

Sessió 9

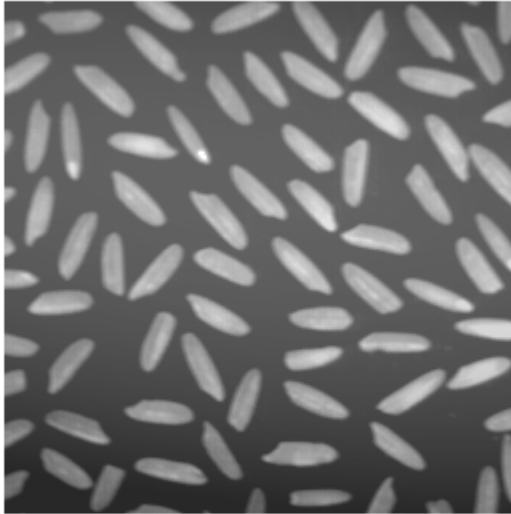
David Agut

Adem Ait

Processat + Segmentació + Etiquetat + Descripció de regions

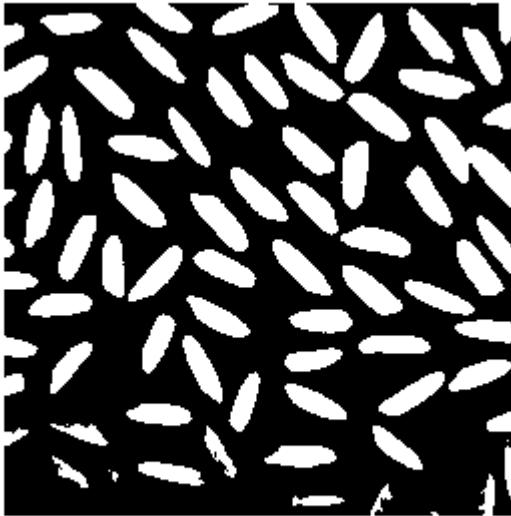
```
clear,close all  
orig = imread ('arros.tif');  
imshow(orig), title('imatge original')
```

imatge original



```
% segmentem per binarització  
bw=im2bw(orig, graythresh(orig)); % Binarització per Otsu  
figure, imshow(bw),title('Threshold automatic sense filtratge')
```

Threshold automatic sense filtratge



No funciona, perdem grans. No confieu mai en els detectors automàtics de llindar.

El problema està en que la il·luminació no és gens homogènea

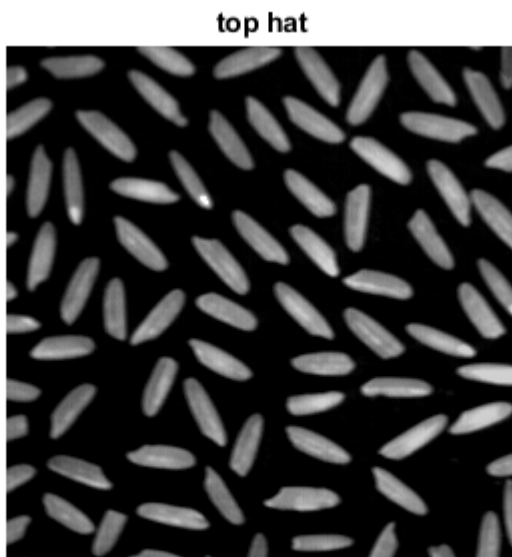
```
% Fem un filtratge morfològic  
bg = imopen(orig,strel('disk',10));  
figure, imshow(bg),title('background')
```

