

## Sessió 5 - 14/03

### Processat Morfològic d'Imatges

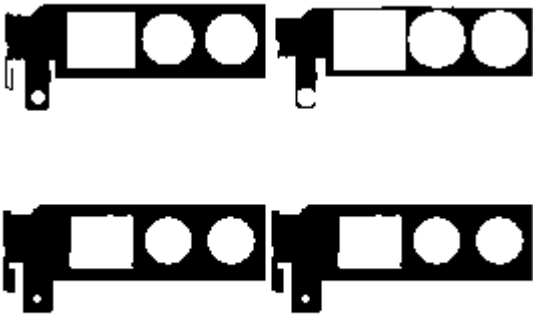
**Dilatació:** agafar el element estructurant i el passem pels px a 1 de la imatge original, i fem una OR. Així aconseguim la imatge dilatada.

**Erosió:** com la dilatació però tot el contrari (fent servir una AND)

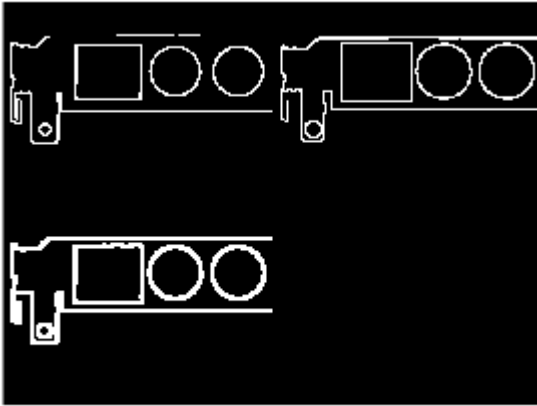
**Translació:** desplaçar.

```
% dilatació
I = imread('Bracket1.tif');
BW = I > 128;
SE = ones(11,11);
BWD = imdilate(BW,SE); % SE: element estructurant
% erosió
BWE = imerode(BW,SE);

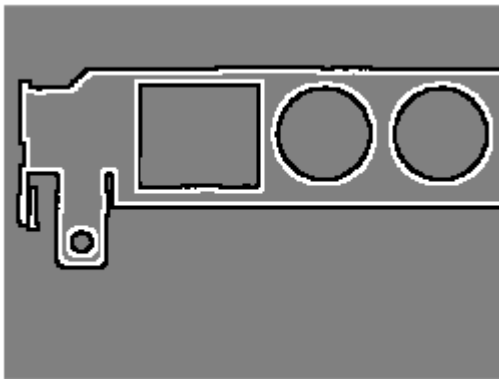
% test: erosio es una dilatació del background
BWT = imdilate(not(BW),SE);
montage({BW,BWD,BWE,not(BWT)}); % veiem com BWE i BWT son iguals
```



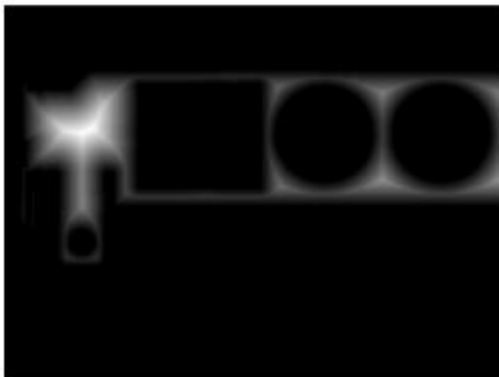
```
% residus
RI = BW & not(BWE); % residu intern = img original - erosionada
RE = BWD & not(BW); % residu exterior = img original - dilatada
RD = BWD & not(BWE); % (dual)
montage({RI,RE,RD});
```



```
% laplacia morfologic
L = double(RE) - double(RI);
imshow(L,[]);
```



```
% transformada de la distancia
TD = bwdist(BW,'euclidean');
imshow(TD,[]);
```



