

Apellidos, Nombre: Fernández Cabrita, Alejandro
--

Apellidos, Nombre: Evangelista Sarabia, Carlos

Apellidos, Nombre: De Almeida Pissarra, Inês

1. Realce de bordes: filtro de nitidez o unsharp (15%)

Adjuntad código de vuestra función y la imagen obtenida para el detalle anterior.

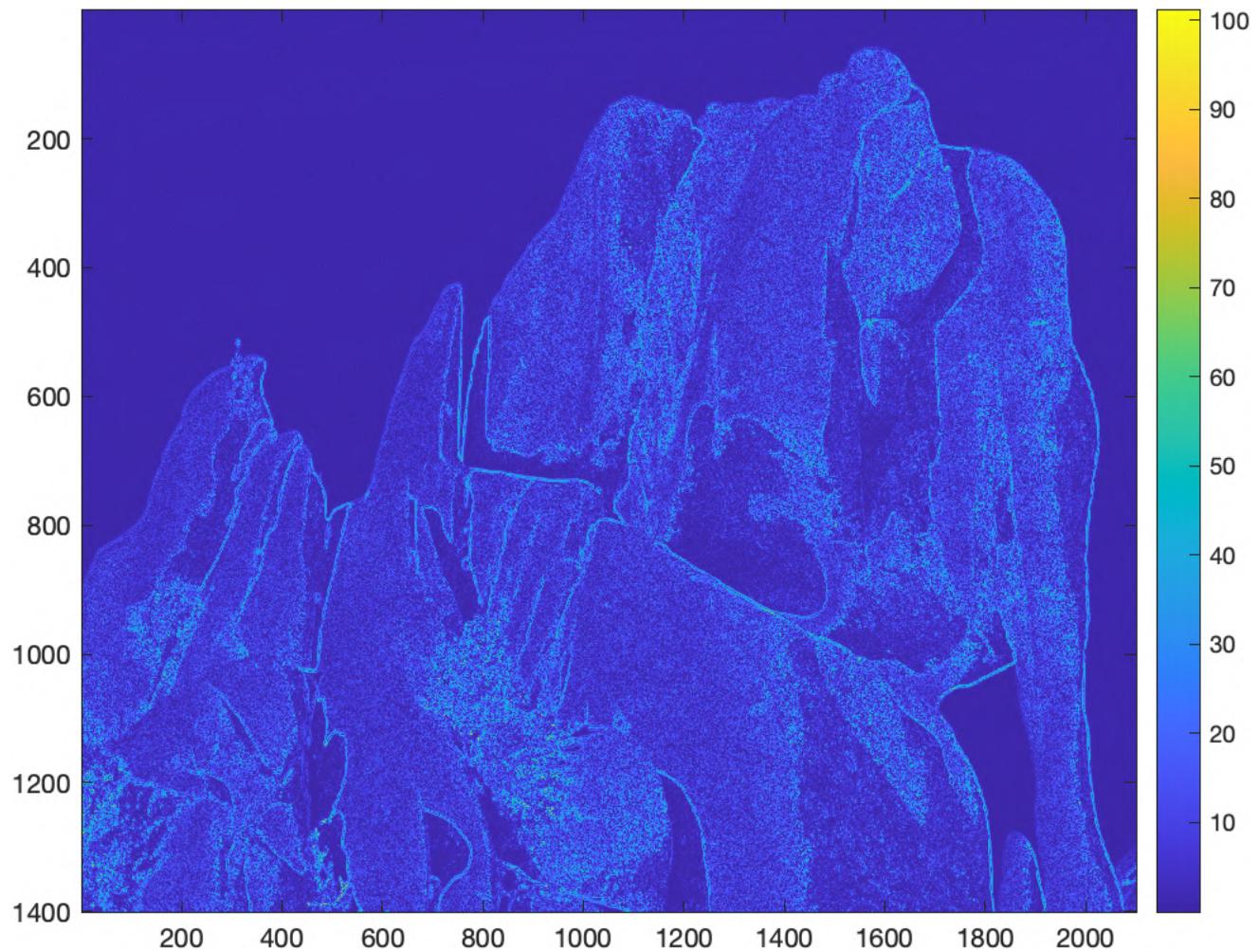
```
function show_detail(det)
```

```
    abs_det = abs(det);
    R = abs_det(:,:,1);
    G = abs_det(:,:,2);
    B = abs_det(:,:,3);

    det_final = 0.3*R+0.55*G+0.15*B;
```

```
    figure();
    imagesc(det_final);
    colorbar('vert');
```

```
end
```



Adjuntad la imagen "compuesta".



Adjuntad zoom. ¿A qué son debidos estos halos que aparecen?



El filtro gaussiano suaviza linealmente, causando um efecto borrado en los detalles de los bordes. Cuando se aplica una cantidad de filtro elevada (en este caso lo hicimos con el 150% del detalle), es creado um halo alrededor de las regiones de alto contraste (causando un efecto de "brillo" que recorre todo el borde).

¿Qué rango de valores típicos observáis en ambas zonas?

Cielo -> 0 ~< index ~< 3

Roca -> 0 ~< index ~< 50

2. Implementación de un filtrado bilateral (30%)

Calcular la diferencia ($f_1 - f_2$) entre los resultados y volcar su máximo y mínimo (que debería ser del orden de la precisión de la máquina $\sim 10^{-16}$).

