

## Proyecto (entrega definitiva en 10 días)

<b>Apellidos, Nombre:</b> Fernández Cabrita, Alejandro
--

<b>Apellidos, Nombre:</b> Evangelista Sarabia, Carlos
---

<b>Apellidos, Nombre:</b> De Almeida Pissarra, Inês
---

En la entrega, junto con este fichero de respuestas (doc o pdf) entregad un script por cada uno de los dos apartados:

1. **video.m** (completado con vuestro código de tracking del apartado 1)
2. **mosaico.m** (con vuestra versión más avanzada del apartado 2)

Entregadlos en un zip o similar. Para la entrega de control en la clase podéis subir sólo esta documento de respuestas.

### 1. MANEJO de VIDEOS en MATLAB (35%)

Pos	1er	2º	3er	4º
X	540	644	648	544
Y	370	370	473	473

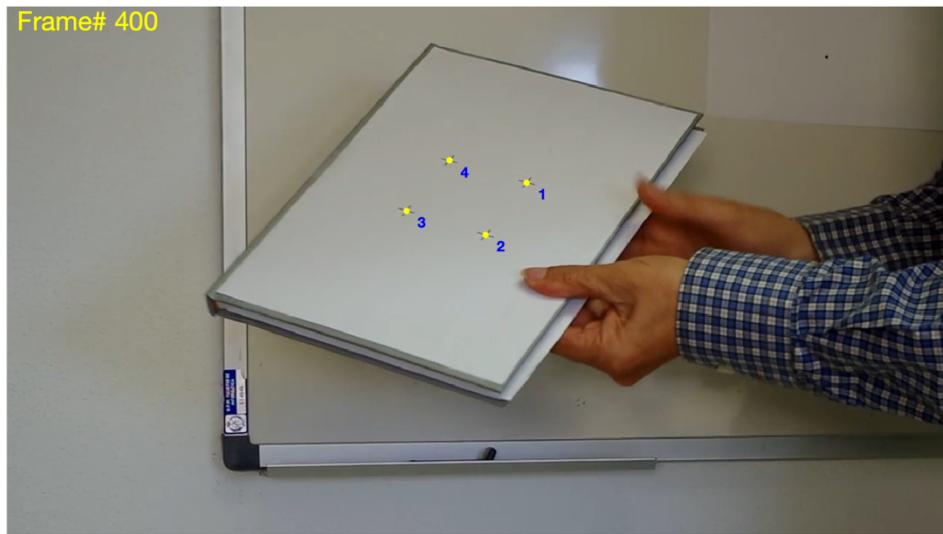
Adjuntad código dentro del bucle en j con la actualización de las posiciones X(j),Y(j)

```

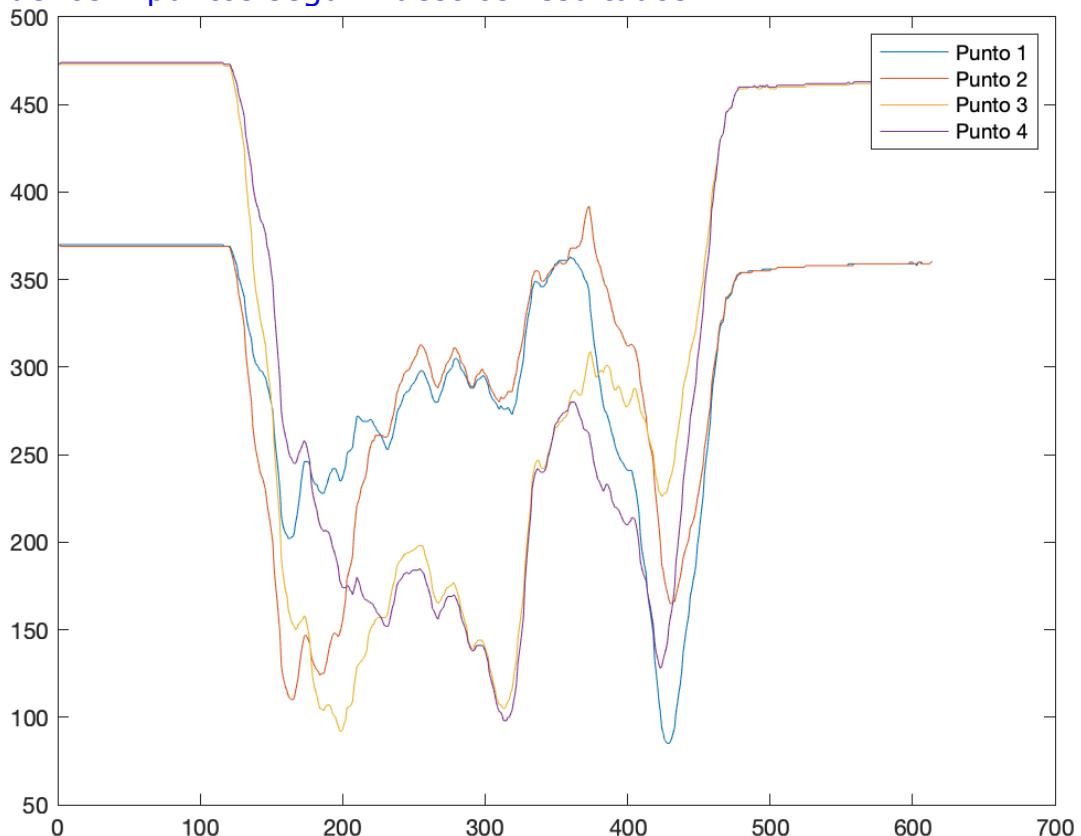
for j=1:4
    x=round(X(j)); y=round(Y(j));
    S=im2double(frame_gray((y+r),(x+r)));
    S0=min(S(:));
    d = abs(S-S0);
    w=exp(-50*d);
    w=w/sum(w(:));
    mult=w.*(x+dx);
    X(j) = sum(mult(:));
    mult2=w.*(y+dy);
    Y(j) = sum(mult2(:));
end