background

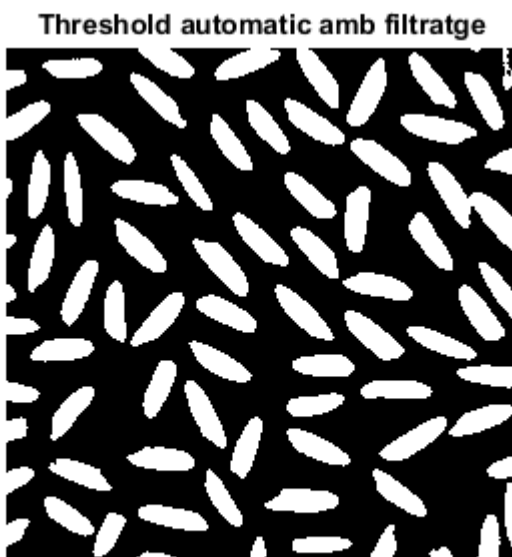


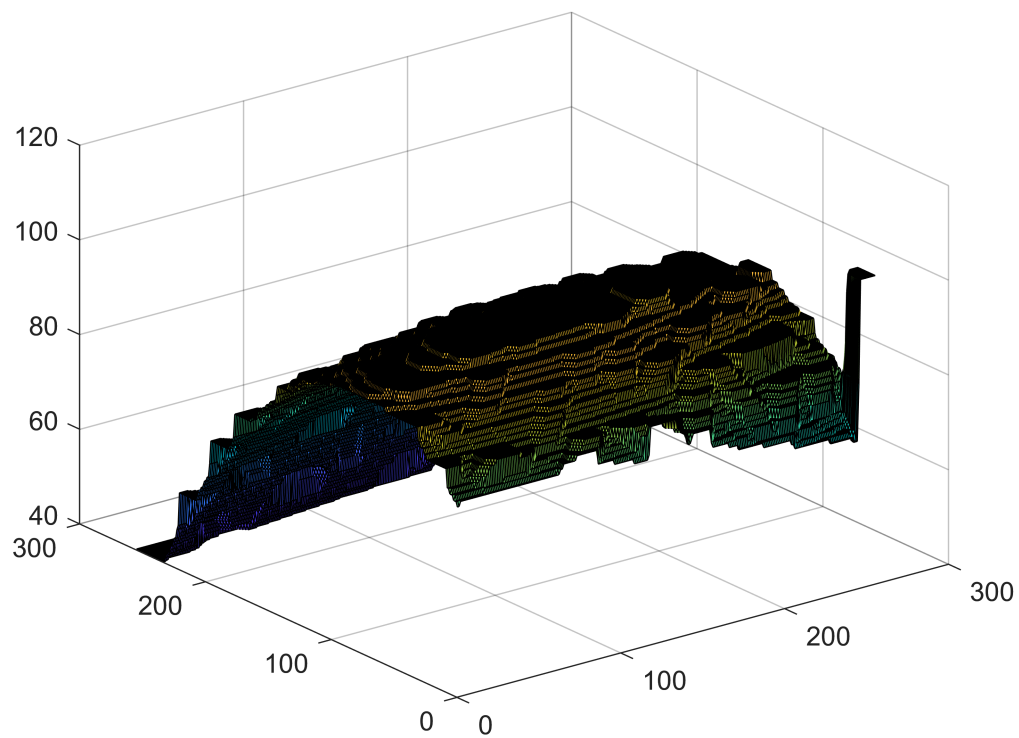
```
figure, surf(bg) % mostrem el background com si fos una superfície  
%Li restem el background a la imatge  
y = imsubtract(orig,bg);
```

```
figure, imshow(y, []), title('top hat')
```



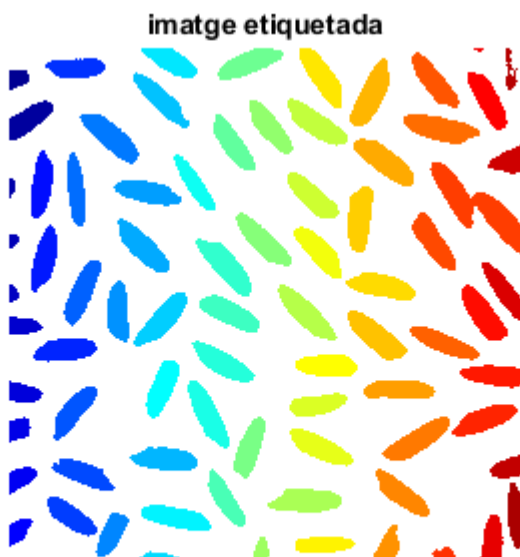
```
% tornem a segmentar per Otsu  
bw=im2bw(y, graythresh(y));  
figure, imshow(bw),title('Threshold automatic amb filtratge')
```





Ara sí!

```
% etiquetem la imatge segmentada
[eti num] = bwlabel(bw,4);
figure,imshow(label2rgb(eti)), title('imatge etiquetada')
```



```
% Comptem ara el nombre d'objectes a la imatge
```

```
num
```

```
num = 81
```

```
%o be:  
max(eti(:))
```

```
ans = 81
```

```
% obtenim descriptors de les regions  
Dades=regionprops(eti, 'all');
```

Comproveu en el workspace la variable Dades.