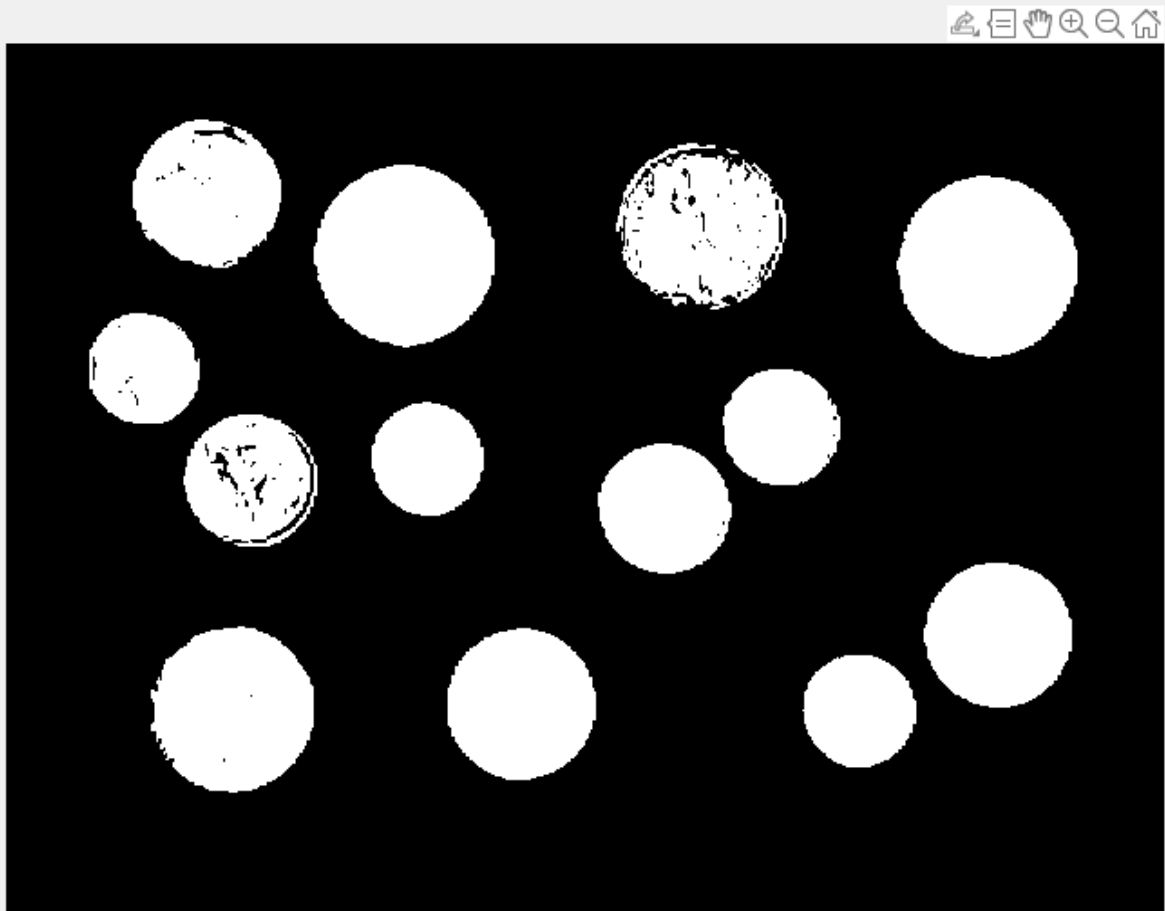
```
% Reconstrucció
```

```
I = imread('Money.tif');
```

```
BW = I > 128;
```

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

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



```
MARK = false(size(BW));
```

```
% uint16: evitar agafar decimals en el problema del getpts
```

```
for i = 1:size(x)
```

```
    MARK(uint16(y(i)),uint16(x(i))) = 1;
```

```
end
```

```
REC = imreconstruct(MARK,BW);
```

```
imshow(uint8(REC).*I,[]);
```



% Exercici eliminar objectes de les vores

```
I = imread("arros.tif");
```

```
BW = I > 128;
```

```
SE = ones(7,7);
```

