

**Apellidos, Nombre: Ortiz Pasamontes, Enrique**

### Ejercicio 1:

Adjuntar los coeficientes (G) del filtro resultante.

G =

	1	2	3	4	5	6	7	8	9	10	11
1	0	0.0053	0.0061	0.0067	0.0071	0.0073	0.0071	0.0067	0.0061	0.0053	0.0044
2	0.0053	0.0063	0.0073	0.0080	0.0085	0.0087	0.0085	0.0080	0.0073	0.0063	0.0053
3	0.0061	0.0073	0.0083	0.0092	0.0098	0.0100	0.0098	0.0092	0.0083	0.0073	0.0061
4	0.0067	0.0080	0.0092	0.0102	0.0108	0.0110	0.0108	0.0102	0.0092	0.0080	0.0067
5	0.0071	0.0085	0.0098	0.0108	0.0115	0.0117	0.0115	0.0108	0.0098	0.0085	0.0071
6	0.0073	0.0087	0.0100	0.0110	0.0117	0.0120	0.0117	0.0110	0.0100	0.0087	0.0073
7	0.0071	0.0085	0.0098	0.0108	0.0115	0.0117	0.0115	0.0108	0.0098	0.0085	0.0071
8	0.0067	0.0080	0.0092	0.0102	0.0108	0.0110	0.0108	0.0102	0.0092	0.0080	0.0067
9	0.0061	0.0073	0.0083	0.0092	0.0098	0.0100	0.0098	0.0092	0.0083	0.0073	0.0061
10	0.0053	0.0063	0.0073	0.0080	0.0085	0.0087	0.0085	0.0080	0.0073	0.0063	0.0053
11	0.0044	0.0053	0.0061	0.0067	0.0071	0.0073	0.0071	0.0067	0.0061	0.0053	0.0044

Adjuntar ambas imágenes submuestreadas (con y sin filtro previo). ¿Se han reducido los efectos de "Moire"?

Hemos conseguido reducir los artefactos:



**Ejercicio 2:**

Adjuntar la figura resultante. Notad como la textura del hielo se ve más marcada en la segunda imagen. ¿Notáis algún efecto indeseable en la imagen procesada?

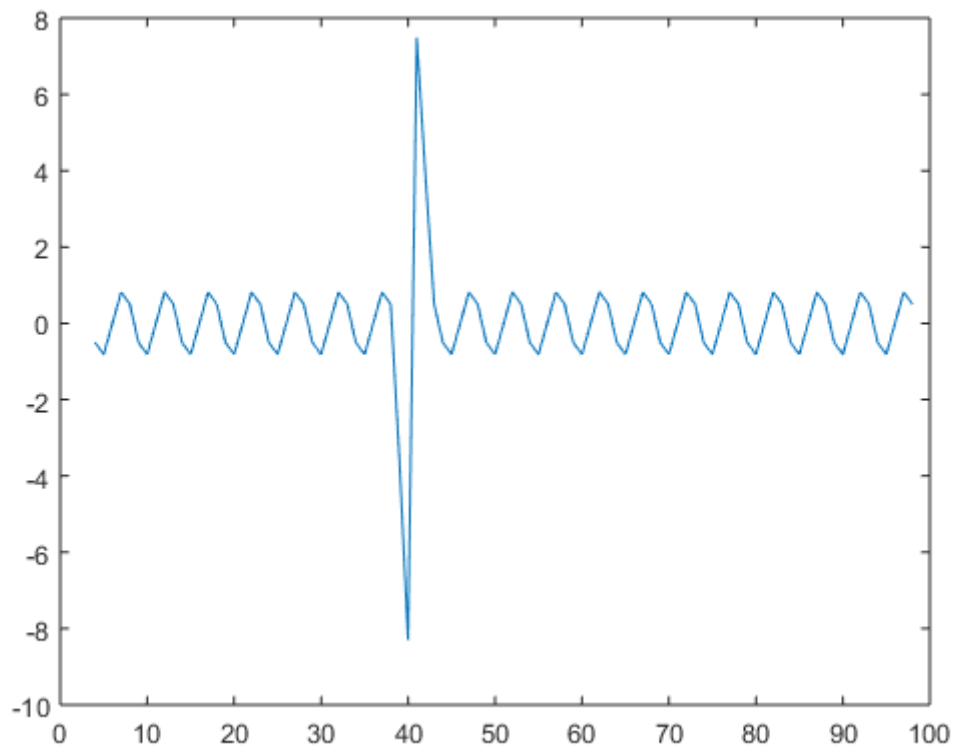
Aparece un halo más claro alrededor de los bordes.



