

# Universitat Politècnica de Catalunya

Visió per Computador  
Grau en Enginyeria Informàtica

---

## EXERCICI 3 DE LABORATORI

### IMAGE AUTO-FOCUSING ALGORITHM

---

*Autor:*  
Daniel DONATE

*Professor:*  
Manuel FRIGOLA  
Q2 Curs 2020-2021

9 de març del 2021



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

---

Facultat d'Informàtica de Barcelona



## Enunciat de l'exercici

El propòsit d'aquest exercici és dissenyar un algoritme molt simple d'auto-enfoc. La idea consisteix en comparar diferents imatges d'una mateixa escena adquirides movent l'enfoc de la càmera. Per exemple, en la *Figura 1* podem veure com al anar variant la distància focal de la lent s'obtenen imatges amb enfocs en diferents profunditats.



*Figura 1: D'esquerra a dreta, enfocament molt proper, a mitja distància i passat l'objecte*

Com podem apreciar, la imatge de més a l'esquerra és la que està millor centrada de les tres. Notem que quan part de la imatge està desenfocada, presenta els contorns suavitzats. La idea és, doncs, construir un algoritme simple que detecti quina de les imatges d'una escena que s'analitzen té més "derivada" (més variació) en la seva part central, ja que aquesta, com passa amb les imatges de la *Figura 1*, és la que estarà millor centrada.

A continuació descriurem les quatre parts de l'algoritme que proposem.

## Construcció d'un algoritme d'auto-enfocament

### 1. Llegint les imatges

Primer, llegim les diferents imatges de l'escena, les convertim de l'espai RGB a nivell de grisos i les passem a format de dades *double*. Per la primera de les imatges, per exemple:

```
I1 = imread('_61A5845.jpg');
I1 = double(rgb2gray(I1));
```

### 2. Aplicant un filtre de Sobel

A continuació aplicarem a cadascuna de les imatges de l'escena un filtre de Sobel. Recordem que el de Sobel és un operador diferencial que calcula una (molt bona) aproximació al gradient de la funció d'intensitat a cada píxel d'una imatge (podríem haver escollit altres operadors diferencials, com el de Prewitt, per exemple). L'operador de Sobel s'utilitza per detectar contorns verticals i contorns horitzontals d'una imatge.

Matemàticament, l'operador utilitza dos *kernels* de 3×3 elements (un pel gradient horitzontal i un altre pel gradient vertical) per aplicar la convolució a la imatge original i així calcular les (aproximacions de) les derivades. Aquests dos *kernels*,  $H_h$  i  $H_v$ , són, respectivament:

$$H_h = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad H_v = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Notem que  $H_v = H_h^T$ , així que podem calcular els gradients  $G_x$  i  $G_y$  de les imatges de la següent manera, utilitzant la funció *imfilter* de MATLAB:

```
Hh = [-1,-2,-1;0,0,0;1,2,1];
Hv = Hh';
GX1 = imfilter(I1,Hh);
GY1 = imfilter(I1,Hv);
```

### 3. Obtenint la magnitud del gradient

Notem que, en cada píxel de la imatge, podem combinar els resultats de les aproximacions dels gradients horitzontal  $G_x$  i vertical  $G_y$  per obtenir la magnitud del gradient. Com en aquesta aplicació és indiferent si el contorn és positiu o negatiu, horitzontal o vertical, calcularem el valor absolut de  $G_x$  i  $G_y$  i els sumarem, de manera que:  $G = |G_x| + |G_y|$ . Per la primera imatge, tenim:

```
G1 = abs(GX1) + abs(GY1);
```

A continuació mostrem els models de contorn a sota de les tres imatges originals (en escala de grisos) de les tres imatges de la *Figura 1*. Notem que hem de fer servir les claus `[]` a la crida `imshow` per re-escalar les imatges en funció del rang de valors dels píxels de les matrius  $I$  i  $G$ .

```
imshow(I1,[])  
imshow(G1,[])
```

Com un esperaria, la imatge que presenta més canvis en la zona central és la primera, doncs és la que està més centrada de les tres. Això equival a dir que els valors del gradient en aquella zona són més alts que a les altres dues imatges. Aquest és el fet que farem servir al pas 4 per determinar quina de les imatges està més ben centrada.



