



**FIB**

Facultat d'Informàtica  
de Barcelona

Departament d'Enginyeria de Sistemes,  
Automàtica i Informàtica Industrial

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# VISIÓ PER COMPUTADOR

## Exercici 3 de Laboratori

Facultat d'Informàtica de Barcelona

Adrian Cristian Crisan

Barcelona, Octubre de 2021

Es pretén dissenyar un algoritme simple d'auto-enfoc. La idea és comparar varies imatges d'una mateixa escena adquirides movent l'enfoc de la càmera (que se suposa motoritzat). Per exemple en la següent seqüència d'imatges (fig1) es pot veure com al anar variant la distància focal de la lent s'obtenen imatges amb enfoc en diferents profunditats. Les imatges les trobareu annexes a aquest document.



Fig 1. D'esquerra a dreta: enfoc molt proper, a mitja distància i passat l'objecte.

Com es pot observar, quant una part de la imatge està desenfocada presenta els contorns suavitzats. Llavors, la idea bàsica del nostre senzill sistema d'auto-enfoc és que la imatge que contingui més "derivada" en la seva part central és que estarà més ben enfocada.

1. El primer que farem és llegir les imatges i les convertim a nivell de grisos (rgb2gray).

```
img1 = imread("_61A5845.jpg");  
img2 = imread("_61A5855.jpg");  
img3 = imread("_61A5861.jpg");  
  
img1 = rgb2gray(img1);  
img2 = rgb2gray(img2);  
img3 = rgb2gray(img3);  
  
montage({img1, img2, img3}, 'Size', [1 3]);
```

Obtenim com a resultat les imatges següents:



2. Passarem un filtre derivatiu (*sobel* o similar) per detectar els contorns verticals  $S_y$  i horitzontals  $S_x$ . Vigileu el tipus de la imatge doncs no us trunqui els valors negatius o valors més grans que 255.
3. Com que en aquesta aplicació és indiferent si el contorn és positiu o negatiu, vertical o horitzontal, calcularem el valor absolut de  $S_x$  i  $S_y$ , i els sumarem.

Els passos 2 i 3 hem fet tres crides (1 per imatge) d'una funció que hem creat.

```
fimg1 = filtre_der(img1);  
fimg2 = filtre_der(img2);  
fimg3 = filtre_der(img3);  
montage({fimg1, fimg2, fimg3}, 'Size', [1 3])
```

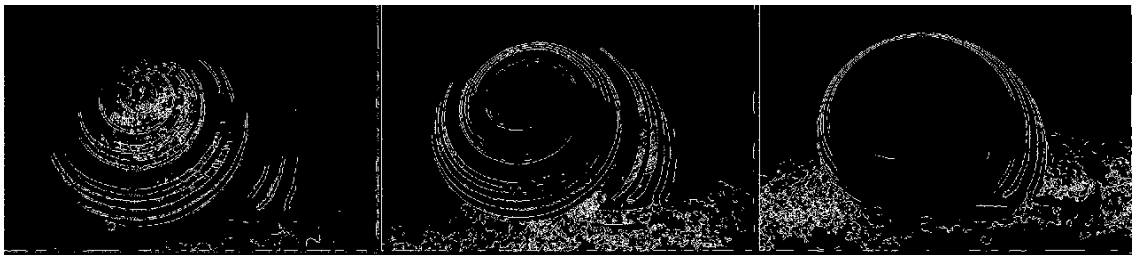
Hem creat dues funcions que es diferencien en el fet que en el primer cas ens ofereix una imatge de contorns binària i la segona ens ofereix una imatge de contorns a escala de grisos.

```
function fimg = filtre_der(img)
Sx = edge(img, 'sobel', 'vertical');
Sy = edge(img, 'sobel', 'horizontal');

fimg = abs(Sx) + abs(Sy);

sup = sum(sum(fimg < 0));
inf = sum(sum(fimg > 255));
if sup > 0 || inf > 0
    error("Valores negativos o mayores a 255");
end
end
```

Utilitzant la funció *edge* la imatge queda de la següent manera:



```
function fimg = filtre_der(img)
h = fspecial('sobel');
Sx = imfilter(img, h);
Sy = imfilter(img, h');
fimg = imadd(Sx, Sy);

sup = sum(sum(fimg < 0));
inf = sum(sum(fimg > 255));
if sup > 0 || inf > 0
    error("Valores negativos o mayores a 255");
end
end
```

Utilitzant la funció *fspecial* la imatge queda de la següent d'aquesta altra manera:



El codi verifica si existeixen valors fora del rang  $[0, 255]$ , en cas d'existir algun valor mostra un error.