```

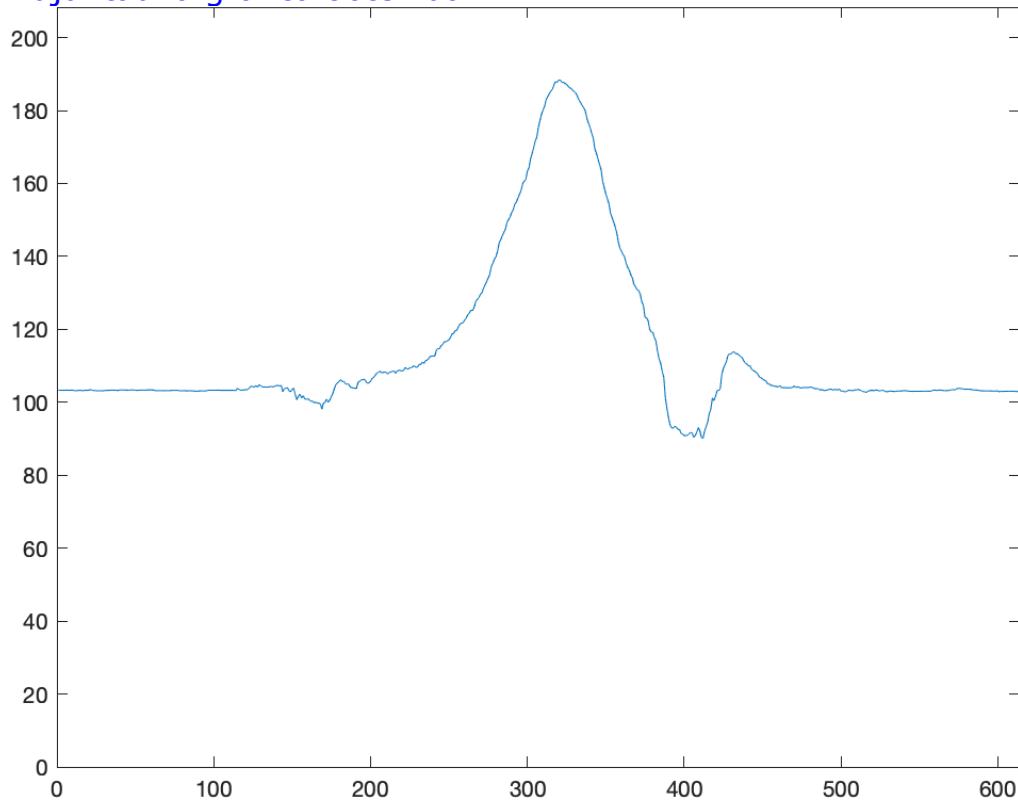
Adjuntad una captura con el frame 400 de la película.