Quines propietats hem extret de cada regió?

Hem extret moltes propietats, per ser específics totes aquestes: Àrea, centroid, BoundingBox, SubarrayIdx, MajorAxisLength, MinorAxisLength, Eccentricity, Orientation, ConvexHull, ConvexImage, ConvexArea, Circularity, Image, FilledImage, FilledArea, EulerNumber, Extrema, EquivDiameter, Solidity, Extent, PixelIdxList, PixelList, Perimeter, PerimeterOld, MaxFeretDiameter, MaxFeretAngle, MaxFeretCoordinates, MinFeretDiameter, MinFeretAngle, MinFeretCoordinates. Hi han tantes propietats perquè s'ha fet servir la opció all, però si es voldrien només algunes determinades es podria fer servir basic ('Area', 'Centroid', 'BoundingBox').

Amb aquestes dades podem calcular centroids en x i y utilitzant la propietat centroids i la funció cat amb una de les dues components a 1, calcular el centre d'objectes circulars amb Major i MinorAxisLength, entre altres.

<https://es.mathworks.com/help/images/ref/regionprops.html>

Consulteu el help de la funció regionprops.

```
% Podem obtenir una dada de l'objecte 50 de la forma:  
Dades(50).Area
```

```
ans = 305
```

```
% O be un vector amb totes les Àrees:  
Arees=[Dades.Area];
```

Fem la suma de totes les arees per comprovar l'àrea total

```
Suma=sum(Arees, "all")
```

```
Suma = 19778
```

Exercici

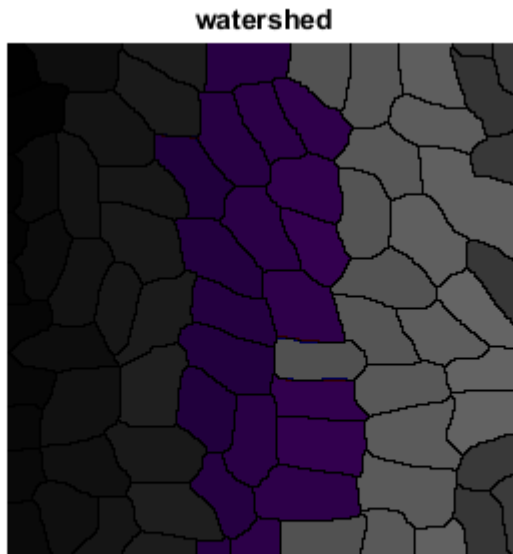
Exercici: en la imatge segmentada, separeu els grans d'arròs que es toquen. Després elimineu els grans d'arròs que toquen les vores. Tots aquests grans falsejaven el resultat.

Amb la nova imatge etiquetada, obteniu les propietats que us semblin adients usant regionprops.

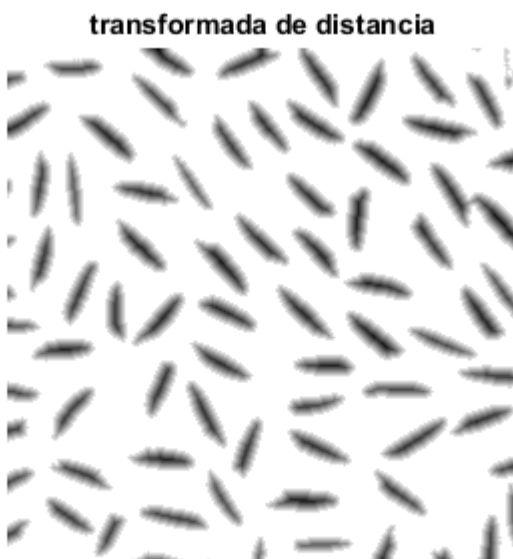
Representeu mitjançant plots o diagrames de barres aquelles propietats que us semblin interessants per a fer un control de qualitat dels grans d'arròs.