Min = -5.1159e-13

Max = 5.9686e-13

Una vez verificada, adjuntad el código de vuestra implementación.

```
function res=filtro_bilat(im,S, R)

L = round(S*2);
Gs = fspecial('gaussian',2*L+1,S); %Creación máscara gaussiana
im=double(im);

[N,M,P]=size(im);

% Añadimos margen de L filas y columnas por cada lado
% repitiendo la 1ª/última fila y la 1ª/última columna
% Equivale a opción symmetric en imfilter
im=[im(:,L:-1:1,:); im(:,M:-1:M-L+1,:)];
im=[im(L:-1:1,:,:); im; im(N:-1:N-L+1,:,:)];

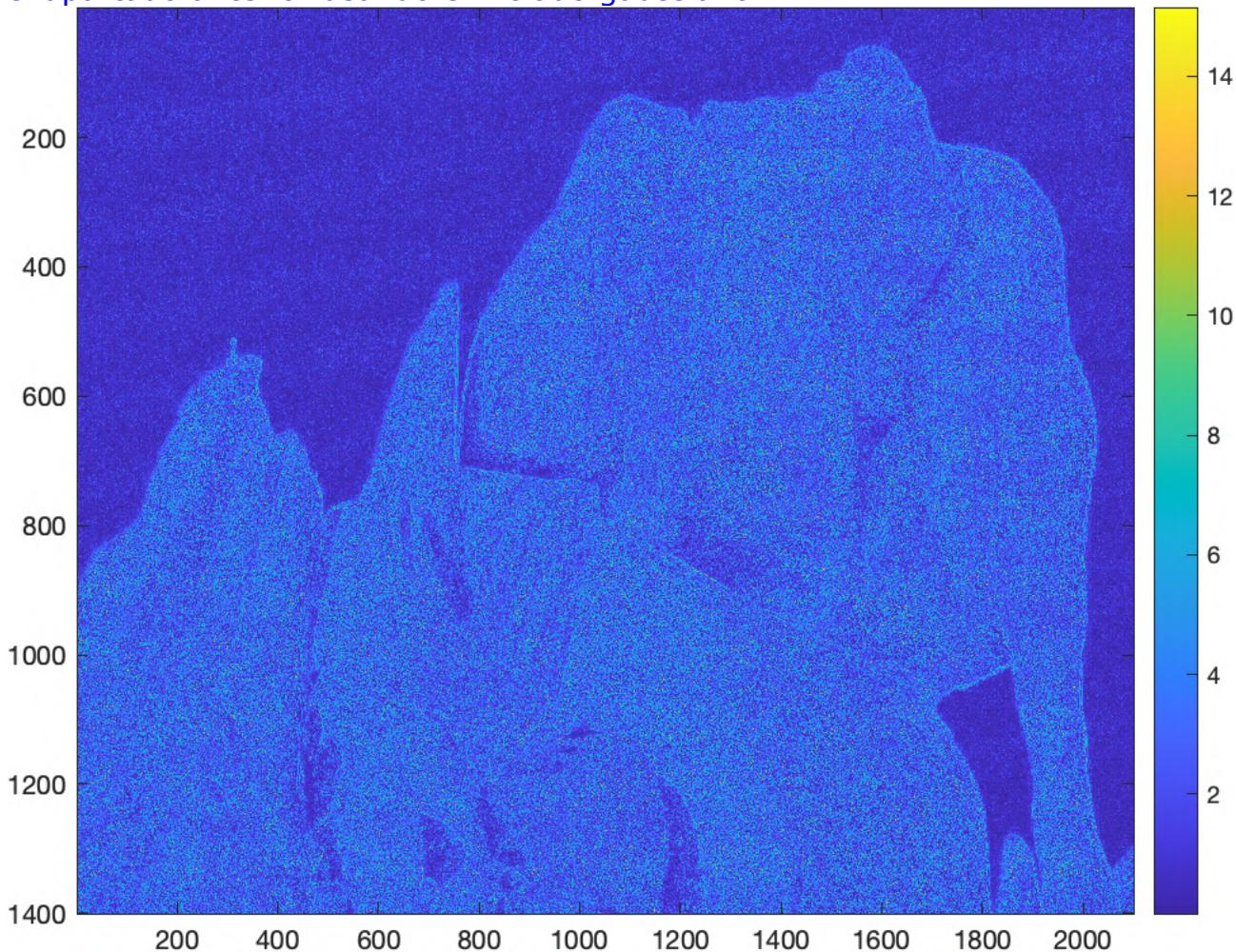
res=im*0; s=(-L:L);
for k=L+1:L+N
    for j=L+1:L+M
        vec = im(k+s,j+s,:);
        D = (vec - im(k,j,:))/R;
        D2 = sum(D.^2, 3);
        Gr= exp(-0.5*D2);
        G = Gr.*Gs;
        G = G/sum(G(:));
        vec = vec.*G;
        res(k,j,:)= sum(sum(vec));
    end
end

% Quitamos el margen que le hemos añadido al principio
res = res(L+(1:N),L+(1:M),:);

return
```

Como nuestra implementación será bastante lenta, para hacer pruebas y comparar con la función de MATLAB podéis usar una versión reducida de la imagen haciendo p.e. `im=imresize(im,[400 600])`. Cuando os funcione dar los resultados para la imagen original.

Adjuntad la imagen del detalle resultante. ¿Qué rango de valores tenemos ahora en el detalle? ¿Cuál es la mayor diferencia con la imagen del detalle que se obtuvo en el apartado anterior usando el filtrado gaussiano?



