     B=double(sub_img(:,:,3));
     % Calculamos la diferencia en canales
     T=10;
     dr=(R-col(1,j))./T;
     dg=(G-col(2,j))./T;
     db=(B-col(3,j))./T;
     % Calculamos w
     w=exp(-(dr.^2 + dg.^2 + db.^2));
     % Normalizamos w
     w=w./sum(w(:));
     % Calculamos estimacion y actualizamos
     Ax=(x+dx).*w;
     X(j) = sum(Ax(:));
     Ay=(y+dy).*w;
     Y(j) = sum(Ay(:));
     % Estabilización del color
     Ar = R.*w;
     col(1,j)=mean2(sum(Ar(:)));
     Ag = G.*w;
     col(2,j)=mean2(sum(Ag(:)));
```

```
      Ab = B.*w;
      col(3,j)=mean2(sum(Ab(:)));
      U(k,j)=X(j);
      V(k,j)=Y(j);
      % calculamos tiempo para hacer el plot de U y V
      tiempo(k,j)= (T/NF) *k;

   end


 % Actualizar plot y etiquetas de los puntos sobre la imagen.
 %set(pp_obj,'Xdata',X,'Ydata',Y);
 hold on;
 plot(X,Y,'y.');
 hold off;
 for z=1:4, set(tt(z) ,'Position',[X(z)+15 Y(z)+15 0]); end
 set(t_frame,'String',sprintf('Frame# %03d',k));
 drawnow
end

% Mostramos plot de U en la figura 2 y plot de V en la figura 3
figure();
plot(tiempo,U);
figure();
plot(tiempo,V);
```