Hacer un plot del "detalle" extraído y adjuntar imagen resultante.

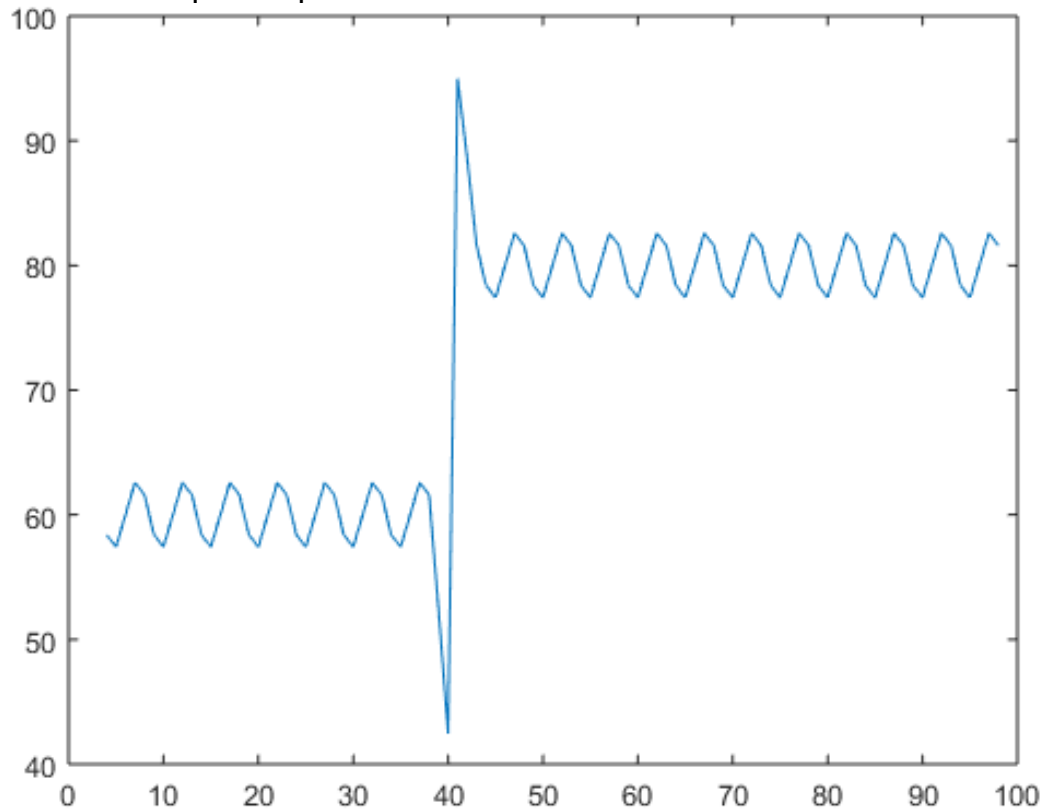
¿Hemos capturado las pequeñas oscilaciones que simulaban el detalle de la imagen?  
¿Qué más aparece en la gráfica?

Hemos conseguido quedarnos con el detalle, pero en la zona donde hay una transición abrupta (el borde) añadimos un efecto indeseado.



Superponer una "imagen" de la señal original ( $y$ ) y de una señal con el detalle resaltado ( $yy + 3*\text{detalle}$ ). Adjuntar la gráfica resultante. ¿Se ha magnificado el detalle? ¿Qué ocurre en el borde?

Tal y como esperábamos hemos conseguido aumentar el detalle, pero también hemos añadido el error que se producía en el borde brusco.