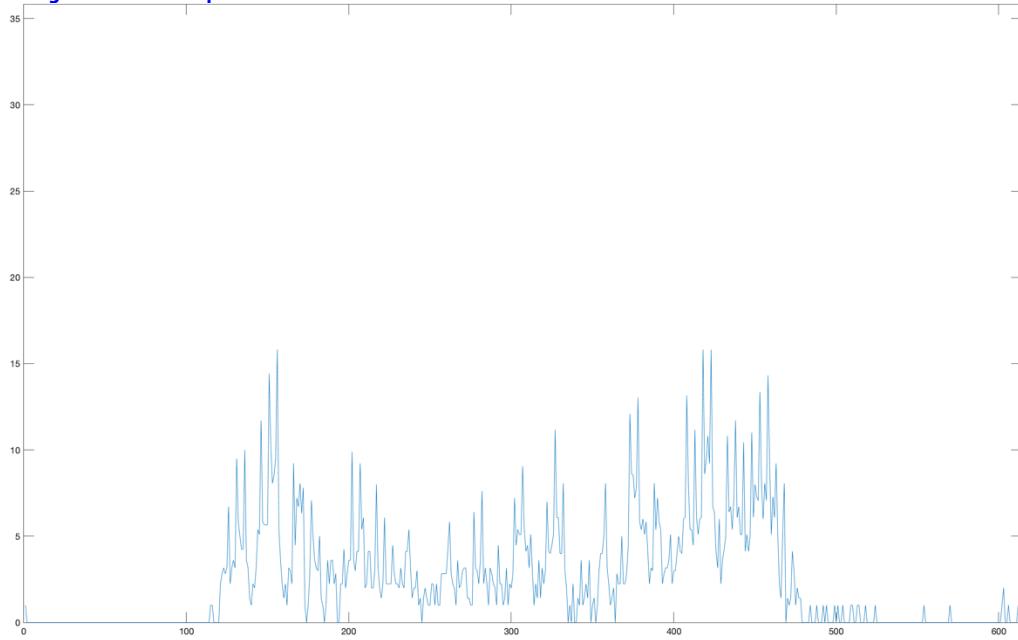
Adjuntad la misma gráfica pero ahora mostrando la evolución de las coordenadas Y de los 4 puntos según vuestros resultados.



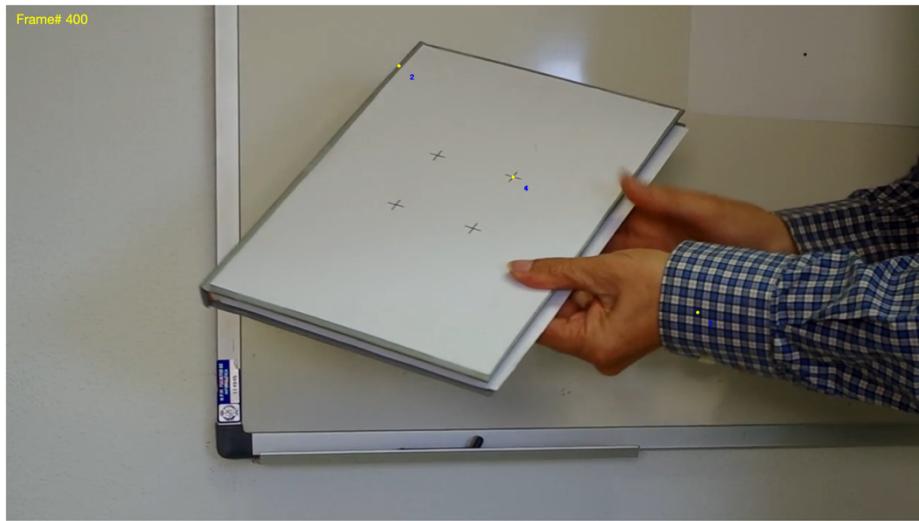
Adjuntad la gráfica obtenida.



