

Sessió 6 - 21/03

Processat morfològic d'imatges II

Ultimate erode: si fem moltes erosions, la imatge acaba sent negra. Ultimate es refereix a la última erosió on es vegi el últim pixel (centre morfològic).

$BW = \text{erode}(BW) \text{ or } \text{hit\&miss}(BW, \dots)$

Envolvent convexa: és el mínim envolcall que no tingui concavitat. Es fa amb un procés iteratiu en el que fem hit&miss. Semblant a l'esquelet però omplint forats.

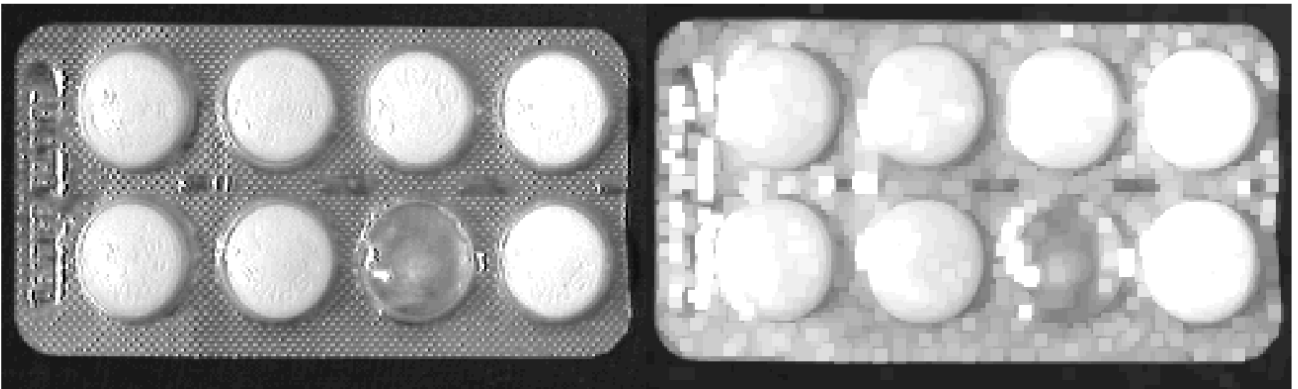
Si a una imatge li faig molts opens -> És com fer-ho una vegada. Torna a quedar igual.

Fer molts dilates -> tot blanc

Morfologia per a imatges multinivell

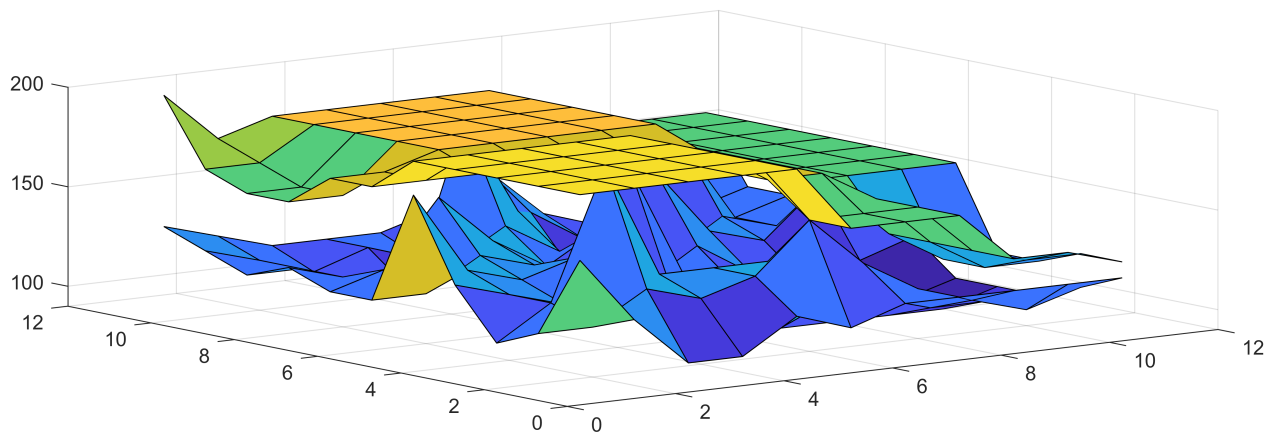
Fer el mateix per a tots els nivells de grisos.

```
I = imread('astablet.tif');  
%dilatació  
SE = ones(5,5);  
ID = imdilate(I,SE);  
montage({I,ID});
```