# Proyecto: representaciones en pirámide. (a entregar, por parejas, dentro de dos semanas)

## 1) Implementación de la pirámide Laplaciana

Adjuntar código de vuestra función lap.m

```
function p=lap(im,N)
if nargin==1, N=5; end
p=cell(1,N);

% Propiedades del filtro
soporte = 11;
s = 3;

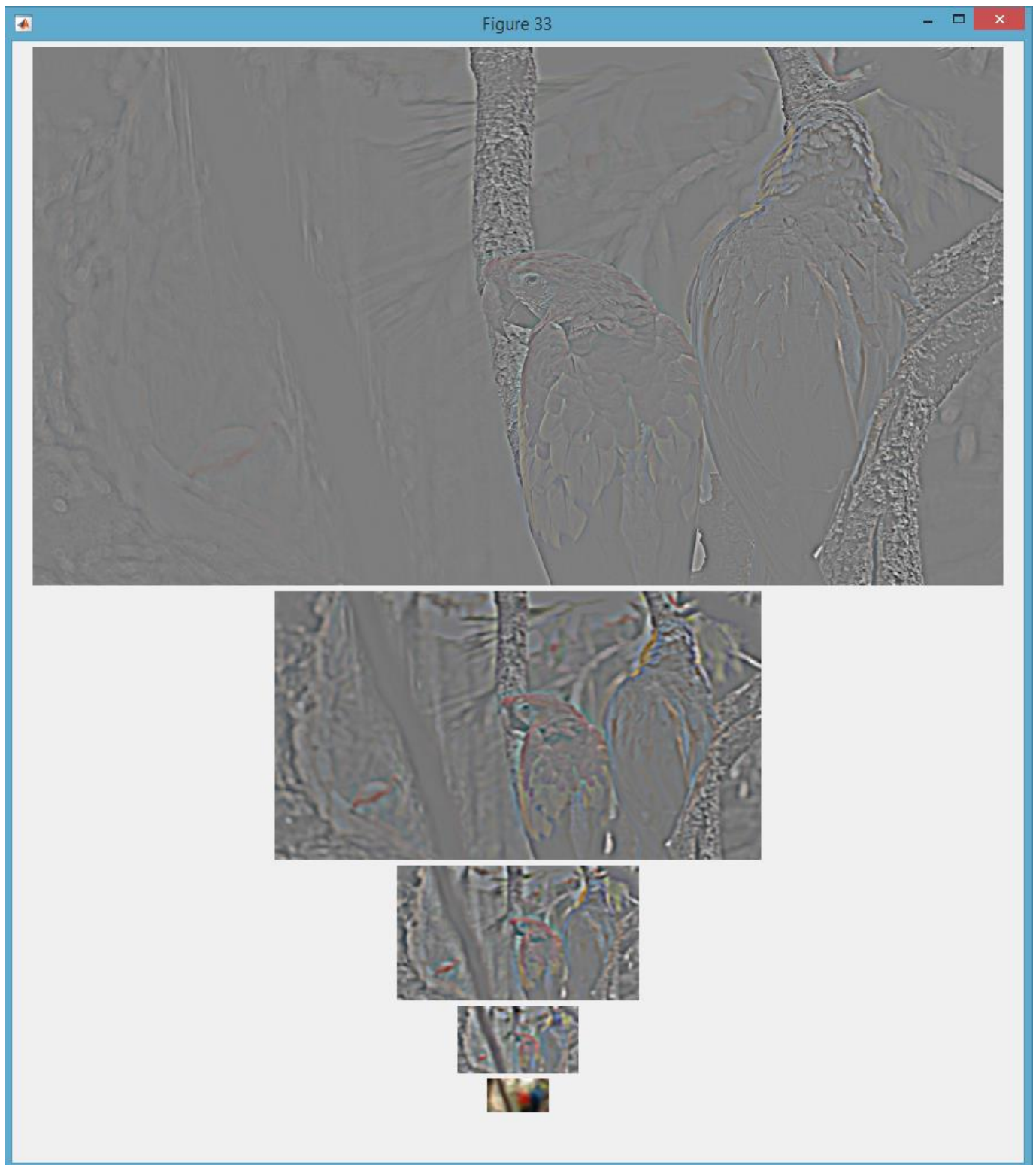
% Crear filtro gaussiano 2D a usar
x = [-floor(soporte/2):floor(soporte/2)];
h = exp(-0.5*(x/s).^2)/(sqrt(2*pi)*s); % Expresion gauss 1D
h = h/sum(h); % Normaliza para que suma=1
filtro_gausiano = h' * h; % Crea mascara 2D

% Crear filtro laplaciano
filtro_laplaciano = -filtro_gausiano;
centro_matriz = floor(soporte/2)+1;
filtro_laplaciano(centro_matriz,centro_matriz) =
filtro_laplaciano(centro_matriz,centro_matriz)+1;

% Calcular los niveles p{1}, p{2}, ..., p{N} de la piramide laplaciana
% Recordad que p{N} es la versión reducida de la imagen
im=double(im);
for k=1:N-1,
    p{k} = imfilter(im, filtro_laplaciano, 'symmetric');
    im = imfilter(im, filtro_gausiano, 'symmetric');
    im = im(1:2:end,1:2:end,:);
end
p{N}=im;

return
```