-Per fer la solució ens hem ajudat del següent link de mathworks, on explica com fer la segmentació per watershed amb la transformada de la distància (<https://blogs.mathworks.com/steve/2013/11/19/watershed-transform-question-from-tech-support/>)

```
L = watershed(label2rgb(eti));  
figure,imshow(L),title("watershed")
```

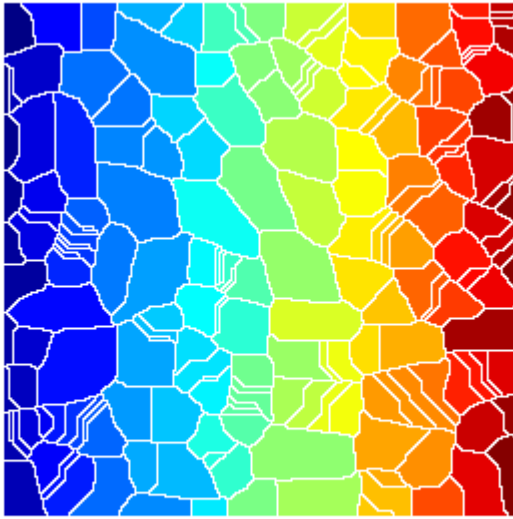


```
D = -bwdist(~bw);  
figure,imshow(D,[]),title("transformada de distancia")
```



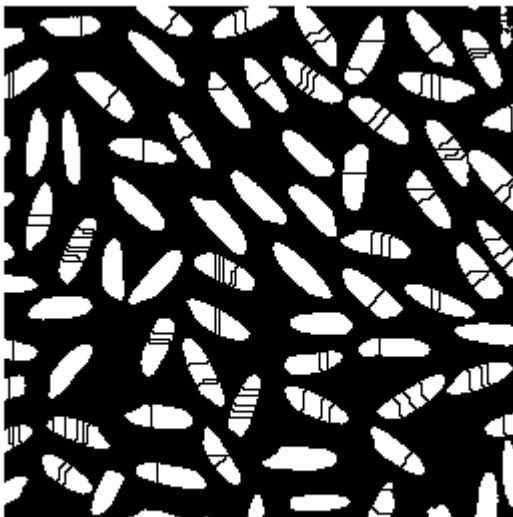
```
Ld = watershed(D);
figure,imshow(label2rgb(Ld)),title("watershed de la transformada de distancia")
```

watershed de la transformada de distancia



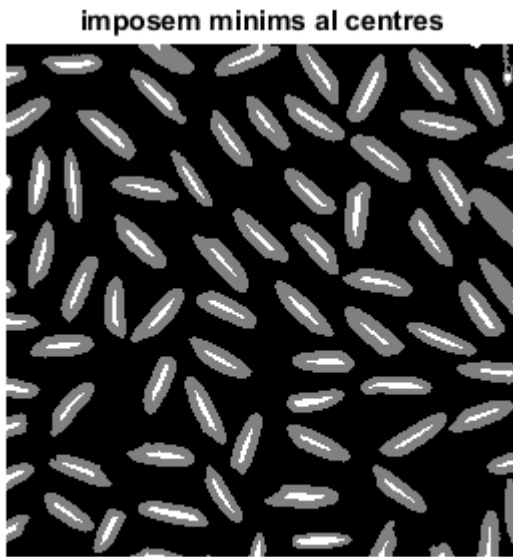
```
bw2 = bw;
bw2(Ld == 0) = 0;
figure,imshow(bw2),title("sobressegmentacio")
```

