

**FACULTAT D'INFORMÀTICA DE BARCELONA**

**VISIÓ PER COMPUTADORS**

*INFORME SESSIÓ 2*

Ricard Abril  
Xavier Pastor  
Grup 11 Q1/19-20

# Exercici 1

## Enunciat

Donada una imatge  $I$ , marcar, amb un cercle en la mateixa imatge, la posició de la fila i la columna amb més intensitat (suma dels nivells de gris de la fila i la columna). En cas de més d'un valor màxim no importa quin dels píxels amb major contrast és el seleccionat.

## Resposta

```
1 function [] = exercici1(I)
2     G = rgb2gray(I);
3     [f,c] = size(G);
4
5     %arrays dels sumatoris de les files/columnes
6     sumFiles = zeros(f,'uint64');
7     sumColumns = zeros(c,'uint64');
8
9     for i = 1:f
10         for j = 1:c
11             pixel = double(G(i,j));
12
13             %afegim el pixel a la f/c corresponent
14             sumColumns(j) = sumColumns(j) + pixel;
15             sumFiles(i) = sumFiles(i) + pixel;
16         end
17     end
18
19     %obtenim els maxims de la f/c
20     [m, posMaxFil] = max(sumFiles);
21     [m, posMaxCol] = max(sumColumns);
22
23     %printem
24     outputImage = insertShape(I, 'circle', [posMaxCol(1,1) posMaxFil(1,1) 5], 'LineWidth', 4);
25     imshow(outputImage);
26 end
```

## Exercici 2

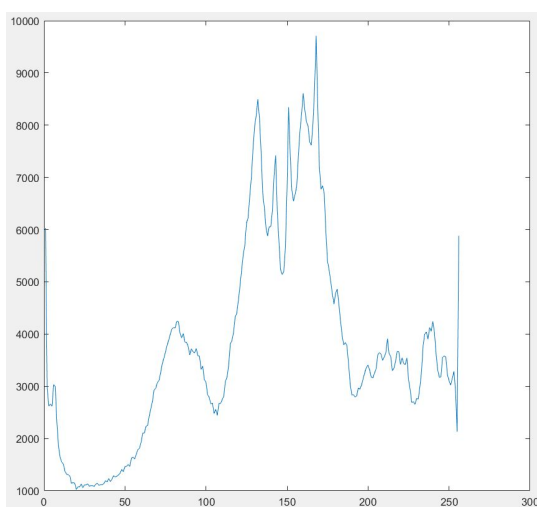
### Enunciat

Implementar un codi que generi l'histograma H d'una imatge (sense utilitzar funcions propies de Matlab) comptabilitzant el nombre d'aparicions dels valors de nivell de gris en N bins (divisions). Utilitzeu el gràfic de barres (bar) per mostrar els resultats.

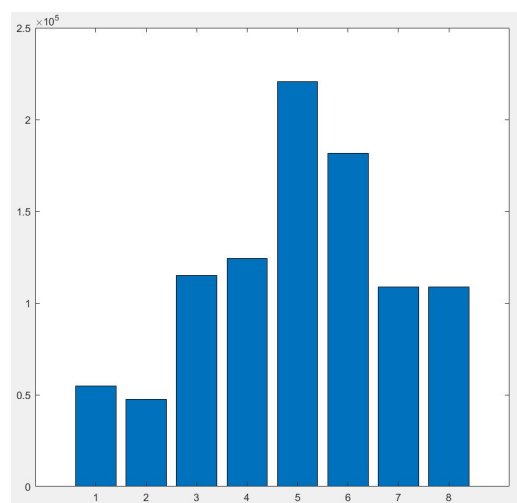
### Resposta

```
1 function [h] = exercici2(I)
2     samples = 8;
3     [f,c]=size(I);
4     h=zeros(1,samples);
5     for i=1:f
6         for j=1:c
7             pixel = I(i,j);
8             z = floor((double(pixel)/256.0) * samples) +1;
9             h(z)=h(z)+1;
10        end
11    end
12 end
```

### Resultat



histograma original (imhist())



histograma de 8 barres (codi)

## Exercici 3

### Enunciat



Calcular el soroll introduït en una imatge al aplicar-li un procés que consisteix en la ampliació de les seves mides en 3 i una reducció en 7 parts i una posterior ampliació per restablir les seves mides. Per calcular el soroll introduït en el procés de reducció-ampliació (Ps) mesurarem la desviació estàndard de la diferència de nivells de gris entre la imatge original i la imatge processada. Compareu els sorolls introduïts mitjançant dos mètodes diferents de interpolació i utilitzant dues imatges: una imatge amb molta textura i una imatge amb poca. Mostreu un quadre comparatiu dels resultats

### Resposta

```
1 function[PS] = exercici3(I)
2     %escala de grisos
3     I = rgb2gray(I);
4     [f,c] = size(I);
5
6     %cambiem tamany amb interpolació
7     I2 = imresize(I, 3/7, 'lanczos3');
8     I2 = imresize(I2,[f c], 'lanczos3');
9
10    %calculem el PS
11    PS = std2(I - I2);
12 end
```

## Resultat

Dins de la funció *imresize* podem passar diferents modes d'interpol·lació. En aquest cas s'ha utilitzat 6 diferents comparant-les amb una imatge amb poca textura (paisatge) i una altra amb molta (carrer).