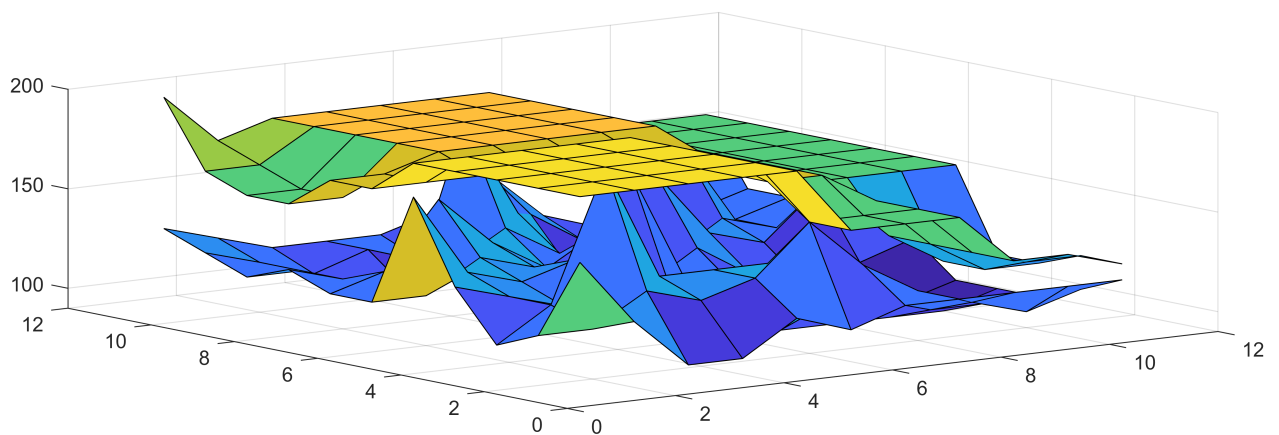


```
% mirem què ha passat  
[f,c] = size(I);  
ICrop = I(floor(f/2)-5:floor(f/2)+5, floor(c/2)-5:floor(c/2)+5);  
IDCrop = ID(floor(f/2)-5:floor(f/2)+5, floor(c/2)-5:floor(c/2)+5);  
  
surf(ICrop);  
hold on
```

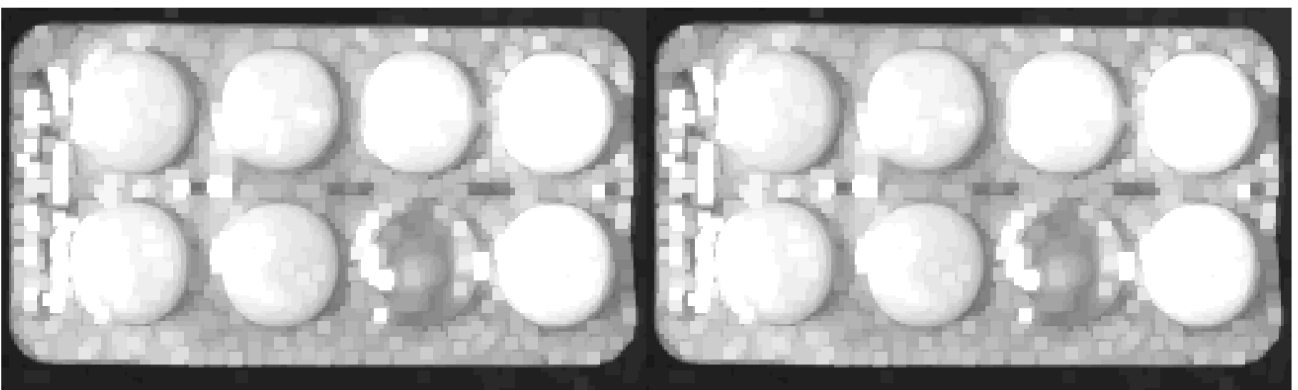
```
surf(IDCrop); % veiem que fa el màxim dels seus veïns
```



```
% dilate = ? max dels seus veïns  
hold off
```



```
ID2 = colfilt(I,[5,5],"sliding",@max);  
montage({ID,ID2});
```