sobressegmentacio

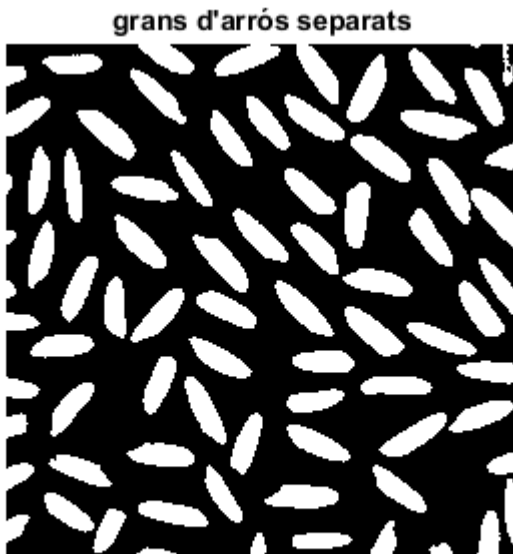


```
mask = imextendedmin(D,1);
```

```
figure,imshowpair(bw,mask,'blend'),title("imposem minims al centres")
```



```
D2 = imimposemin(D,mask);  
Ld2 = watershed(D2);  
bw3 = bw;  
bw3(Ld2 == 0) = 0;  
figure,imshow(bw3),title("grans d'arrós separats")
```



```
mark(1:256,1:256)=true;
```



```
mark(2:end-1,2:end-1)=false;  
rec=imreconstruct(mark,bw3);  
figure,imshow(rec),title('reconstruccio de blancs per eliminar els grans dels bordes')
```

reconstruccio de blancs per eliminar els grans dels bordes



```
res=imsubtract(bw3,rec);  
figure,imshow(res),title("grans d'arros que limiten amb la vora eliminats")
```

grans d'arros que limiten amb la vora eliminats



```
[eti num] = bwlabel(res,4);
```

```
figure,imshow(label2rgb(eti)), title('imatge etiquetada final')
```



El nombre de grans d'arrós ara són

```
num
```

```
num = 54
```

```
DadesSol=regionprops(eti, 'all');
```

```
% Podem obtenir una dada de l'objecte 50 de la forma:  
DadesSol(50).Area
```

```
ans = 299
```

```
% O be un vector amb totes les Àrees:  
AreesSol=[DadesSol.Area];
```

```
SumaSol=sum(AreesSol, 'all')
```

```
SumaSol = 15947
```

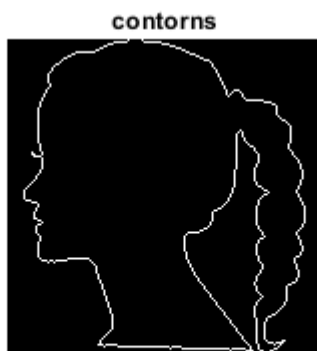
Podem apreciar que ara tenim menys grànols d'arrós i menys àrea total, degut a que hem eliminat tots els que tocaven amb la vora.

Codis de cadena

```
clear all, close all  
im=imread('head.png');  
% no cal tan gran  
im=imresize(im,1/4);  
imshow(im), title('imatge original')
```



```
% obtenim el contorn
ero=imerode(im,strel('disk',1));
cont=xor(ero,im);
figure,imshow(cont), title('contorns')
```



```
% obtenim les coordenades del contorn
[fila col] = find(im,1); % Busquem el primer píxel
B = bwtraceboundary(im,[fila col],'E'); %direccio est a l'atzar
% B conté les coordenades
% Ho comprovem mostrant el resultat
aux=zeros(size(im));
aux(sub2ind(size(aux),B(:,1),B(:,2)))=1;
figure,imshow(aux),title ('contorns a partir de coordenades')
```