4. Calcularem un valor *m* de mèrit de l'enfoc i compararem els resultats de les tres imatges. La imatge que proporcioni un valor *m* més alt és la que es considerarà de millor enfoc. La funció que calcula el valor *m* queda a la vostra elecció, però seria

interessant que ponderes amb major pes els píxels centrals de la imatge que els de la perifèria.

Es fa la crida d'una funció per les 3 imatges donades amb l'objectiu d'obtenir el valor m.

```
m1 = get_m(img1_)
m2 = get_m(img2_)
m3 = get_m(img3_)

function m = get_m(img)
mida = size(img);
if mida(1) > mida(2)
    max_dist = mida(1) / 2;
else
    max_dist = mida(2) / 2;
end

x0 = mida(1) / 2;
y0 = mida(2) / 2;

m = 0;
for x1 = 1:mida(1)
    for y1 = 1:mida(2)
        dist = sqrt((x1 - x0)^2 + (y1 - y0)^2);
        m = m + int32(img(x1, y1)) * (1 - dist/max_dist);
    end
end
end
```

La funció primerament calcula el nombre de píxels de l'horitzontal i la vertical i depenent de quina de les dues sigui major l'agafa, el divideix entre 2 i l'utilitza com a màxima distància (per aquestes imatges sabem que hi ha més píxels d'ample que de llarg, però l'algorisme està pensat per si la situació canvia).

Posteriorment, obtenim els píxels que actuaran com a centre de la imatge ( $x_0$  i  $y_0$ ) i mitjançant la següent fórmula:

$$Distancia = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$

Calculem la distància entre dos píxels.

Finalment, accedim a la posició  $(x_1, y_1)$  de la matriu, per saber si existeix o no contorn (int32 el posem perquè la matriu està representada amb valors lògics amb els quals no podem operar, en el cas del filtratge *edge*, en el cas del filtratge *fspecial*, si no poséssim int32, el valor màxim de m seria 255 perquè la matriu està representada amb int8). Obtenim el valor de la matriu en la posició especificada i la multipliquem pel ràtio de la distància amb la distància màxima i aquest ràtio el restem a 1, això és perquè la ponderació sigui favorable pels píxels més propers al centre i no a la inversa.

## Conclusió

Utilitzant la funció on utilitzem el filtre *edge* obtenim que la imatge millor enfocada és la primera (d'esquerra a dreta), en canvi utilitzant el filtre *fspecial* la imatge millor enfocada, segons el mateix algorisme, és la segona.

L'explicació que trobo, és que la funció que utilitza *edge* únicament detecta contorns de forma binària (0 si no és contorn, 1 si és contorn), en canvi amb la funció *fspecial* ens ofereix els contorns on es pot observar quina és la profunditat d'aquests, amb rang que van des de 0 a 255. Per tant, amb la funció que utilitza *edge* detecta que la imatge està enfocada al mig i amb la funció que utilitza *fspecial* detecta que el mig està lleugerament enfocat (detecta alguns contorns amb no gaire profunditat degut al suavitzat) i que la imatge en conjunt conté una quantitat de suavitzat a causa de la distància focal de la lent més baixa comparada amb les altres imatges.

## Proves amb altres imatges

Aplicant el mateix algorisme que en el cas anterior i utilitzant la funció *fspecial* per filtrar les imatges, per les següents imatges obtenim els següents resultats:

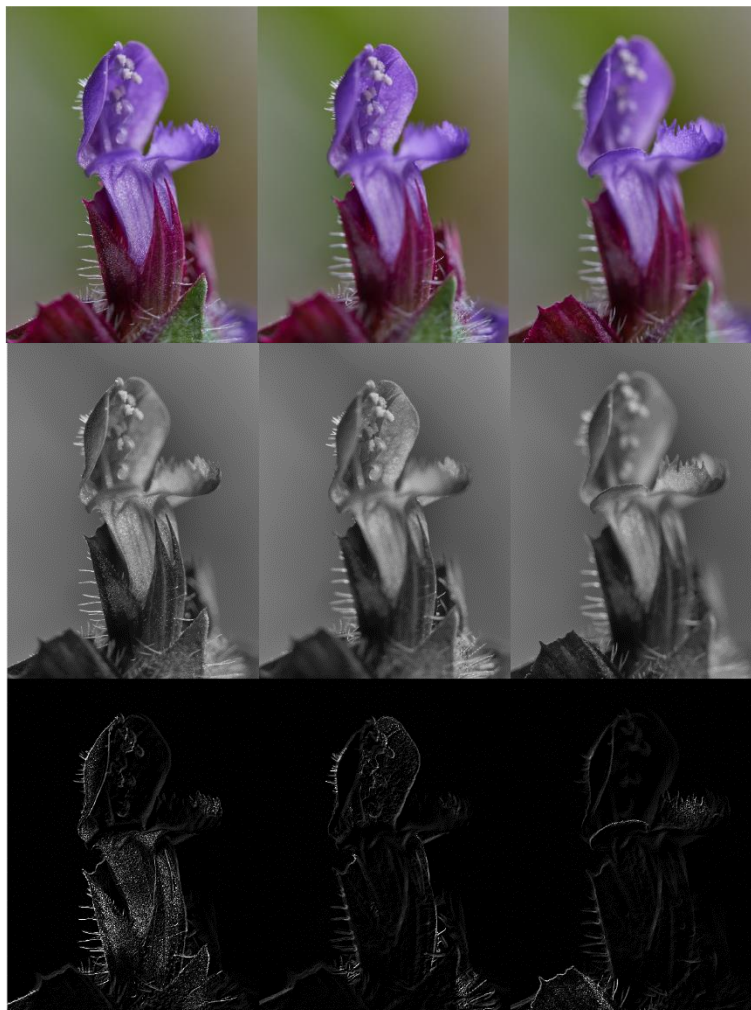


Fig 2: de menys a més enfocada (d'esquerra a dreta)

L'algorisme ens dona com a resultat que la primera imatge és la més enfocada al mig.



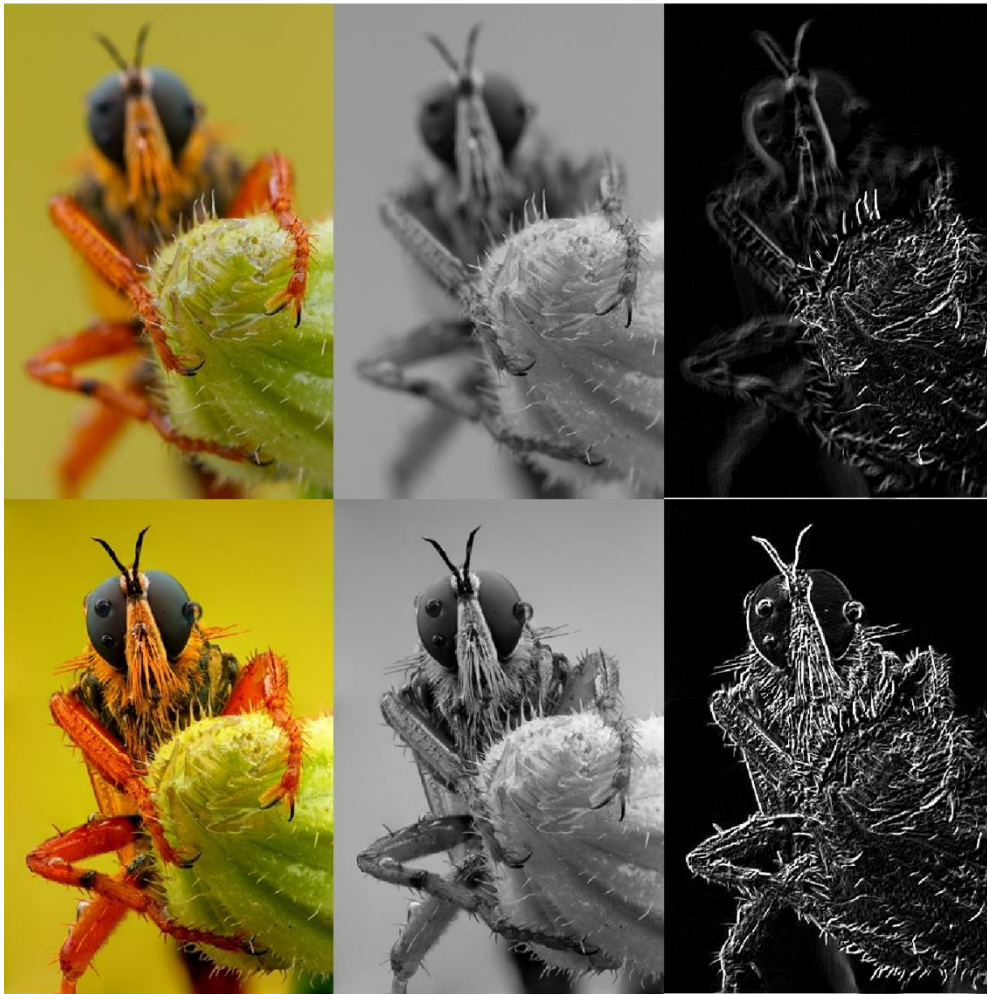


Fig 3: imatge més enfocada a baix i menys enfocada a dalt

Obtenim que la imatge inferior està millor enfocada