Adjuntar vuestra imagen de la pirámide resultante usando visualiza\_lap



## **2) Inversa de la pirámide Laplaciana**

### Adjuntar código de vuestra función invlap.m

```
function im=invlap(p)
N = length(p); % Número de niveles de la pirámide
im=p{N}; % Inicializamos con version tamaño sello.

for k=N-1:-1:1,
    im = amplia_2(im);
    im = im + p{k};
    %     figure; image(uint8(im));
end
im = uint8(im);
return
```

Comando usado para calcular la discrepancia entre las imágenes original y recuperada. ¿De que orden es dicha diferencia?

La diferencia es de un orden de  $10^{-2}$ .

```
diferencia = mean2(abs(imagen_original-imagen_final));
diferencia =
    0.0114
```



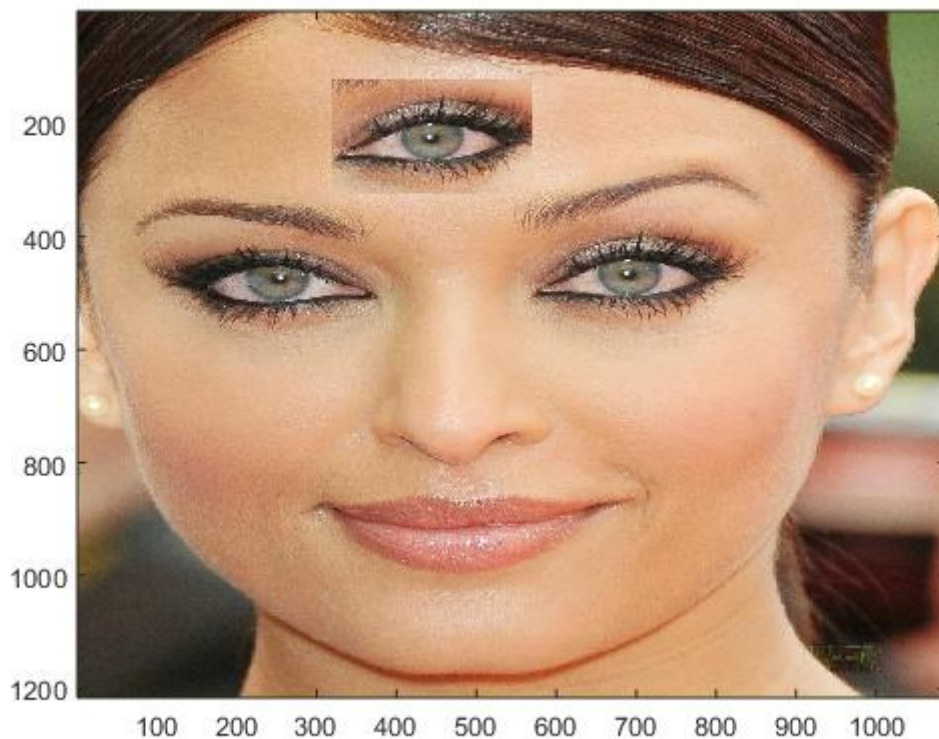
## **Aplicación: Fusión de imágenes con pirámide Laplaciana.**

Adjuntar vuestra imagen im1. Hacer una fusión directa con la máscara binaria. Adjuntar código e imagen final.