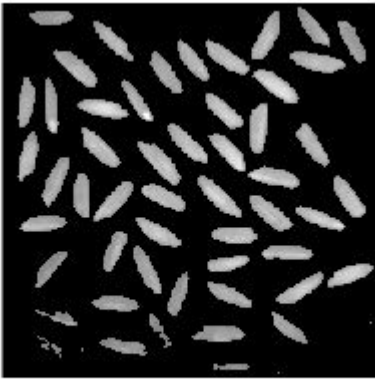
```
imshow(BW,[]);
```



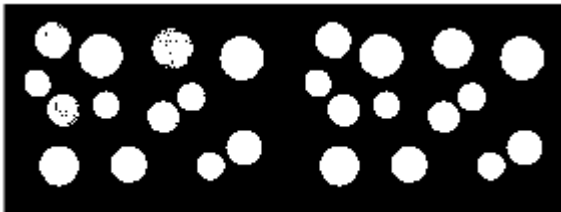
```
%creem una imatge de marques
MARK = false(size(BW));
% marca a les vores
MARK(1,:) = 1;
MARK(end,:) = 1;
MARK(:,1) = 1;
MARK(:,end) = 1;
% fem el reconstruct
REC = imreconstruct(MARK,BW);
imshow(uint8(REC).*I,[]);
```



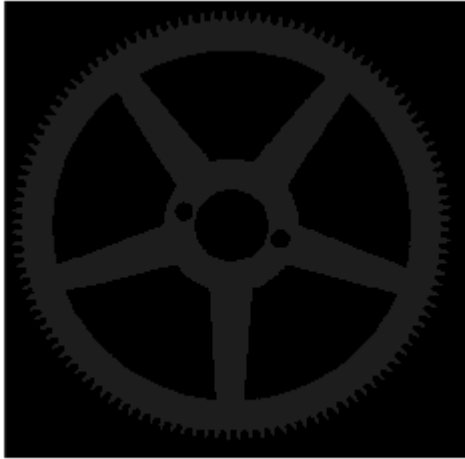
```
BWM = BW & not(REC); % bw millorada
imshow(uint8(BWM).*I,[]);
```



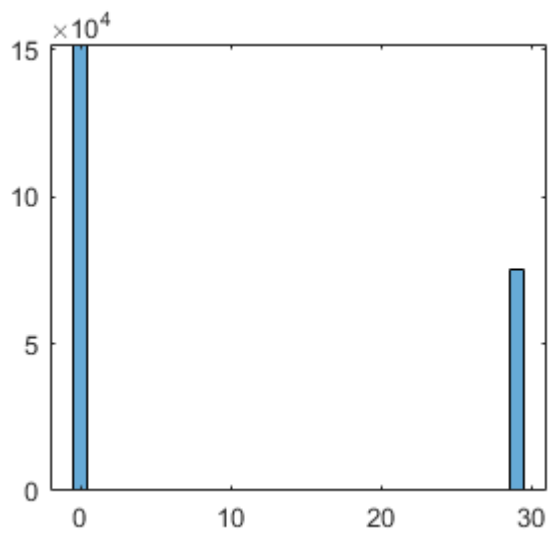
```
% close: fer una dilatació i després una erosió
% - tancar coses que estan properes
% la operació al revés allunyaria els objectes propers (open)
I = imread("money.tif");
BW = I > 128;
SE = ones(7,7);
BWC = imdilate(BW,SE);
BWC = imerode(BWC,SE);
montage({BW,BWC});
```



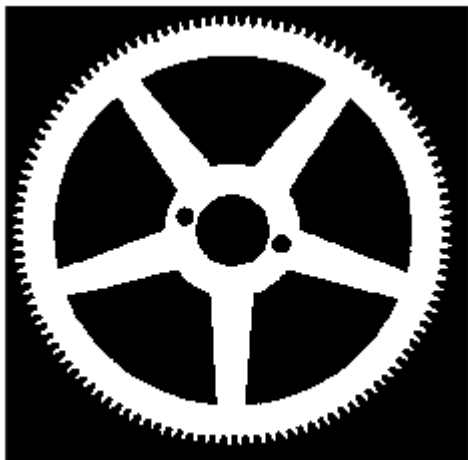
```
% contar les dents d'una roda dentada
I = rgb2gray(imread('Wheel.bmp'));
imshow(I);
```