*Figura 2: Model de contorn de les tres imatges (en escala de grisos) originals. Notem que les línies són més nítidament blanques a les zones de cada imatge on es situa el punt focal, i menys a les zones suavitzades*

### 4. Calculant un valor de mèrit de l'enfoc

L'últim pas de l'algoritme consisteix en calcular un valor  $m$  de mèrit de l'enfoc i comparar el resultat de les tres imatges, de tal manera que la imatge que proporcioni un valor de  $m$  més alt sigui la que es considerarà de millor enfoc. Com hem anunciat anteriorment, la funció que calculi aquest valor  $m$  ha de ponderar amb major pes els píxels centrals de la imatge que els de la perifèria. Per tant, hi ha moltes funcions que podríem fer servir per calcular aquest valor.

La meua elecció ha estat aplicar una funció gaussiana de dos dimensions,  $f(x, y)$ , sobre la matriu  $G$  de cada imatge filtrada. Com sabem, la campana de Gauss 2D és una funció exponencial on l'exponent de la potència de  $e$  és qualsevol valor negatiu i definit de forma quadràtica. En conseqüència, els nivells de la funció sempre són el·lipses. Formalment, tenim:

$$f(x, y) = A \exp \left( - \left( \frac{(x - x_0)^2}{2\sigma_x^2} + \frac{(y - y_0)^2}{2\sigma_y^2} \right) \right)$$

on  $A$  és l'amplitud,  $(x_0, y_0)$  és el centre i  $\sigma$  són les variàncies de les variables  $x$  i  $y$ . En particular, la funció que he escollit té  $A = 1$  i  $\sigma = 0.2$  (veure *Figura 3*). Recordem que el valor de la variància

és el que determina l'amplitud de la campana de Gauss. Notem que quan el valor de  $\sigma$  és molt petit, la importància recau molt acusadament sobre la part central de la imatge. A mesura que  $\sigma$  creix, comencen a tenir una mica més d'importància els valors dels píxels que no són centrals.

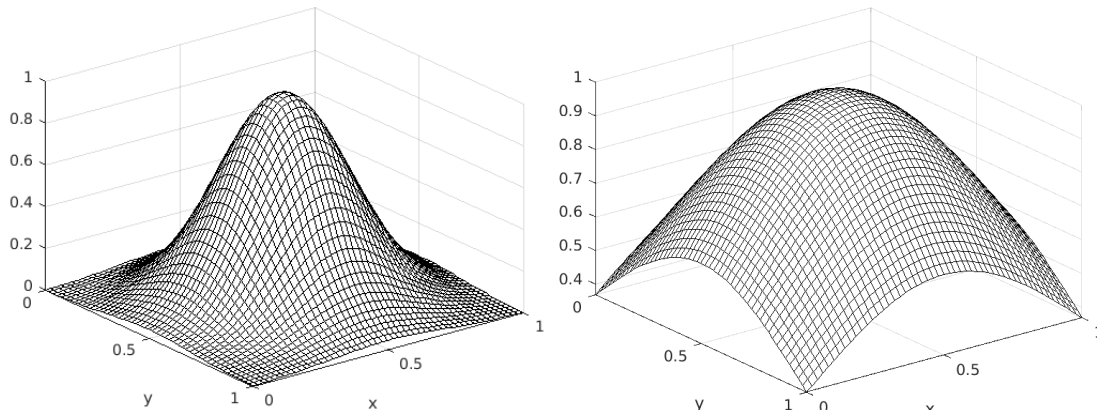


Figura 3: Forma d'una funció gaussiana amb  $A = 1$ ,  $\sigma = 0.2$  (esquerra) i  $\sigma = 0.5$  (dreta)

Tenim moltes altres funcions que podríem, igualment, fer servir en aquest exercici com, per exemple, una finestra Hamming 2D (Figura 4). Ara bé, totes han de produir valors de  $m$  'similars'.

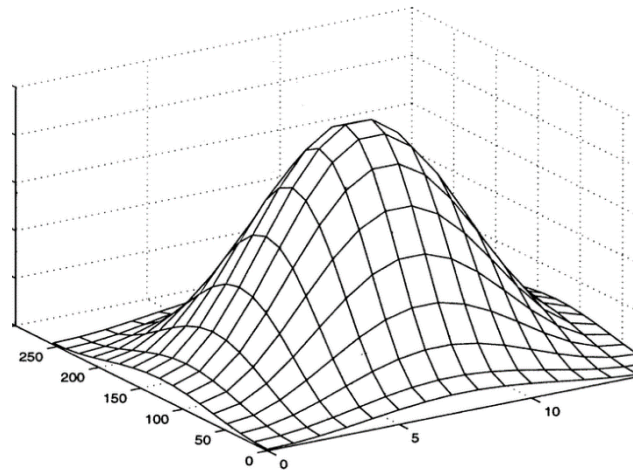


Figura 4: Forma d'una finestra de Hamming,  $h$

Així, el valor de mèrit de l'enfoc l'he definit com:  $m = \sum_{i=1}^f \sum_{j=1}^c \exp \left( - \left( \frac{(i-x_0)^2 + (j-y_0)^2}{2\sigma^2} \right) \right) G_{i,j}$ ,