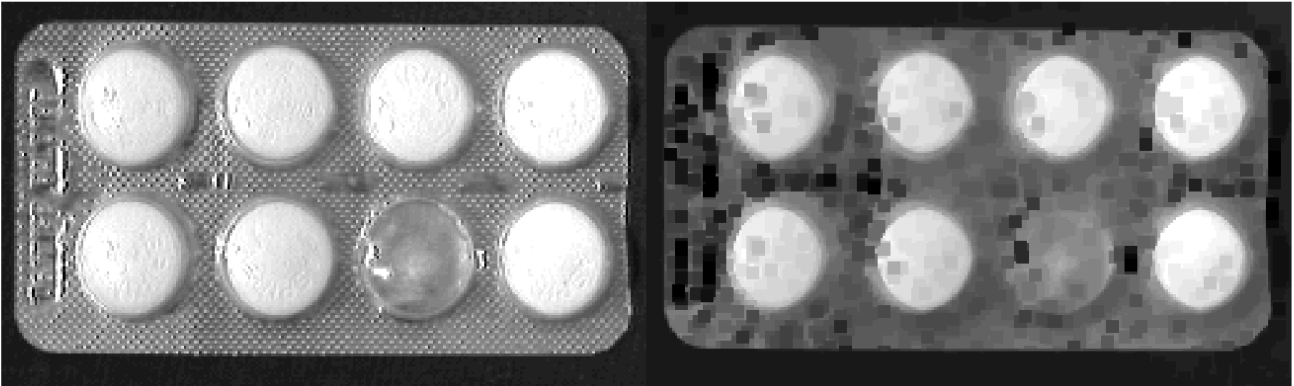


```
% erode
```

```
SE = ones(5,5);
```

```
IE = imerode(I,SE);
```

```
montage({I,IE}); % fa el minim dels seus veïns
```



```
% open
```

```
SE = ones(3,3); % més petit pq els piquets son molt petits
```

```
IO = imopen(I,SE);
```

```
montage({I,IO});
```



```
% close
```



```
SE = ones(3,3); % més petit pq els piquets son molt petits
IC = imclose(I,SE);
montage({I,IC});
```

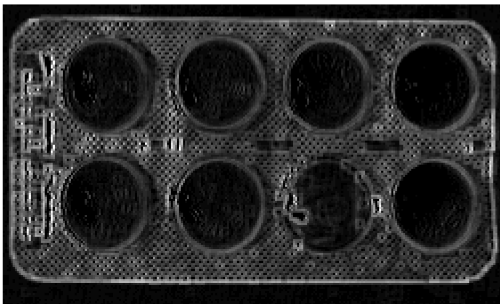
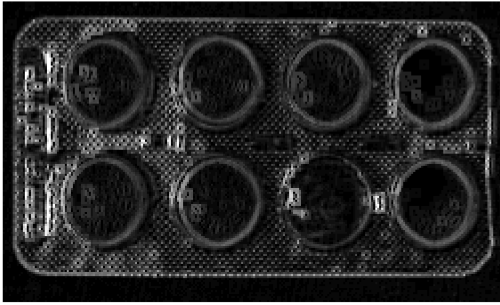
```
% filtre open close
SE = ones(3,3);
IOC = imclose(IO,SE);
montage({I,IOC}); % així treiem els punt blancs i negres dels piquets
```