Adjuntad el plot con las diferencias dx.



Volver a correr el programa con  $R=10$  y volver a adjuntad una captura de la película (con los puntos superpuestos) para el frame #400. ¿Qué sucede en este caso?



En este caso,  $R$  es demasiado pequeño y los puntos de la imagen a veces ya no están dentro de la subimagen, lo que hace que se elijan coordenadas incorrectas para los puntos (se eligen las coordenadas de los puntos más oscuros de la subimagen, pero no contiene los puntos previstos).

## 2. CREACIÓN de MOSAICOS (65%)

### 2.1 Primer mosaico (15%)

Adjuntad código e imagen resultante.

```
im=zeros(3456,5760);
```

```
rx=(1:192);
ry=(1:288);

for j = 1:12
    for i = 1:30
        im(ry,rx) = imgs(:,:,(j-1) * 30 + i);
        rx = rx + 192;
    end
    ry = ry + 288;
    rx=(1:192);
end

imshow(im);
```



Adjuntad script con el código usado. Ejecutadlo (con un umbral U=0.4) y adjuntad (imshow) la imagen mosaico resultante.

```
L = 360; U = 0.4;
M = 192; N = 288;

im_original = imread("target.jpg");
im_original = im2double(im_original);

dims_target = size(im_original);
im = zeros(dims_target(1),dims_target(2));
vertical = dims_target(1)/N;
horizontal = dims_target(2)/M;

rx=(1:M); ry=(1:N);
for j = 1:vertical
    for i = 1:horizontal
        im(ry,rx) = imags(:, :, (j-1) * horizontal + i);
        rx = rx + M;
    end
    ry = ry + N;
    rx=(1:M);
end

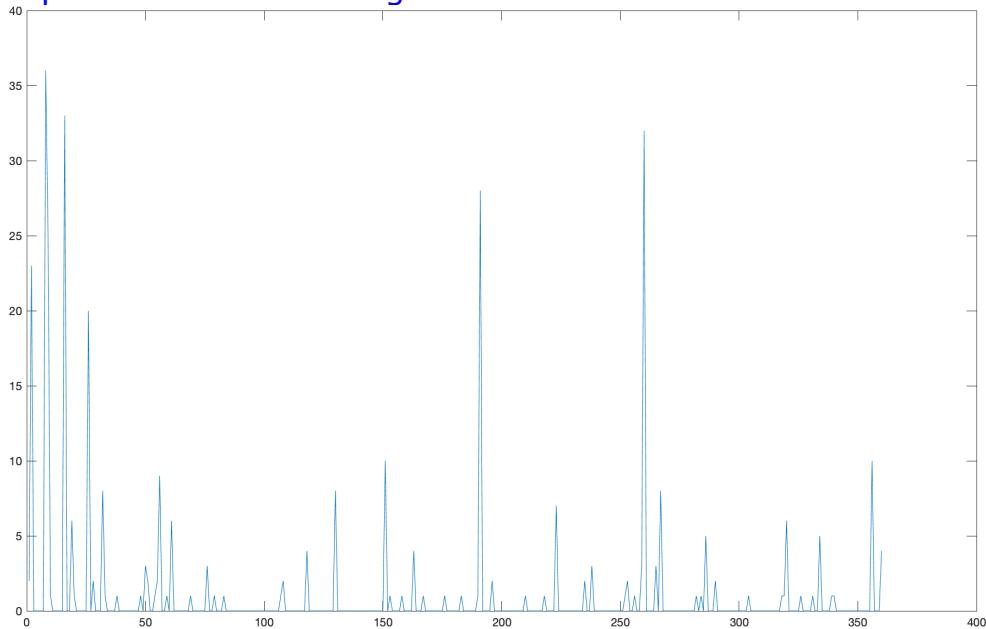
rx=(1:M); ry=(1:N);
for j = 1:vertical
    for i = 1:horizontal
        sub_im = im_original(ry, rx);
        seleccionado = im(ry, rx);
        e0 = mean2(abs(sub_im - seleccionado));
        min = e0;
        for n=1:L
            e = mean2(abs(sub_im - imags(:, :, n)));
            if e/e0<U
                min = e;
                k = n;
                seleccionado = imags(:, :, n);
                break;
            elseif e < min
                k = n;
                min = e;
                seleccionado = imags(:, :, n);
            end
        end
        im(ry,rx) = seleccionado;
        rx = rx + M;
    end
    ry = ry + N;
    rx=(1:M);
end

imshow(im);
```