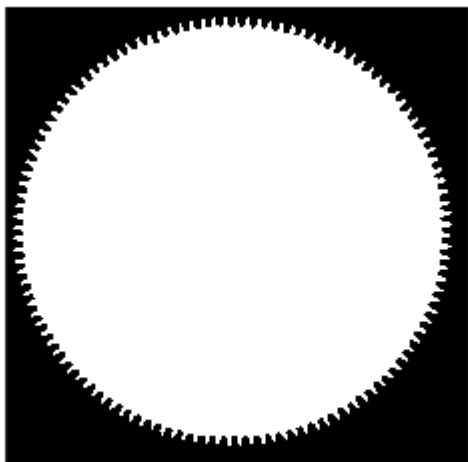
```
histogram(I);
```



```
SE = ones(11,11);  
BW = I > 25;  
imshow(BW);
```

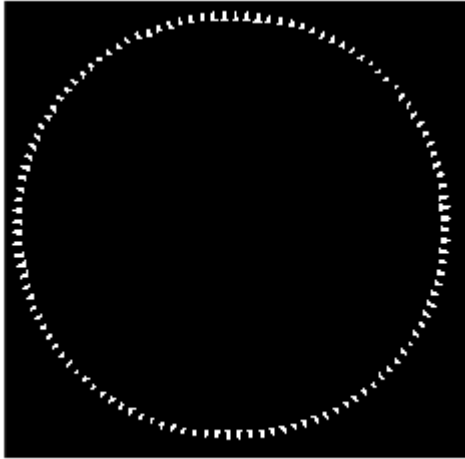


```
MARK = false(size(BW));
% marca a les vores
MARK(1,:) = 1;
MARK(end,:) = 1;
MARK(:,1) = 1;
MARK(:,end) = 1;
REC = imreconstruct(MARK,not(BW));
imshow(not(REC));
```

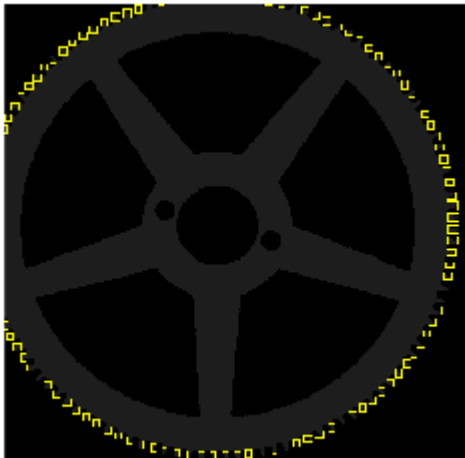


```
test = imopen(not(REC),SE); % aixi queden les dents. amb erosio no quedaven soles
RI = BW & not(test);
imshow(RI);
```





```
% region props
RP = regionprops('table', RI, 'BoundingBox', 'Area');
RGB = insertShape(I, 'rectangle', RP.BoundingBox);
imshow(RGB);
```



```
% amb RP podem saber quants elements hi ha
```