Tenemos el rango [0,15].

Valores típicos:

Cielo -> 0 ~< index ~< 3

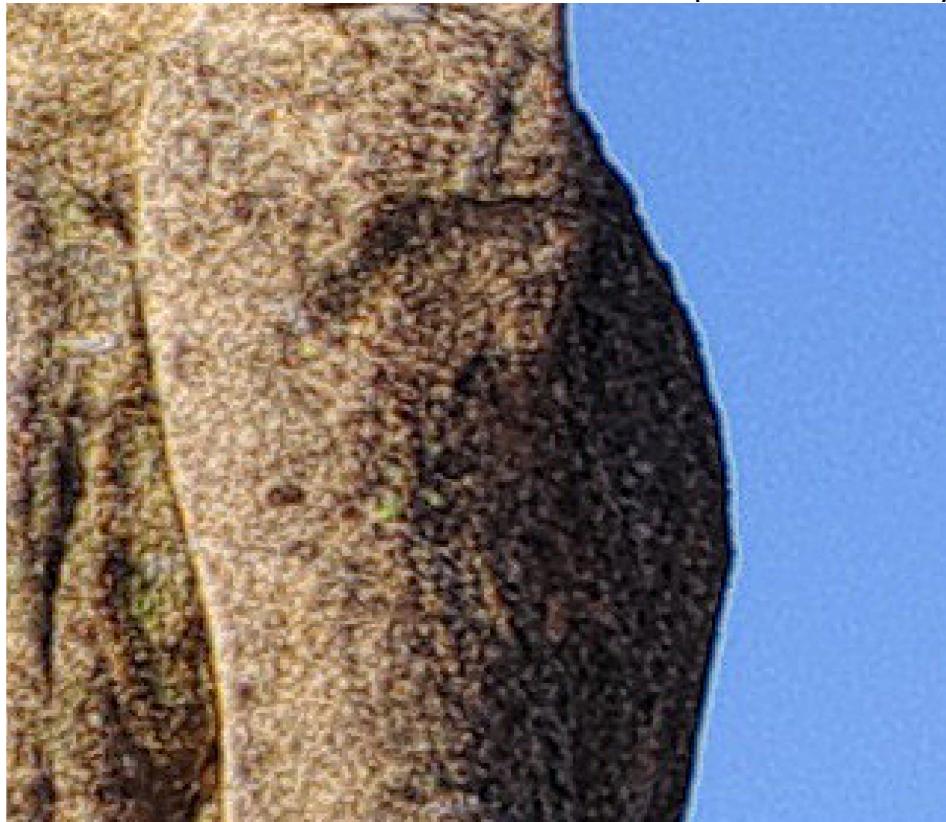
Roca -> 0 ~< index ~< 10

La mayor diferencia se nota principalmente en la zona de la roca (y en sus bordas), donde se pierde mucho menos detalle. El filtro bilateral, a diferencia del filtro gaussiano, preserva las regiones de alto contraste (bordes) y los detalles finos.

Adjuntad la imagen realizada con $\alpha=250\%$.



Hacer zoom sobre una zona donde antes aparecían halos y adjuntad una captura.



Adjuntad las 2 imágenes resultantes.