## 2.2 Penalización para igualar el uso de las diferentes imágenes (15%)

Al terminar, haced un plot del vector usos y adjuntadlo. ¿Cuántas imágenes son usadas más de 20 veces? ¿Qué tanto por ciento del total de imágenes usadas (360) representan dichas imágenes?



Usadas más de 20 veces: 6  
Qué representa 1,67% del total de imágenes usadas (360)

Adjuntad modificaciones del código, el mosaico obtenido y la gráfica final del vector usos con el nº de veces que cada imagen es usada. ¿Cuál es ahora el número máximo de veces que una imagen es usada?

...

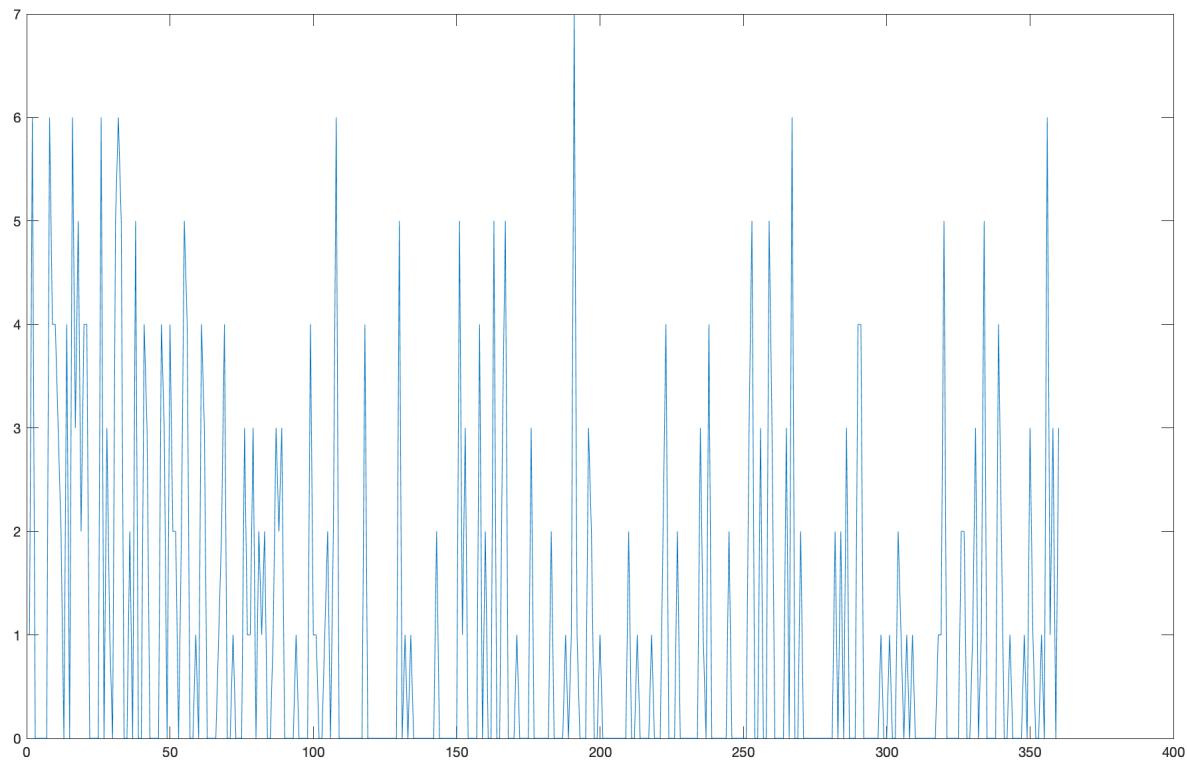
```

timeout=zeros(1,L); % agregado
PENALTY = 50; % agregado

for j = 1:vertical
    for i = 1:horizontal
        timeout = timeout - 1; % agregado
        sub_im = im_original(ry, rx);
        seleccionado = im(ry, rx);
        e0 = mean2(abs(sub_im - seleccionado));
        min = e0;
        for n=1:L
            if timeout(n)>0 % agregado
                continue;
            end
            e = mean2(abs(sub_im - imagenes(:,:,n)));
            if e/e0<U
                min = e;
                k = n;
                seleccionado = imagenes(:,:,n);
                break;
            elseif e < min
                k = n;
                min = e;
                seleccionado = imagenes(:,:,n);
            end
        end
        im(ry,rx) = seleccionado;
        if(min~=e0) % agregado
            timeout(k)=PENALTY;
        end
        rx = rx + M;
    end
    ry = ry + N;
    rx=(1:M);
end
imshow(im);

