

# **VC: Informe de Laboratori 3**

## **Anàlisi de contorns i enfocat d'imatges**



**Pere Ginebra Solanellas**

**8/3/2021 – Q2 Curs 2020-21**

**Visió per Computador, FIB UPC**

# 1. Introducció

En aquesta sessió començarem a analitzar els contorns dels elements d'una imatge i a experimentar amb les aplicacions que podem donar a aquesta anàlisi. En concret analitzarem el “valor” dels contorns d'una fotografia per determinar-ne la qualitat de l'enfocat al prendre-la.

## 2. Exercici

Començarem llegint les imatges a analitzar i passant-les a escala de grisos en format *double* per facilitar la seva manipulació i anàlisi:

```
img1 = double(rgb2gray(imread('_61A5845.jpg')));  
img2 = double(rgb2gray(imread('_61A5855.jpg')));  
img3 = double(rgb2gray(imread('_61A5861.jpg')));
```

Figura 2.1. Importació d'imatges i transformació a grayscale en format double

A continuació declararem els filtres derivatius que utilitzarem per aïllar els contorns, en concret utilitzarem dos filtres sobel (un horitzontal i un vertical). També es podria utilitzar un sol filtre laplacian per obtenir resultats similars:

```
h_Sobv = fspecial('sobel');  
h_Sobh = h_Sobv';  
% h_Lap = fspecial('laplacian');
```

Figura 2.2. Declaració dels filtres derivatius

Ara apliquem aquests filtres a cada imatge, per això els aplicarem independentment a cada imatge i sumarem el valor absolut dels resultats, ja que el filtre ens donarà valors negatius:

```
s1 = abs(imfilter(img1,h_Sobv)) + abs(imfilter(img1,h_Sobh));  
s2 = abs(imfilter(img2,h_Sobv)) + abs(imfilter(img2,h_Sobh));  
s3 = abs(imfilter(img3,h_Sobv)) + abs(imfilter(img3,h_Sobh));
```

Figura 2.3. Aplicació dels filtres de sobel (horitzontal i vertical)

Amb les següents línies de codi podem observar els resultats obtinguts (figura 3.1):

```
montage({uint8(img1), uint8(img2), uint8(img3)})  
figure  
imshow(s1, [])  
figure  
imshow(s2, [])  
figure  
imshow(s3, [])
```

Figura 2.4. Representació de les imatges inicials i els contorns de cada una

Finalment només ens queda analitzar aquestes imatges per tal de valorar l'enfocament de les fotografies inicials. Per això he decidit ponderar el valor de la posició de cada píxel seguint una distribució normal, així l'enfocament dels píxels del centre té més importància que el dels voltants. Això ho he aconseguit definint una matriu amb la distribució gaussiana al centre (usant un *fspecial('gaussian')* encara que no l'utilitzaré com a filtre) de la mida de les imatges i amb una desviació estàndard definida en funció també de la mida i de la importància a donar al centre (una desviació més petita dona menys importància als exteriors).

```
[rows, columns] = size(img1);
centre_pond = 0.15 %com més baix el valor, més importància se li dona al
matgauss = fspecial('gaussian', [rows, columns], ((rows+columns)/2)*centre_pond);
matgauss = (matgauss)*10000;
```

Figura 2.5. Generació d'una matriu amb una distribució gaussiana

Per acabar he multiplicat aquesta *matgauss* de forma “element-wise” amb cada imatge de contorns per donar a cada píxel la seva ponderació i a continuació he calculat la valoració de cada una a partir del resultat obtingut:

```
m1 = sum(sum(s1.*matgauss))/(rows*columns)
m2 = sum(sum(s2.*matgauss))/(rows*columns)
m3 = sum(sum(s3.*matgauss))/(rows*columns)
```