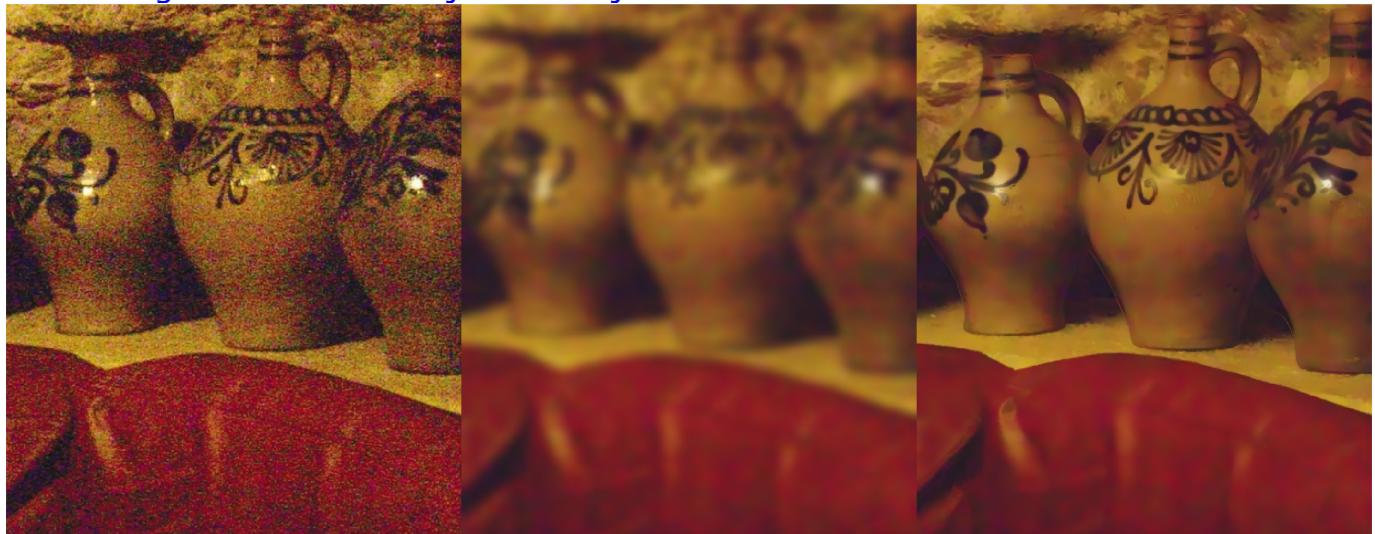


2b. Cross-bilateral filter

Adjuntad las línea/s de código cambiadas en vuestra función

```
function res=filtro_cross(im,im2,S,R)
...
im2=double(im2);
% Añadimos margen de L filas y columnas por cada lado
% repitiendo la 1ª/última fila y la 1ª/última columna
% Equivale a opción symmetric en imfilter
% im2 tiene las mismas dimensiones que im
im2=[im2(:,L:-1:1,:);im2(:,M:-1:M-L+1,:)];
im2=[im2(L:-1:1,:,:);im2;im2(N:-1:N-L+1,:,:)];
...
vec2 = im2(k+s,j+s,:);
D = (vec2 - im2(k,j,:))/R;
...
end
```

Adjuntad figura. ¿Por qué con el filtro bilateral se preservan mejor los detalles finos de la imagen como los dibujos de los jarrones?

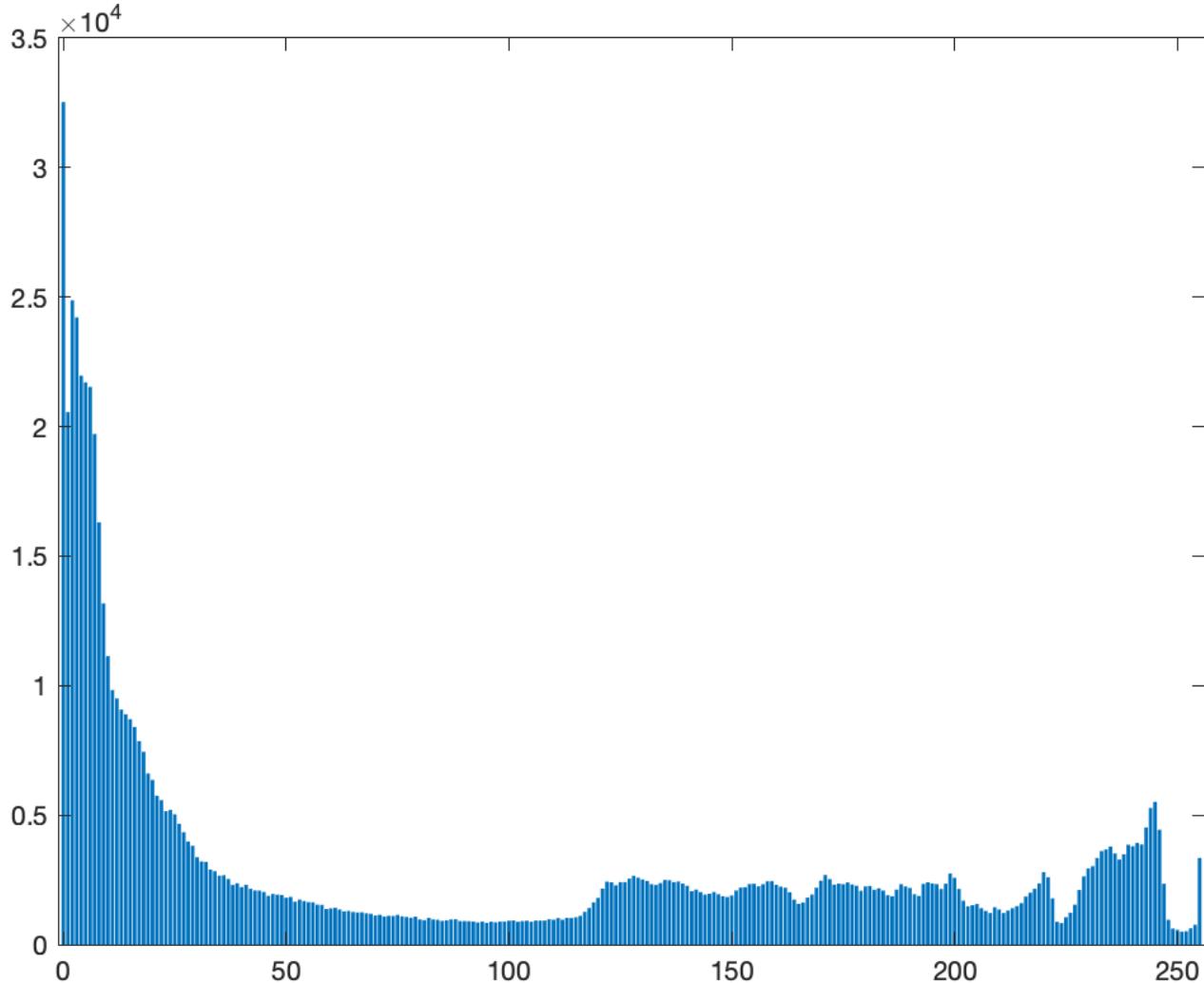


El filtro Gaussiano suaviza la imagen uniformemente.