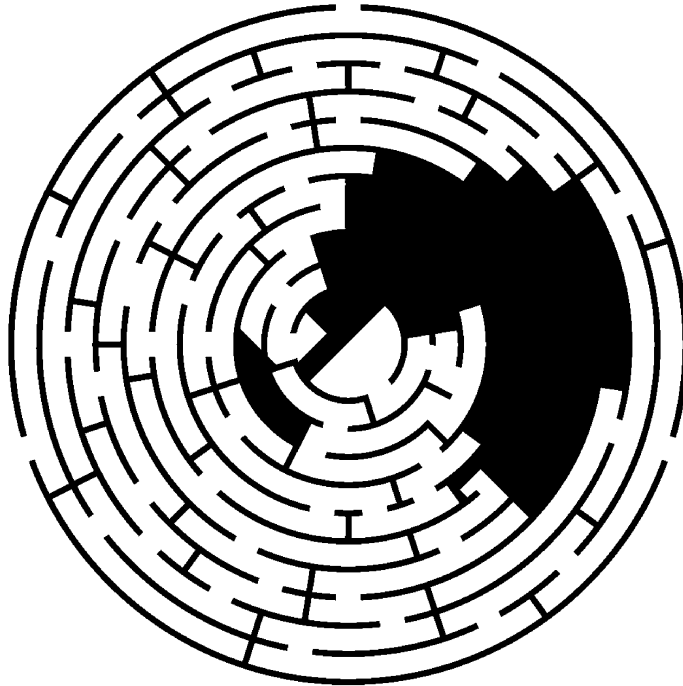
```
% entrar en un laberint
I = rgb2gray(imread('Laberint.png'));
BW = I > 128;
MARK = false(size(BW));
SE = [0,1,0;1,1,1;0,1,0];
DT = zeros(size(BW));
MARK(1,1) = 1; % posicio ini
[f c] = size(BW);
centre = [floor(f/2) floor(c/2)];
```

```

fi = false;
while not(fi)
    MARK = imdilate(MARK,SE) & BW;
    DT = DT + MARK;
    if MARK(centre(1),centre(2))
        fi = true;
    end
end

imshow(MARK);

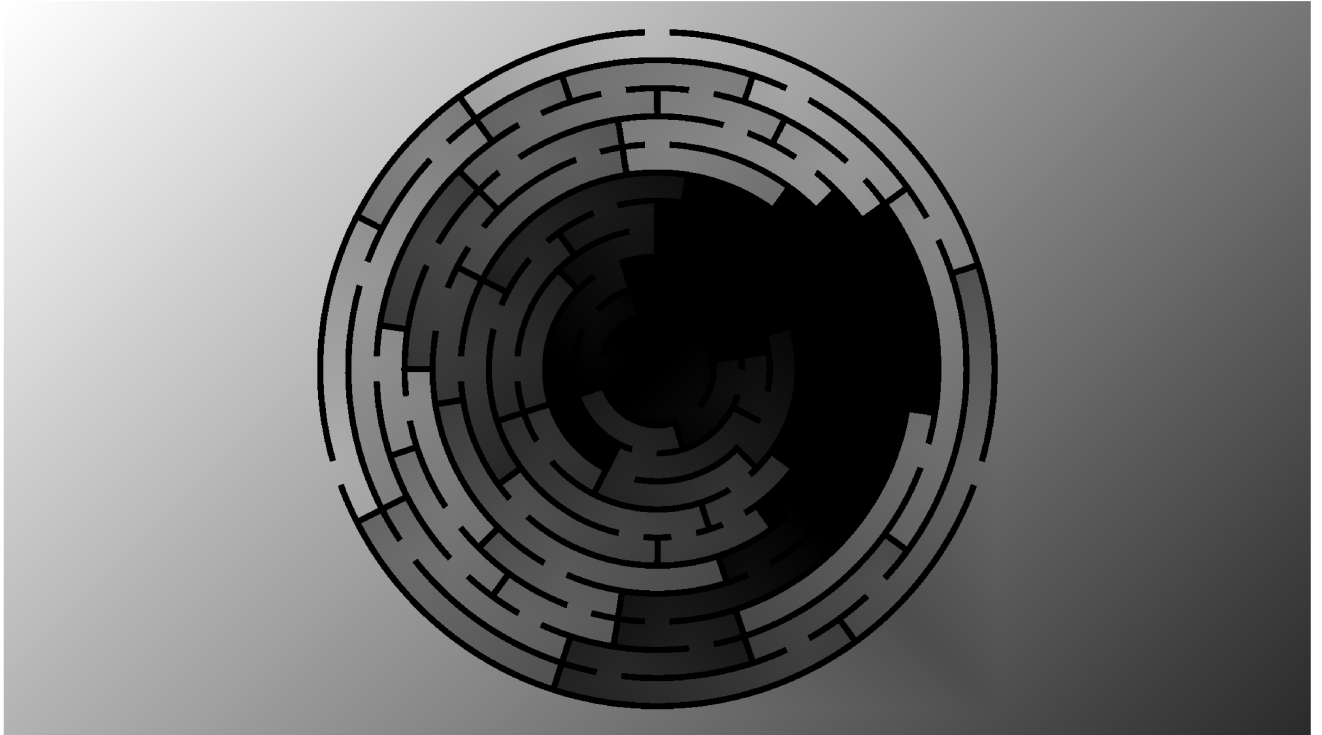
```



```

imshow(DT,[]);

```



% dilatar i fer una and