```





Ahora, el número máximo de veces que una imagen es usada es 7.

### 2.3 Combinación de diferentes escalas (15%)

Adjuntad el mosaico resultante.



En esta segunda pasada, de los  $24 \times 60 = 1440$  trozos que probamos, ¿qué porcentaje es sustituido por una de las nuevas fotos reducidas? De ellas, ¿cuántas se aceptan con el criterio de que el error sea menor que  $U \cdot e_0$ ?

Porcentaje de fotos substituidas: 56,736% (817 fotos)

Fotos substituidas por la condición: 220

Adjuntad mosaico final combinando las fotos con las 4 escalas distintas.



Extraer la zona del mosaico correspondiente al rango de filas de 700 a 1500 y columnas 2200 a 3500. Adjuntad una captura de esa zona.



En la última pasada (con las imágenes más pequeñas), ¿qué porcentaje de "trozos" del mosaico son substituidas?

Porcentaje de "trozos" del mosaico substituidos 28,038%

## 2.4 Modificando las imágenes de la biblioteca (20%)

Adjuntad código y el mosaico obtenido tras correr el nuevo algoritmo (solo para la primera escala, usando las imágenes de los retratos en su tamaño original).

```
L = 360; U = 0.4;
PENALTY = 50;
M = 192; N = 288;
```

```
im_original = imread("target.jpg");
im_original = im2double(im_original);
imagenes = imgs;

dims_target = size(im_original);
im = zeros(dims_target(1),dims_target(2));
vertical = dims_target(1)/N;
horizontal = dims_target(2)/M;

rx=(1:M); ry=(1:N);
for j = 1:vertical
    for i = 1:horizontal
        im(ry,rx) = imgs(:,:, (j-1) * horizontal + i);
        rx = rx + M;
    end
end
```

```

    ry = ry + N; rx=(1:M);
end

for r=1:1
    rx=(1:M); ry=(1:N);
    timeout=zeros(1,L);
    for j = 1:(vertical * 2^(r-1))
        for i = 1:(horizontal * 2^(r-1))
            timeout = timeout - 1;
            sub_im = im_original(ry, rx);
            seleccionado = im(ry, rx);
            e0 = mean2(abs(sub_im - seleccionado));
            min = e0;
            for n=1:L
                if timeout(n)>0
                    continue;
                end
                a=mean2(sub_im-imagenes(:,:,:,n));
                e = mean2(abs(sub_im - (Imagenes(:,:,:,n) + a)));
                if e/e0<U
                    k = n; min = e;
                    seleccionado = Imagenes(:,:,:,n) + a;
                    break;
                elseif e < min
                    k = n; min = e;
                    seleccionado = Imagenes(:,:,:,n) + a;
                end
            end
            if(min~=e0)
                timeout(k)=PENALTY;
            end
            im(ry,rx) = seleccionado;
            rx = rx + M;
        end
        ry = ry + N; rx=(1:M);
    end
    Imagenes=imresize(imagenes,1/2);
    PENALTY = PENALTY*2;
    U = sqrt(U);
    M = M/2; N = N/2;
end

figure(1);
imshow(im);

```



Adjuntad mosaico final.



Adjuntad un detalle del mosaico correspondiente al rango de filas (700:1500) y de columnas (2200:3500).