El filtro bilateral utiliza dos gaussianas para calcular el promedio: en el dominio de la imagen (distancia entre píxeles) y en el rango de la imagen (valores de píxeles), para determinar la cantidad de suavizado que se debe aplicar a un píxel determinado. De este modo, conserva los bordes y los detalles finos, mientras suaviza las regiones más homogéneas de la imagen. Además, el filtro cruzado bilateral utiliza una segunda imagen para determinar regiones y coeficientes a aplicar en el filtro bilateral, aplicándolos finalmente a la primera imagen.

3. Mejora del contraste: ecualización del histograma (35%)

Adjuntad gráfica. ¿Qué valor es el más común en la imagen? ¿Y el más raro? ¿Cuántas veces aparecen cada uno de esos valores en la imagen?

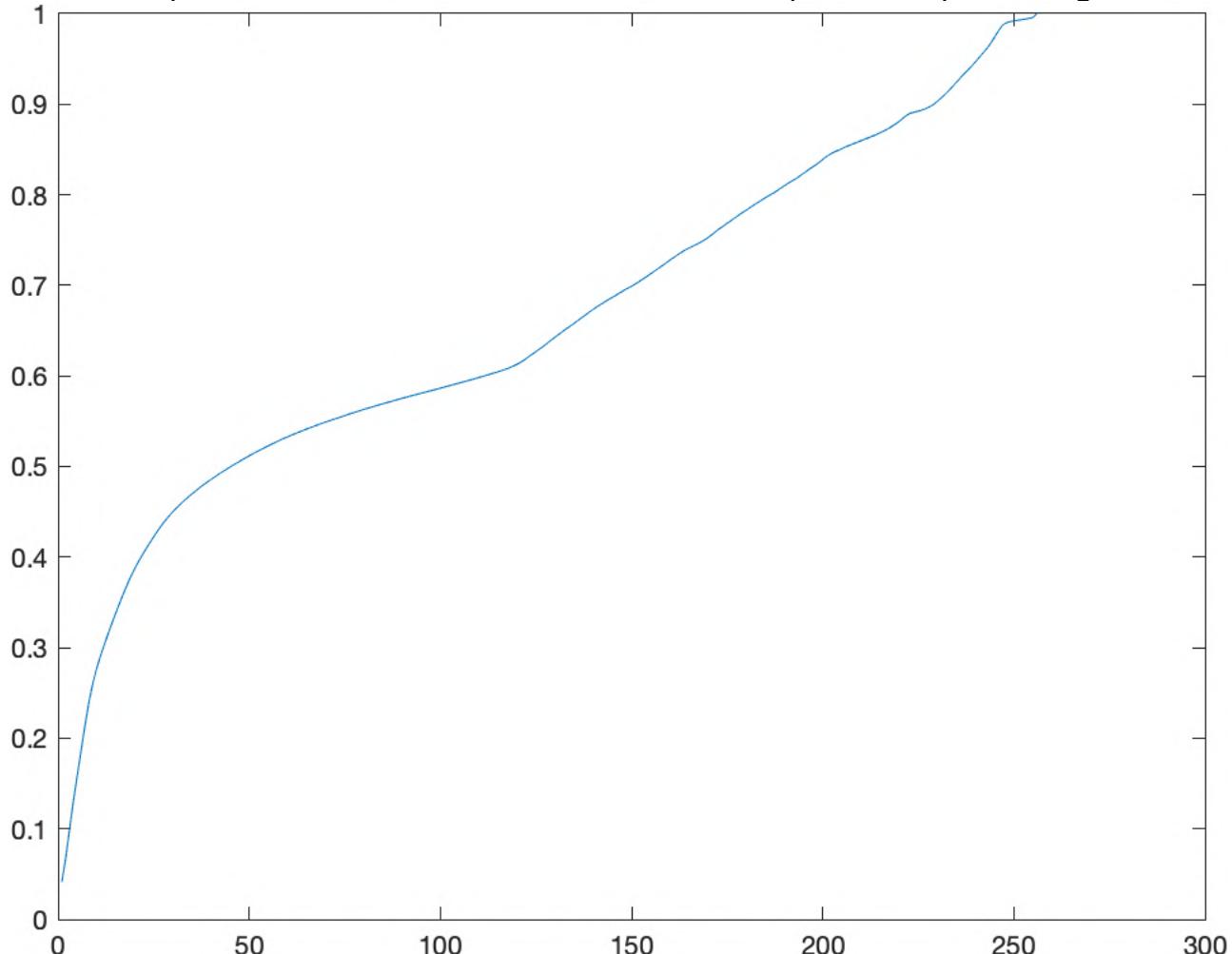


0 es lo valor más común (aparece 32532 veces). 251 és el más raro (aparece 500 veces).

¿Qué probabilidad (en %) tenemos de encontrar un valor de 50 en alguna de las componentes RGB de un píxel?

Probabilidad = 0.2299%

Calculad T y adjuntad su gráfico (plot). Se ve que $T(x)$ es una función siempre creciente que toma valores de 0 a 1. Esto será así para cualquier imagen. ¿Por qué?



Será así porque es la suma cumulativa de la probabilidad de encontrar los valores. La suma de todas las probabilidades de (de cada uno de los posibles valores) será siempre 1. En este caso se muestra que la mayor densidad de píxeles se encuentra entre los primeros valores posibles (esto es antes del valor 50, los colores claros).