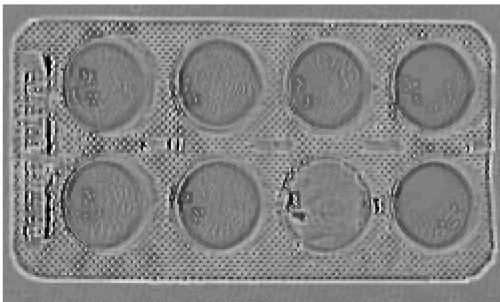
```
% residus
RI = I - IE; % mai serà negatiu pq la imatge erosionada mai serà més gran que la original
imshow(RI,[]);

RE = ID - I; % aquí igual pq agafem el màxim
imshow(RE,[]);

L = double(RE) - double(RI);
```

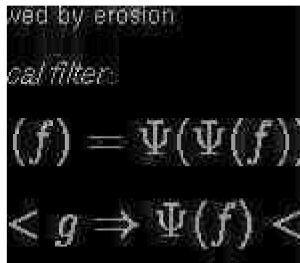


```
imshow(L,[]); %laplacia: passos per 0
```

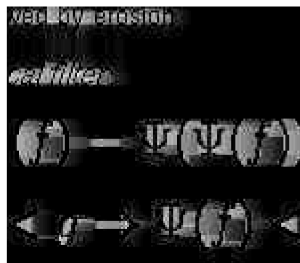


```
% residus II
I = imread('nshadow.tif');
SE = ones(10,10);
IO = imopen(I,SE);
IC = imclose(I,SE);
```

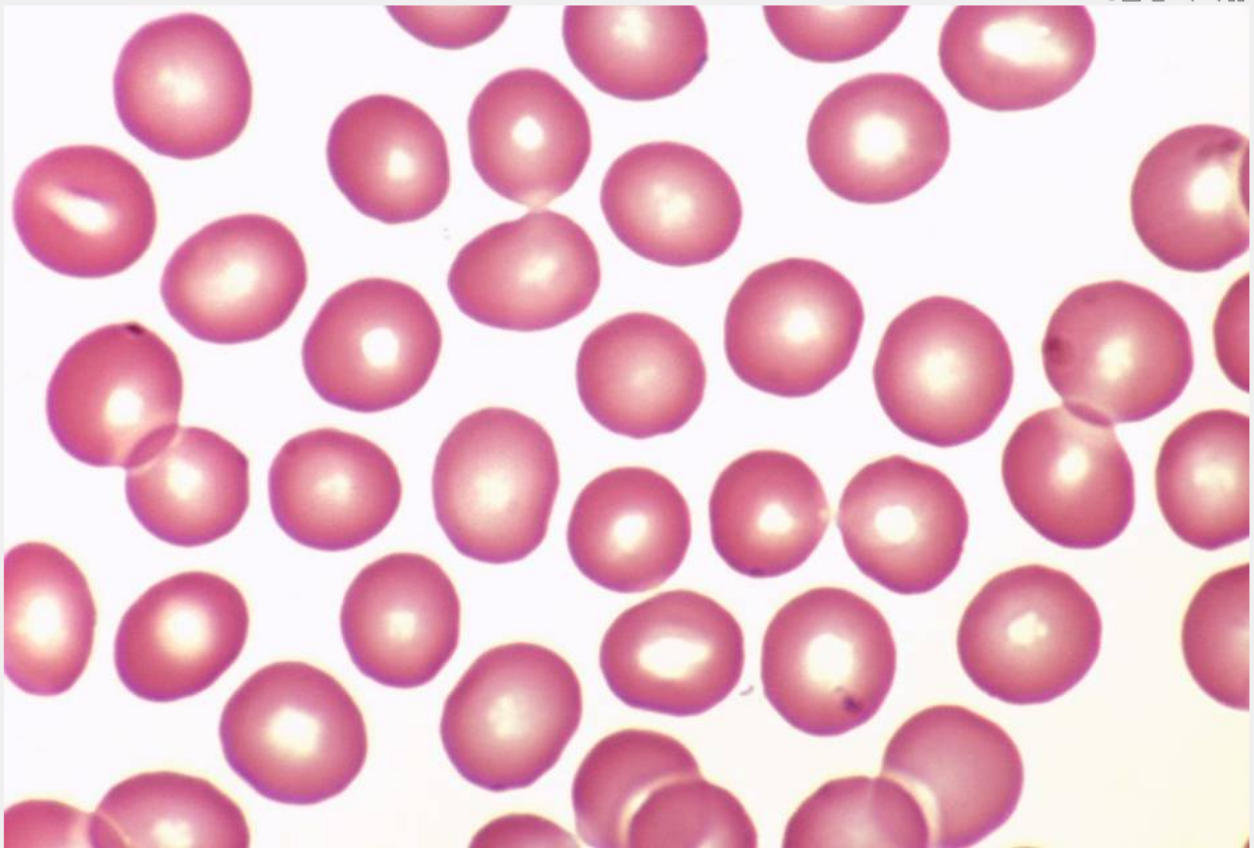
```
% top-hat
% dona la part blanca de la imatge
TH = I - IO;
imshow(TH,[]); % li em tret el bg que era uniforme. Si volguessim acabar de filtrar, es podria
```