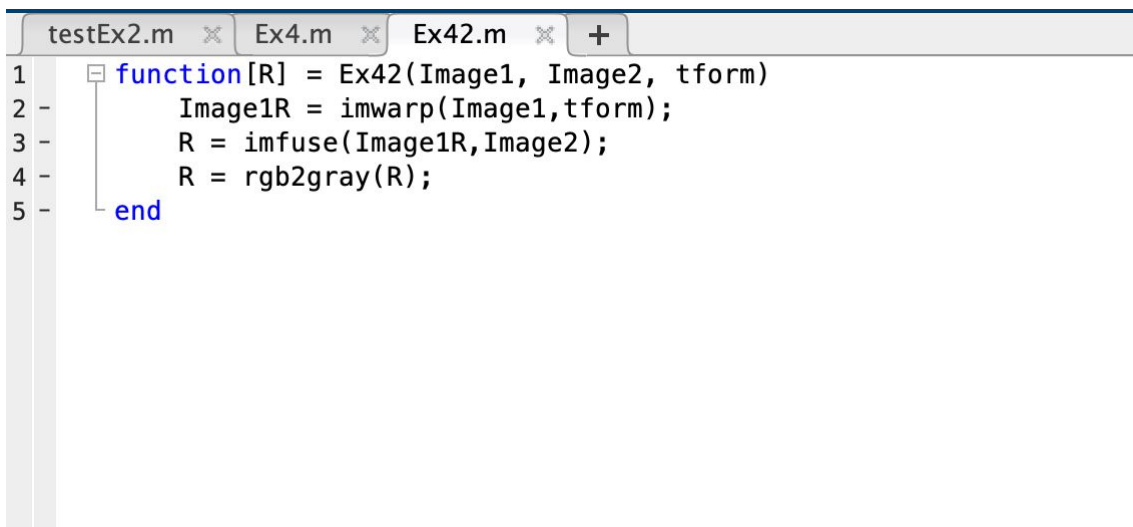
Imatges/Interpolació	nearest	bilinear	box	bicubic	lanczos2	lanczos3
	6.7478	5.1441	5.6532	4.8821	4.8762	4.8057
	20.294	14.886	16.187	14.183	14.1555	13.9598

Es pot observar com en el cas de la interpol·lació bilinear hi ha un bon resultat amb les imatges amb alta textura amb comparació amb una imatge amb poca textura.

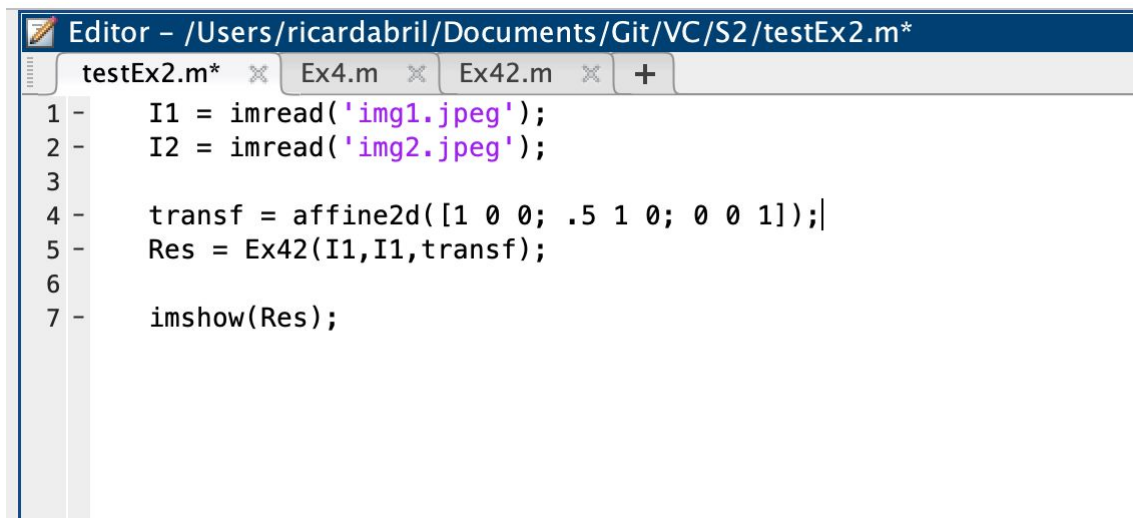
## Exercici 4

Implementar una funció que realitzi una composició d'una imatge A sobre una altre imatge B (image stitching). La transformació que ha de patir la imatge A s'indicarà amb una matriu T que es passarà per paràmetre a la funció. El resultat ha de ser una nova imatge C que contingui a les dues imatges (la imatge B i la imatge A transformada).

### Resposta



```
testEx2.m  x  Ex4.m  x  Ex42.m  x  +
1  function[R] = Ex42(Image1, Image2, tform)
2  -      Image1R = imwarp(Image1,tform);
3  -      R = imfuse(Image1R,Image2);
4  -      R = rgb2gray(R);
5  -  end
```



```
Editor - /Users/ricardabril/Documents/Git/VC/S2/testEx2.m*
testEx2.m*  x  Ex4.m  x  Ex42.m  x  +
1  -      I1 = imread('img1.jpeg');
2  -      I2 = imread('img2.jpeg');
3
4  -      transf = affine2d([1 0 0; .5 1 0; 0 0 1]);
5  -      Res = Ex42(I1,I1,transf);
6
7  -      imshow(Res);
```

## Resultat