Figura 2.6. Ponderació dels contorns de les imatges

### 3. Resultats

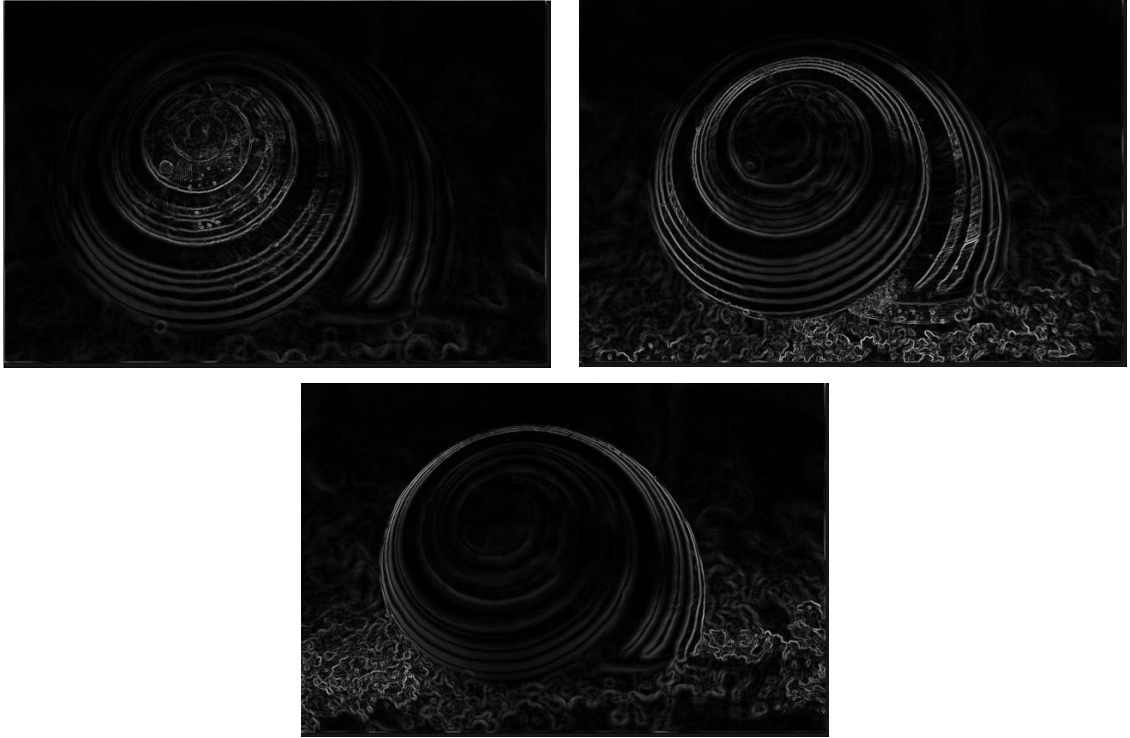


Figura 3.1. Resultat obtingut al aplicar els filtres de sobel

Les imatges de l'enunciat i la ponderació del script ens donen els següents valors de  $m$  per cada imatge:

$m_1 = 3.9022$ ,  $m_2 = 3.6732$  i  $m_3 = 1.9642$

Per tant obtenim que la primera imatge es la que es considera millor enfocada al tenir ben enfocat just el centre de la imatge, seguida de prop per la 2a que esta una mica desenfocada a la part mes central. La ultima imatge podem veure que esta molt mal enfocada.

Si decidíssim donar una mica menys d'importància als píxels centrals i augmentéssim la desviació estàndard de la matriu gaussiana, la segona imatge podria considerar-se més ben enfocada pel script. Amb un valor de *centre\_pond* de 0.2 per exemple, obtenim els següents resultats:

$m_1 = 3.3459$ ,  $m_2 = 3.8284$  i  $m_3 = 2.3058$

A més també veiem que la tercera imatge també es considera una mica més ben enfocada.