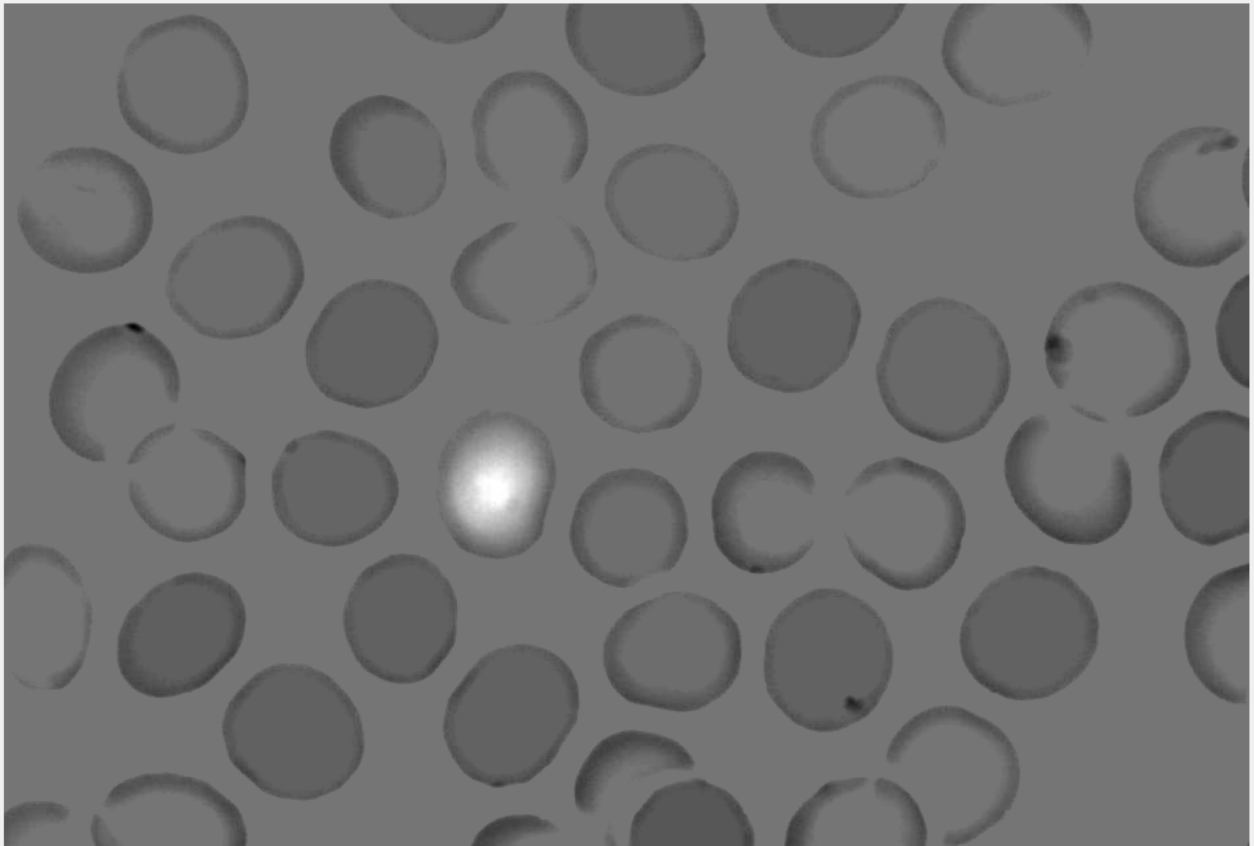
```
% bottom-hat
% dona la part negra de la imatge (no ens serveix de res aqui)
BH = IC - I;
imshow(BH,[]);
```



```
% reconstruccio multinivell: dilatació a partir d'un punt
RGB = imread("normal-blood1.jpg");
I = rgb2gray(RGB);
imshow(RGB);
[x y] = getpts;
```



```
[f,c] = size(I);  
MARK = uint8(zeros([f,c]));  
MARK(uint16(y),uint16(x)) = 255;  
REC = imreconstruct(MARK,I);  
imshow(REC,[]);
```