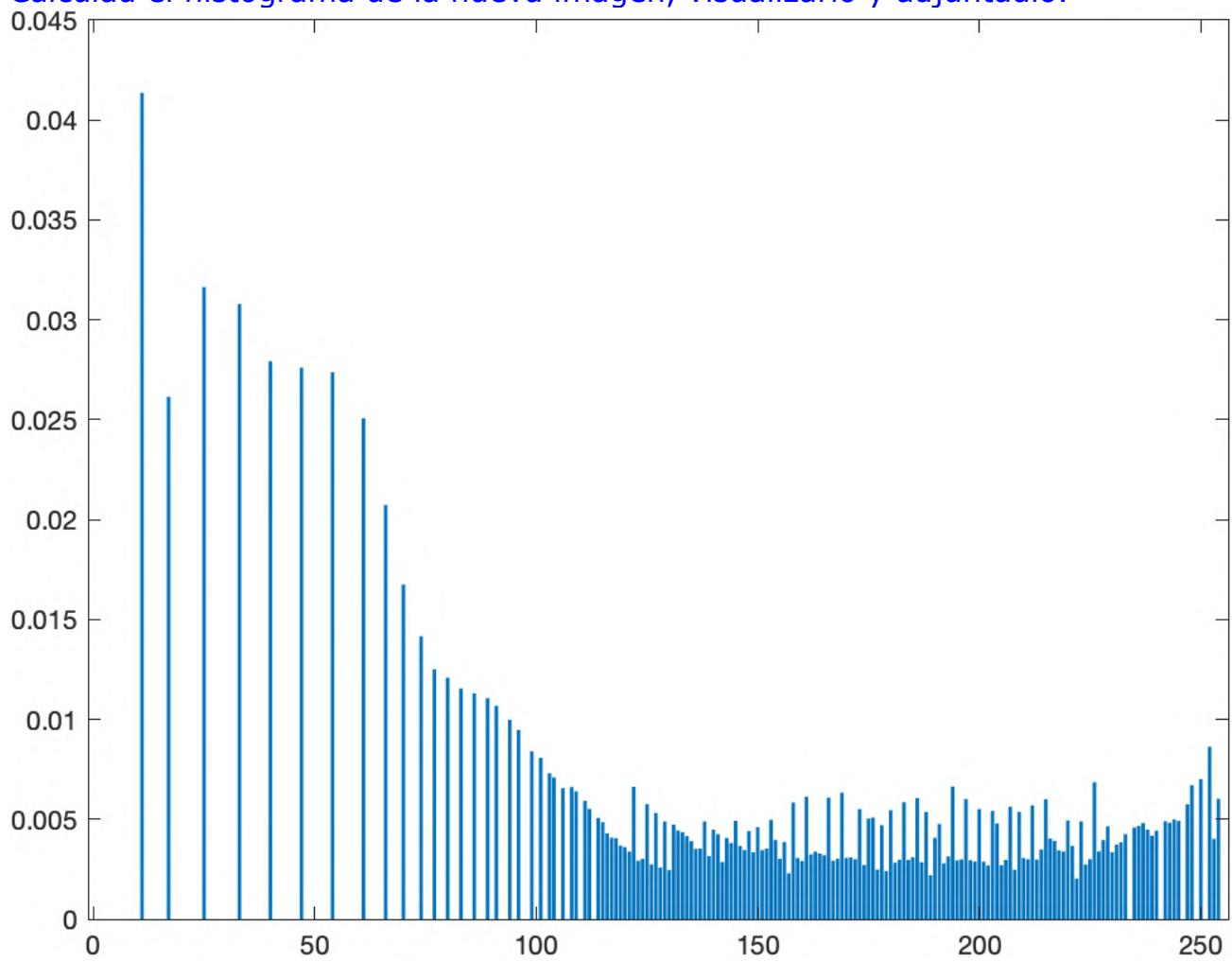
Adjuntad el código usado para obtener la nueva imagen.

```
im2 = T(im + 1) * 255;
```

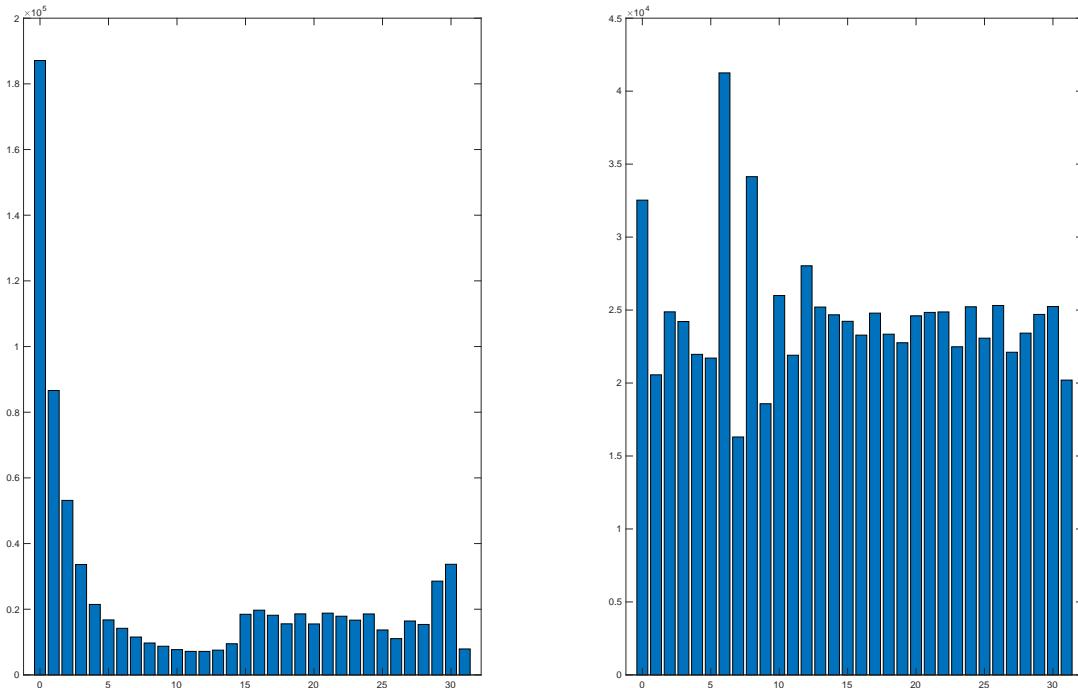
Adjuntad la imagen resultante.