contorns a partir de coordenades



Exercici

Trobar els codis de cadena incrementals a partir de B

```
% Per trobar el codi incremental si restem per cada fila el seu posterior
% trobarem la variació
[fil col]=size(B);
var=B(2:fil,:) - B(1:fil-1,:);
codCad=zeros(fil-1,1);
%Pel codi incremental suposarem aquesta distribucio
% 765
% 8 4
% 123
for j = 1:fil-1
    if (var(j,1) == -1 && var(j,2) == -1)
        codCad(j) = 1;
    end
    if (var(j,1) == 0 && var(j,2) == -1)
        codCad(j) = 2;
    end
    if (var(j,1) == 1 && var(j,2) == -1)
        codCad(j) = 3;
    end
    if (var(j,1) == 1 && var(j,2) == 0)
        codCad(j) = 4;
    end
    if (var(j,1) == 1 && var(j,2) == 1)
        codCad(j) = 5;
    end
    if (var(j,1) == 0 && var(j,2) == 1)
        codCad(j) = 6;
    end
    if (var(j,1) == -1 && var(j,2) == 1)
        codCad(j) = 7;
    end
    if (var(j,1) == -1 && var(j,2) == 0)
        codCad(j) = 8;
    end
end
```

```
end
```

Una forma més eficient de fer aquest càlcul es podria fer amb la funció `atan2d`, la qual calcula l'arctangent amb coordenades cartesianes (<https://www.mathworks.com/help/matlab/ref/atan2d.html>)

```
%Obtenim l'angle de cada fila, en un rang de [-180 180]
codCad2=atan2d(var(:,2),var(:,1));
%Sumem 360 per tenir els angles en [180 540] i així no tenir negatius
codCad2=codCad2+360;
%Dividim entre 45, ja que tenim 8 formes de moviment, en angles de 45º
codCad2=codCad2/45;
% D'aquesta manera tenim els valors entre 4 i 12, rectifiquem a valors
% entre 1 i 8
codCad2=codCad2-4;
%Així tindriem la mateixa distribucio
```

I de forma compacte seria la següent

```
%Per fer-ho de forma més directa es podria calcular també així
codCad3=(atan2d(var(:,2),var(:,1))+360)/45-4;
```

Descriptors de Fourier

```
% centrem coordenades
mig=mean(B);
B(:,1)=B(:,1)-mig(1);
B(:,2)=B(:,2)-mig(2);

% Convertim les coordenades a complexes
s= B(:,1) + i*B(:,2);

% Cal que la dimensio del vector sigui parell
[mida bobo]=size(B);
if(mida/2~=round(mida/2))
    s(end+1,:)=s(end,:); %duplicuem l'ultim
    mida=mida+1;
end

% Calculem la Fast Fourier Transform
z=fft(s);
% representem l'espectre
figure,plot(abs(z))
% ho displaïem logarítmic perquè no es veu res
figure,plot(log(abs(z))), title ('descriptors de Fourier')
```

Observeu que l'espectre surt duplicat, amb un efecte mirall. És una propietat de la transformada de Fourier, que ara no ve al cas.

```
% Recuperem la imatge original per comprovar que el procés és reversible
ss=ifft(z); % Transformada de Fourier inversa
files= round(real(ss)+mig(1));
cols=round(imag(ss)+mig(2));
```

```

aux(:,:)=0;
aux(sub2ind(size(aux),files,cols))=1;
figure,imshow(aux), title('imatge recuperada')

```



```

%% Reduim la quantitat de descriptors de Fourier
N=30; % agafem N descriptors
tmp=z;
tmp(N+1:end-N)=0; % eliminem els del mig porque es duplica l'espectre
figure, plot(log(abs(tmp))), title ('Només 30 descriptors')

```

```

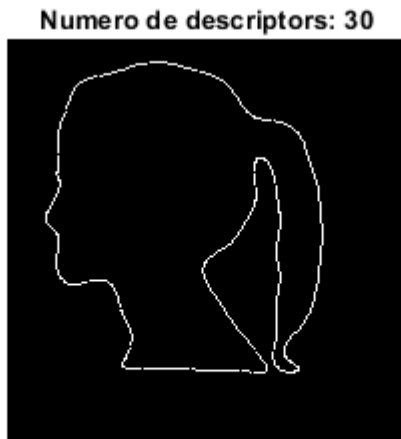
% Tornem al pla imatge a partir de l'espectre modificat
ss2=ifft(tmp);
% Les coordenades resultants poden sortir del rang de la imatge original
% Creo una imatge més gran per a que les coordenades no s'em surtin de mare
mida=200;
files= round(real(ss2)+mida/2);
cols=round(imag(ss2)+mida/2);
aux=logical(zeros(mida));
aux(sub2ind(size(aux),files,cols))=1;
figure,imshow(aux)