Aplicant de nou el script sobre unes noves imatges obtenim resultats semblants:

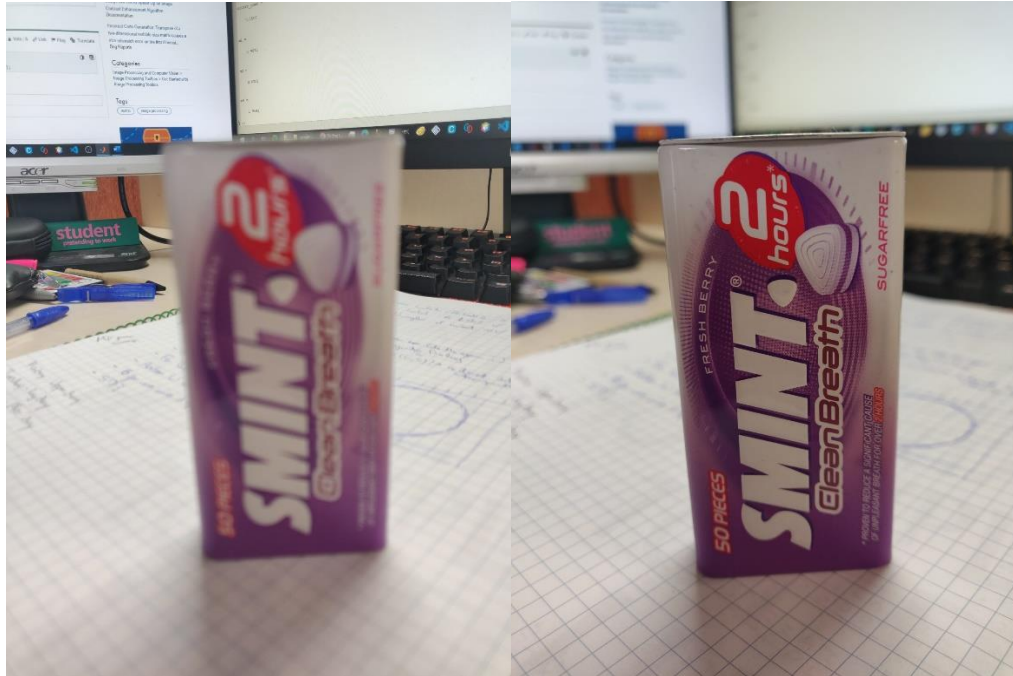


Figura 3.1. Imatges noves



Figura 3.2. Resultat d'aplicar els filtres de sobel sobre les noves imatges

Els valors de  $m$  obtinguts per aquestes imatges son  $m_1 = 0.0513$  i  $m_2 = 0.1174$ , per tant el script determina que la segona imatge està més ben enfocada que la primera al enfocar millor l'element central en comptes del fons.

## 4. Annexos

```
clear
close all

i1 = imread('_61A5845.jpg');
i2 = imread('_61A5855.jpg');
i3 = imread('_61A5861.jpg');

img1 = double(rgb2gray(i1));
img2 = double(rgb2gray(i2));
img3 = double(rgb2gray(i3));

h_Sobv = fspecial('sobel');
h_Sobh = h_Sobv';
% h_Lap = fspecial('laplacian');

s1 = abs(imfilter(img1,h_Sobv)) + abs(imfilter(img1,h_Sobh));
s2 = abs(imfilter(img2,h_Sobv)) + abs(imfilter(img2,h_Sobh));
s3 = abs(imfilter(img3,h_Sobv)) + abs(imfilter(img3,h_Sobh));

montage({i1, i2, i3})
figure
imshow(s1,[])
figure
imshow(s2,[])
figure
imshow(s3,[])

[rows, columns] = size(img1);
centre_pond = 0.2; %com més baix el valor, més importància se li dona al centre
matgauss = fspecial('gaussian',[rows,columns],((rows+columns)/2)*centre_pond );
matgauss = (matgauss)*10000;

m1 = sum(sum(s1.*matgauss))/(rows*columns)
m2 = sum(sum(s2.*matgauss))/(rows*columns)
m3 = sum(sum(s3.*matgauss))/(rows*columns)
```

Figura 4.1. Script usat en la sessió

Per usar imatges diferents només cal canviar el nom dels arxius importats pels *imread*, ja que la matriu gaussiana es genera automàticament a partir de les mides d'aquestes.

## 5. Bibliografia / documentació

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