Calculad el histograma de la nueva imagen, visualizarlo y adjuntadlo.



Adjuntad la figura resultante con los dos histogramas.



3b. Mejora local del contraste

En este ejercicio, para el filtrado bilateral, podéis usar la función de MATLAB (mucho más rápida) en vez de vuestra implementación. Ya vimos que daban los mismos resultados si tenemos en cuenta la forma diferente de pasarle los parámetros.

Adjuntad vuestro código de la función f_alfa (se valorará no usar bucles).

```
function alfa=f_alfa(w,G)
    alfa = w;
    alfa(w<=0.5) = 128*(1-(w(w<=0.5)/0.5).^G);
    alfa(w>0.5) = -128*(1-((1-w(w>0.5))/0.5).^G);
end
```

Adjuntad código de function L_alfa(I,alfa)

```
function I_out=L_alfa(I,alfa)
    I_out = I;
    I_out(alfa>0) = log(1+alfa(alfa>0).*I(alfa>0))./log(1+alfa(alfa>0));
    I_out(alfa<0) = 1-log(1+abs(alfa(alfa<0))).*(1-I(alfa<0)))./log(1+abs(alfa(alfa<0)));
end
```

Adjuntad código del proceso completo.

```
im = imread('img2.bmp');
im = im2double(im);
I = rgb2gray(im);

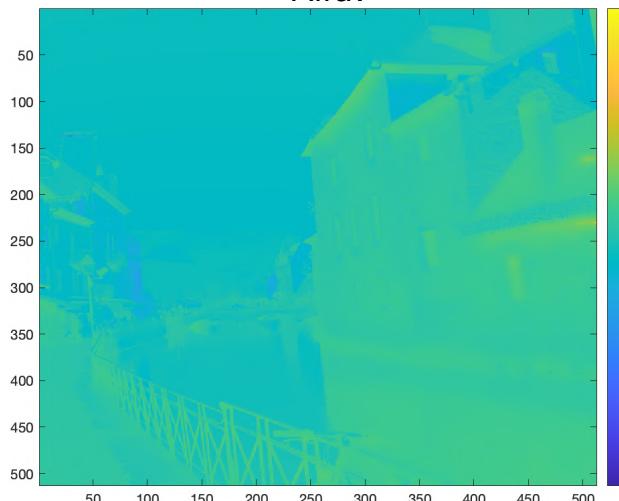
S=5; R=0.05; L=round(2*S);
w=imbilatfilt(I,R^2,S,'Padding','symmetric','NeighborhoodSize',2*L+1);

w = (w-min(w(:)))/(max(w(:))-min(w(:)));

y = 0.05;
alfa = f_alfa(w, y);
I_out = L_alfa(I, alfa);
R = I_out./(I+0.001);
I_res = im.*R;
```

Mostrad los pasos intermedios (alfa y ratio R) como sendas imágenes (usad imagesc). Acompañarlas de una barra vertical al lado mostrando la relación entre colores y valores. Adjuntad ambas imágenes.

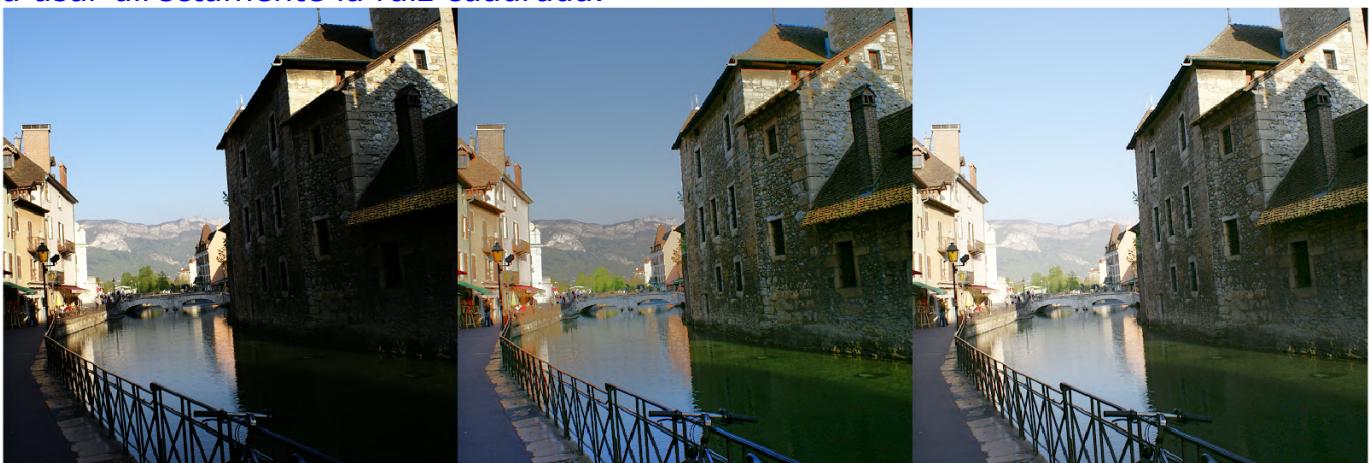
Alfa:



R:



Adjuntad el resultado. Indicad en qué zonas este algoritmo muestra mejoras frente a usar directamente la raíz cuadrada.