```

```

title(['Numero de descriptors: ',num2str(N)]);

```

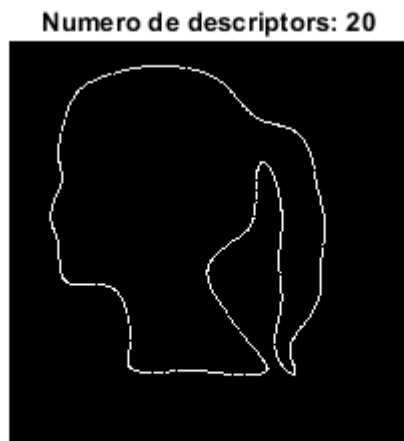


Exercici

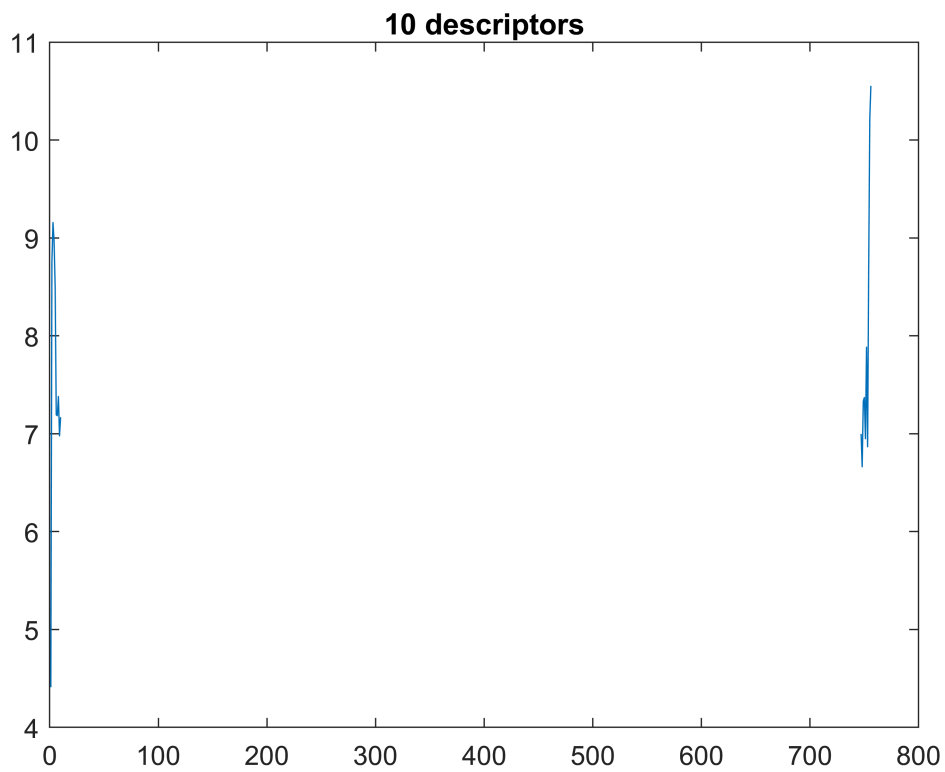
Repetiu el procés utilitzant diferents quantitats de descriptors i expliqueu quin efecte té això en el detall de la imatge obtinguda

```
N=20;
tmp=z;
tmp(N+1:end-N)=0;
figure, plot(log(abs(tmp))), title ('20 descriptors')

ss2=ifft(tmp);
mida=200;
files= round(real(ss2)+mida/2);
cols=round(imag(ss2)+mida/2);
aux=logical(zeros(mida));
aux(sub2ind(size(aux),files,cols))=1;
figure,imshow(aux)
title(['Numero de descriptors: ',num2str(N)]);
```



```
N=10;  
tmp=z;  
tmp(N+1:end-N)=0;  
figure, plot(log(abs(tmp))), title ('10 descriptors')
```