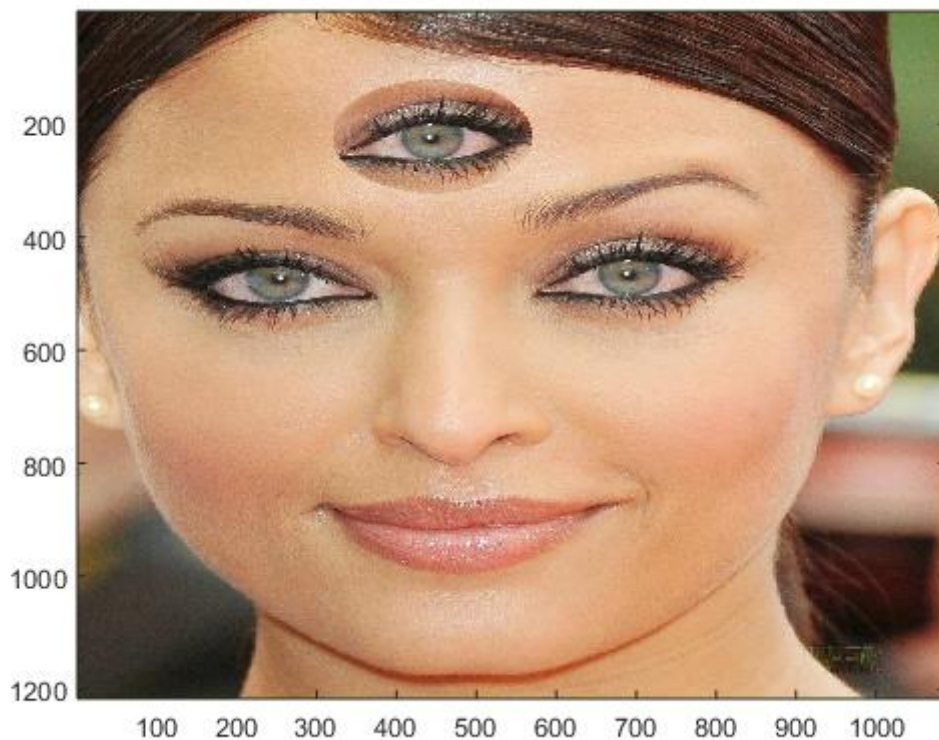
```
% Limpieza
clear all;
close all;

% Definimos las posiciones del recorte
x0=692; y0=463;
x1=445; y1=222;
xwidth = 250; ywidth=200;

% Obtenemos las imagenes a operar
im0 = imread('face.jpg');
im1 = im0;
im1(y1-ywidth/2:1:y1+ywidth/2,x1-xwidth/2:1:x1+xwidth/2,:) = im1(y0-
ywidth/2:1:y0+ywidth/2,x0-xwidth/2:1:x0+xwidth/2,:);
image(im1);
% im=double(im);
mask = crea_mask(im0,[x1 y1],[xwidth/2 ywidth/2]); % El ancho es medido desde el
centro por eso /2
im0(mask)=im1(mask);
figure;
image(im0);
```







Hacer fusión usando pirámides tal como se ha descrito anteriormente, usando 5 o 6 niveles en las pirámides (tanto en las pirámides laplacianas de las imágenes como en la pirámide gaussiana de la máscara).