Muestra mejoras (frente a usar directamente la raíz cuadrada) en las áreas más claras porque, aunque la raíz cuadrada aclara menos los puntos más claros, todavía los aclara. El algoritmo los oscurecerá si es necesario.

Adjuntad la imagen final. ¿Qué problemas se observan ahora que no aparecían con el filtro bilateral? ¿Cuál es su causa?



El problema que antes no veíamos con el filtro bilateral es que se observa una gran aparición de halos y aberraciones cromáticas, causados por el propio comportamiento del filtro gaussiano que ocasiona distorsión o suavizado en los bordes.

4. Deconvolución de una imagen (20%)

Adjuntad el código del algoritmo.

```
G = im2double(imread('degradado.png'));
imshow(G);

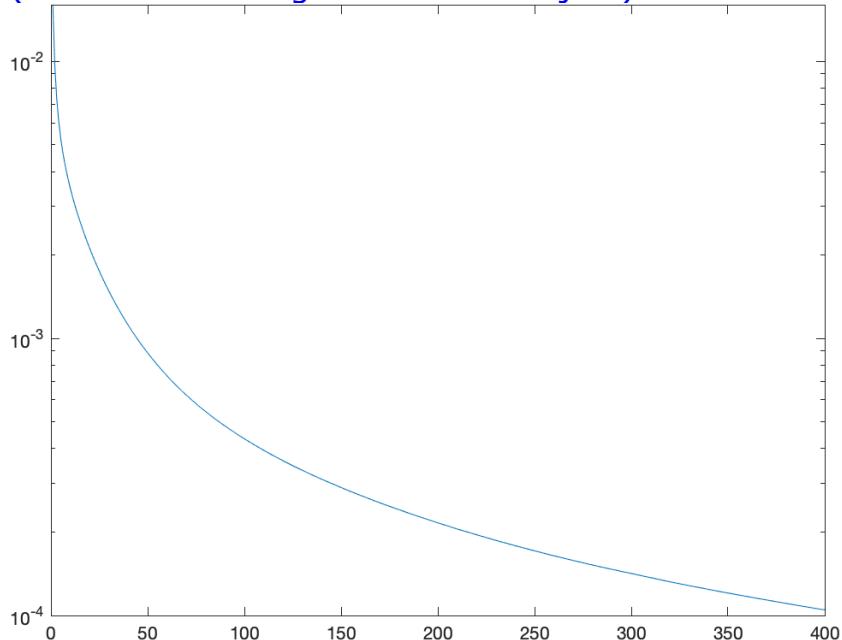
load K;
imagesc(K);colormap(gray)

K2=fliplr(flipud(K));
F = G;

std_dF = zeros(400);

for i=1:400
    G2 = imfilter(F, K, "symmetric");
    Q = G./(G2+0.001);
    Q = imfilter(Q, K2, "symmetric");
    F_old = F;
    F = F_old.*Q;
    F(F>1)=1; F(F<0)=0;
    dF = F-F_old;
    std_dF(i) = std2(dF);
end
```

Tras finalizar el bucle adjuntad la gráfica de la evolución de la desviación de dF (usando escala logarítmica en el eje Y) en función del número de iteración.



Adjuntad la imagen final resultante. ¿Cuál es la última frase del texto?

But we went back down to the valley under our own steam; and to the innkeeper who asked us, with a snicker, how things had gone, and meanwhile was staring at our wild, exalted faces, we answered flippantly that we had had an excellent outing, then paid the bill and departed with dignity. This was it—the bear meat; and now that many years have passed, I regret that I ate so little of it, for nothing has had, even distantly, the taste of that meat, which is the taste of being strong and free, free also to make mistakes and be the master of one's destiny. That is why I am grateful to Sandro for having led me consciously into trouble, on that trip and other undertakings which were only apparently foolish, and I am certain that they helped me later on.

They didn't help Sandro, or not for long. Sandro was Sandro Delmastro, the first man to be killed fighting in the Resistance with the Action Party's Piedmontese Military Command. After a few months of extreme tension, in April of 1944 he was captured by the Fascists, did not surrender, and tried to escape from the Fascist Party house in Cuneo. He was killed with a tommygun burst in the back of the neck by a monstrous child-executioner, one of those wretched murderers of fifteen whom Mussolini's Republic of Salò recruited in the reformatories. His body was abandoned in the road for a long time, because the Fascists had forbidden the population to bury him.

Today I know that it is a hopeless task to try to dress a man in words, make him live again on the printed page, especially a man like Sandro. He was

Última frase del texto: "Today I know that it tis a hopeless task to try to dress a man in words, make him live again on the printed page, especially a man like Sandro." (+ "He was"...).

Adjuntar captura de la imagen final y el gráfico de la desviación σ de dF en función del nº de iteración.