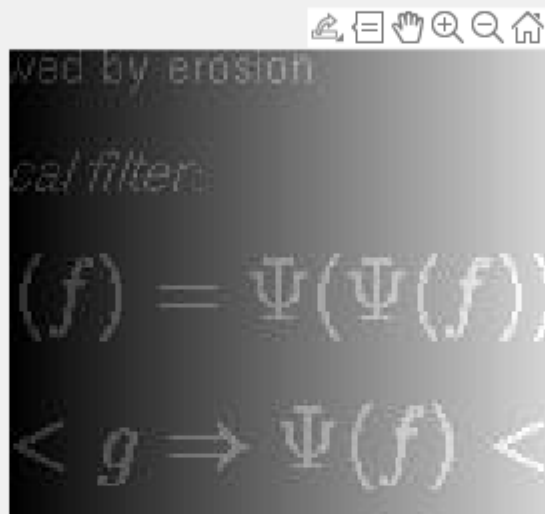


```
% reconstruccio multinivell: dilatació a partir d'un punt. Cas 2
```

```
I = imread("nshadow.tif");  
imshow(I);
```

```
SE = ones(10,10);  
IO = imopen(I,SE);  
TH = I - IO;
```

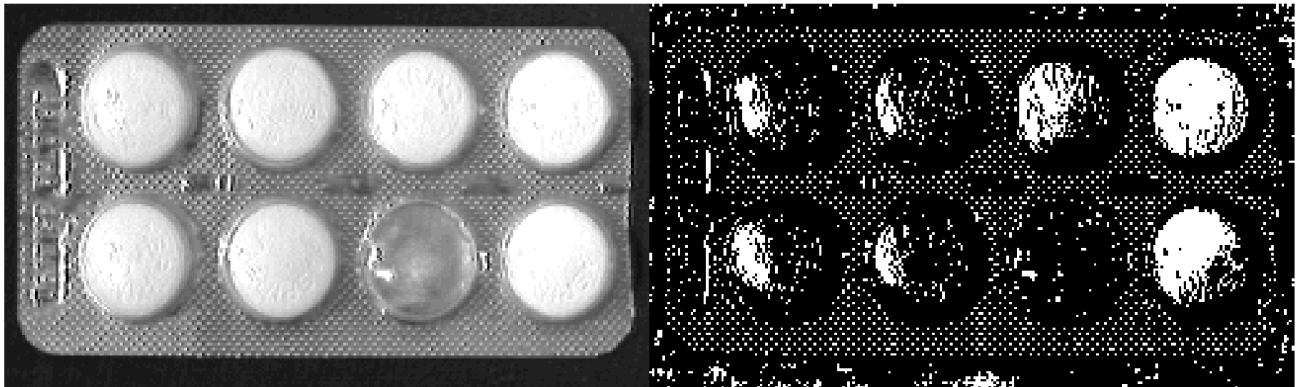
```
[x y] = getpts;
```

```
[f,c] = size(I);  
MARK = uint8(zeros([f,c]));  
MARK(uint16(y),uint16(x)) = 255;  
REC = imreconstruct(MARK,TH);  
imshow(REC,[]);
```