Adjuntar vuestro código y la imagen final.

```
% Limpieza
clear all;
close all;

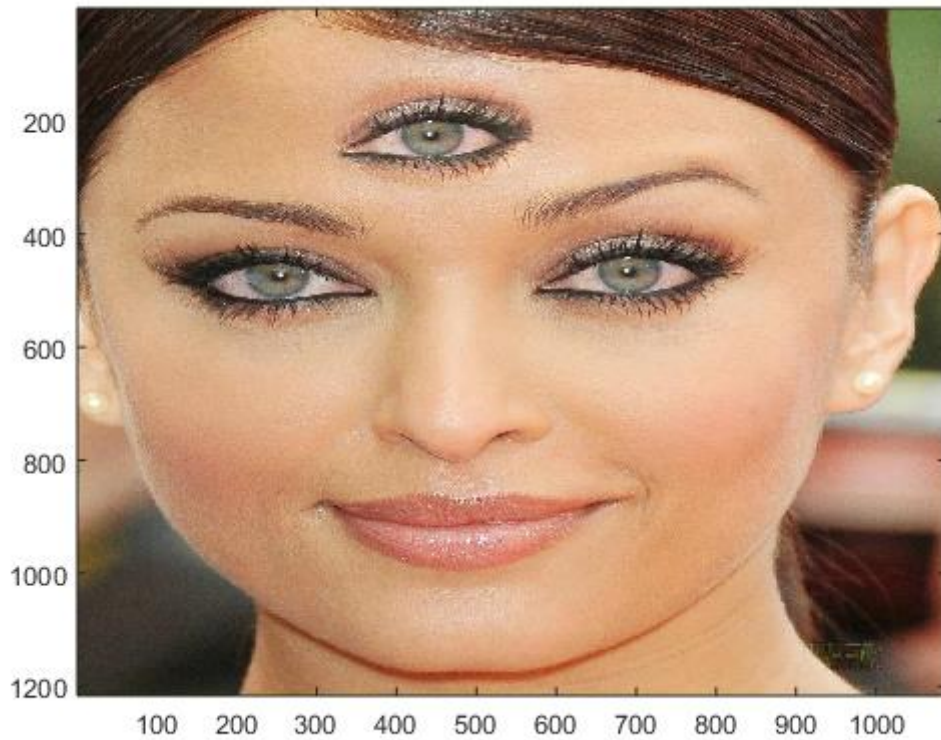
% Definimos los niveles que va a tener la piramide
levels = 6;

% Definimos las posiciones del recorte
x0=692; y0=463;
x1=445; y1=222;
xwidth = 250; ywidth=200;

% Obtenemos las imagenes a operar
im0 = imread('face.jpg');
im1 = im0;
im1(y1-ywidth/2:1:y1+ywidth/2,x1-xwidth/2:1:x1+xwidth/2,:) = im1(y0-
ywidth/2:1:y0+ywidth/2,x0-xwidth/2:1:x0+xwidth/2,:);
image(im1);
% im=double(im);
mask = crea_mask(im0,[x1 y1],[xwidth/2 ywidth/2]); % El ancho es medido desde el
centro por eso /2

% Creamos piramides de las imagenes a fusionar y la mascara
p0 = lap(im0,levels);
p1 = lap(im1,levels);
```

```
m = gauss(mask, levels);  
% figure; image(mask);  
  
% Fusionamos  
new = cell(1, levels);  
for k=levels:-1:1,  
    new{k} = (1-m{k}).*p0{k} + m{k}.*p1{k};  
end  
fusion = invlap(new);  
  
figure; image(fusion);
```



**EXTRAS:** Cuando hayáis replicado estos resultados, usar vuestro programa con una foto o fotos de vuestra elección. Tened en cuenta que:

- Dependiendo del objeto a "fusionar" seguramente tendréis que cambiar el tamaño de la máscara usando el tercer argumento opcional de la función `crea_mascara` explicado antes.
- Vuestra función `lap` (e `inv_lap`) solo va a funcionar correctamente si las dimensiones (alto/ancho) de la imagen son múltiplos de una potencia de 2 suficientemente alta para seguir siendo enteros después de las sucesivas divisiones por 2 en cada nivel de la pirámide. Recortar las imágenes a usar para que tengan las dimensiones adecuadas.

¿Cómo podríais modificar `lap` (e `invlap`) para que funcionaran con imágenes de un tamaño arbitrario?

Una solución sería recortar la imagen a algún tamaño compatible, pero la mejor forma sería poder operar con los decimales reescalando las imágenes para que coincidiesen entre sí cuando no lo hagan directamente. Esto se haría calculando los pesos de cada pixel de la imagen para cada pixel final al que tiene que parar la imagen.