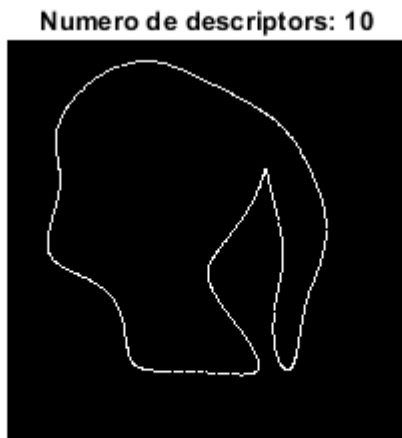
```
ss2=ifft(tmp);  
mida=200;
```



```

files= round(real(ss2)+mida/2);
cols=round(imag(ss2)+mida/2);
aux=logical(zeros(mida));
aux(sub2ind(size(aux),files,cols))=1;
figure,imshow(aux)
title(['Numero de descriptors: ',num2str(N)]);

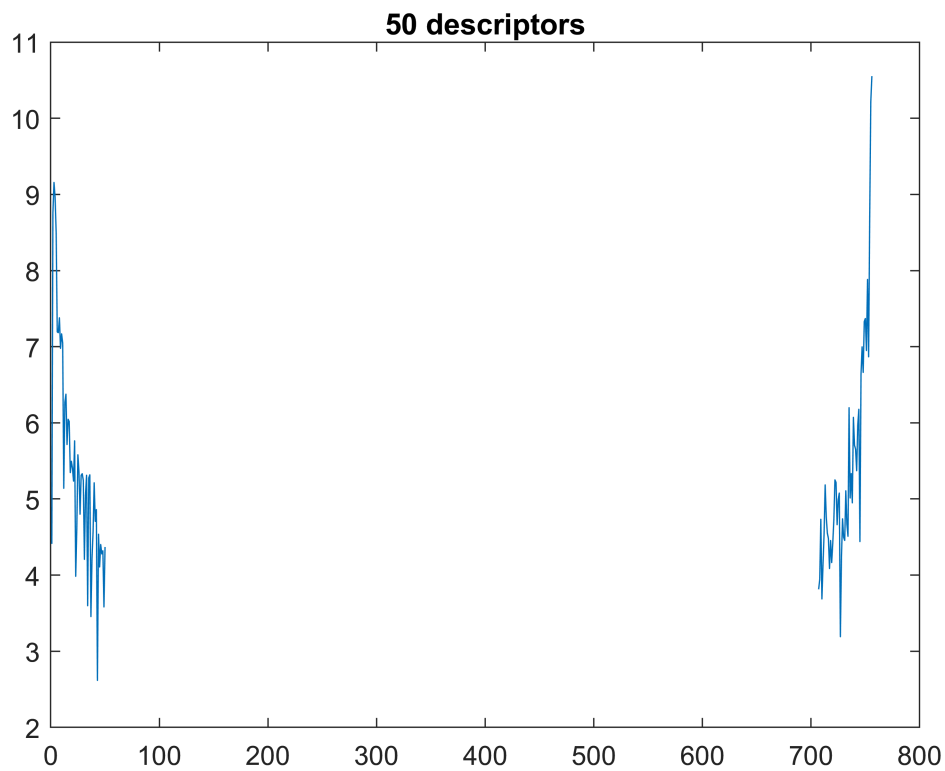
```



```

N=50;
tmp=z;
tmp(N+1:end-N)=0;
figure, plot(log(abs(tmp))), title ('50 descriptors')

```



```
ss2=ifft(tmp);  
mida=200;  
files= round(real(ss2)+mida/2);  
cols=round(imag(ss2)+mida/2);  
aux=logical(zeros(mida));  
aux(sub2ind(size(aux),files,cols))=1;  
figure,imshow(aux)
```

```
title(['Numero de descriptors: ',num2str(N)]);
```

Numero de descriptors: 50



El numero de descriptors que agafem té un efecte en la qualitat i el detall de la imatge obtinguda. Si agafem pocs descriptors, només estem agafant els de més freqüència, per tant l'imatge resultant no serà molt detallada, simplement ens donarà una idea general de la forma bàsica que tenia la figura. A mesura que augmentem el nombre de descriptors agafats, la imatge va guanyant més detall i precisió.