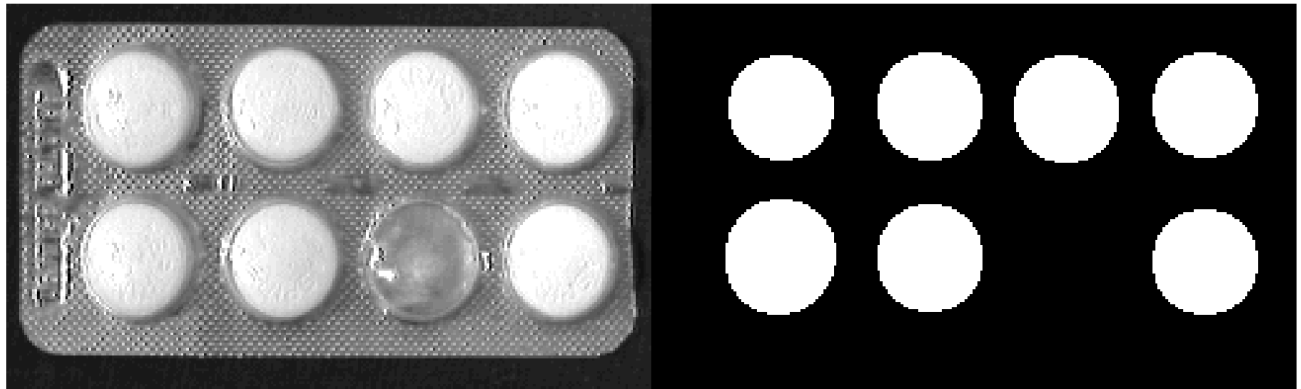


```
% regional max
I = imread("astablet.tif");
MR = imregionalmax(I); % masses maxims regionals
montage({I,MR});
```



```
%[f,c] = size(I);

ES = fspecial("disk",20) > 0; % 20-> ajustat a la mida que busquem. el > 0, es pq ha de ser binari
IO = imopen(I,ES);
MRO = imregionalmax(IO); % hem detectat on no hi ha pastilla
montage({I,MRO}); % ha fet un open tan gran que se l'ha menjat.
```



```
% treure lo que no sigui pastilla
```

```
IM = I;
```

```
IM(not(MRO)) = 0;
```

```
REC = imreconstruct(IM,I);
```

```
imshow(REC,[]);
```



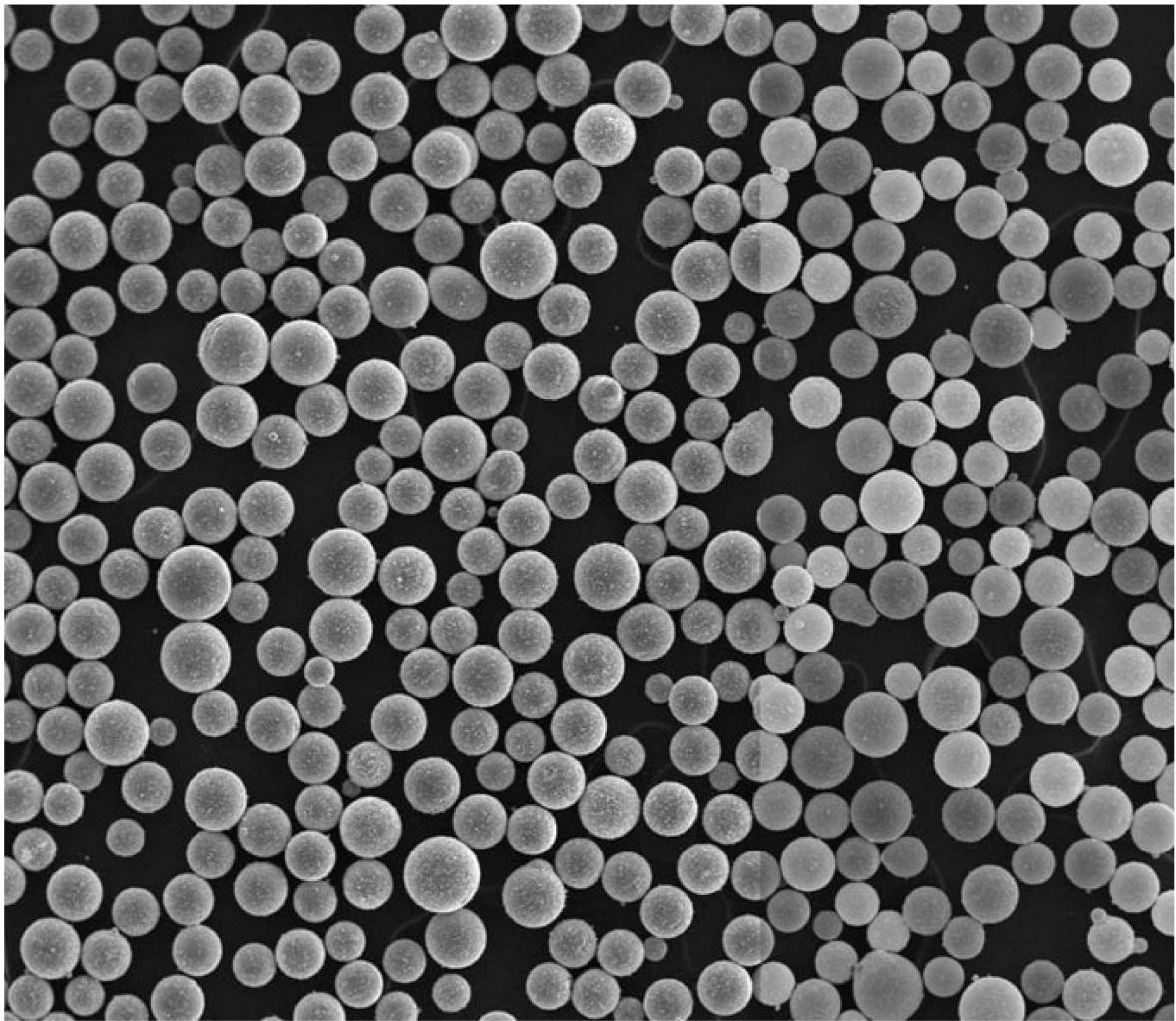
```
% ex realitzar un plot granulomètric
```

```
% plt size vs elements
```

```
RGB = imread("granulometria.png");
```

```
I = rgb2gray(RGB);
```

```
imshow(I);
```



```
for i=1:30
    SE = fspecial("disk",i) > 0; % ajustat a diferents mides. de petit a gran
    IO = imopen(I,SE);
    % calculem el llindar per binaritzar la imatge
    t = otsuthresh(imhist(IO));
    t = max(64,t*255);
    BW = IO > t;
    if sum(sum(BW)) > 0
        BWUE = bwulterode(BW);
        imshow(BWUE);
        CC = bwconncomp(BWUE);
        A(i) = CC.NumObjects;
    else
        A(i) = 0;
    end
end
```



```
plot(A);
```