essent  $(f, c)$  el nombre de files i columnes de la matriu  $G$ , i havent definit el centre de la imatge de contorns  $(x_0, y_0)$  de la forma lògica, és a dir:  $x_0 = \lfloor (f+1)/2 \rfloor$ ,  $y_0 = \lfloor (c+1)/2 \rfloor$ . Com la mida de les tres imatges és la mateixa, podem calcular els seus valors de  $m$  amb un sol bucle:

```
[f,c] = size(G1);
xc = floor((f+1)/2); yc = floor((c+1)/2);
sigma = 0.2; m1 = 0; m2 = 0; m3 = 0;
for i=1:f
    for j=1:c
        exponent = ((i-xc).^2 + (j-yc).^2)./(2*sigma^2);
        m1 = m1 + (exp(-exponent))*G1(i,j);
        m2 = m2 + (exp(-exponent))*G2(i,j);
        m3 = m3 + (exp(-exponent))*G3(i,j);
    end
end
```

Lògicament, el valor de  $m$  més gran que obtenim és el de la primera imatge. No obstant, cal a dir que, amb una  $\sigma$  tan petita ( $\sigma = 0.2$ ), la funció dona 'massa' pes a la regió central i tenim que el valor  $m$  que calculem per la segona imatge és més petit que el que obtenim per la tercera, i no hauria de ser així, ja que la tercera està menys centrada que la segona. Això es soluciona fàcilment incrementant aquest valor de la variància. Per exemple, amb un  $\sigma \geq 0.5$  ja tenim un ordre relatiu de  $m$  correcte per les tres imatges.

## Provant altres imatges

Per acabar, comprovarem que l'algoritme que hem presentat funciona correctament amb altres imatges i, de pas, aprofitarem per parlar d'un altre tipus d'imatges amb les que podem treballar.

La següent figura mostra dues fotografies amb dàlies que s'han pres usant un punt focal diferent (marcat amb un quadrat vermell). Clarament, la imatge de l'esquerra està més centrada i produeix un efecte visual molt més agradable que la imatge de la dreta.



Figura 5: Fotografia d'unes dàlies feta amb diferents punts focals. Autora: Suzanne Williams

Utilitzant l'anterior algoritme obtenim, amb un valor de  $\sigma = 0.5$ ,  $m_1 = 49.05$ ,  $m_2 = 30.98$ , de manera que, com esperàvem, escollim la primera com a imatge més centrada. En aquest cas la inspecció de la imatge obtinguda després de passar pel filtre de Sobel és molt clara:

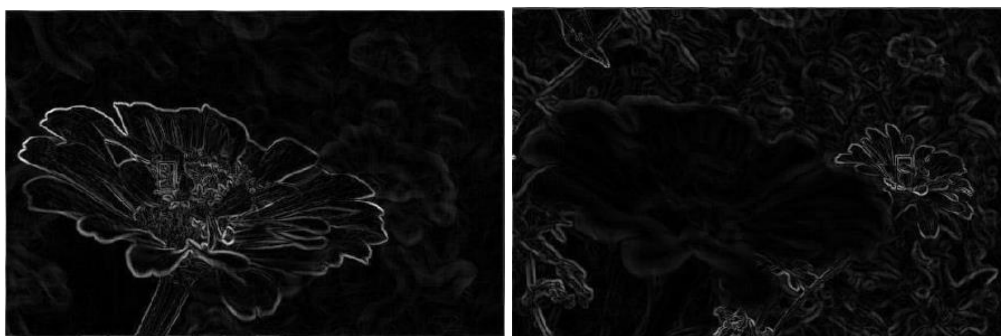


Figura 6: Contorns de les flors de les dues imatges anteriors

Una altra àrea on es poden aplicar tècniques derivades de les que hem vist en aquest exercici és en la microscòpia. Les següents imatges, per exemple, mostren un microfòssil diatom en terra de diatomees. Les tres imatges estan capturades amb tres distàncies focals diferents. Una aplicació habitual del processat d'imatges consisteix en combinar imatges com aquestes mitjançant una tècnica coneguda com *focus stacking* per produir una imatge resultant amb una profunditat de camp més gran que qualsevol de les imatges d'entrada.

Amb una ràpida inspecció veiem que la imatge de més a la dreta és la que mostra més nítidament els detalls de l'alga, mentre que la de més a l'esquerra és la que està més descentrada.



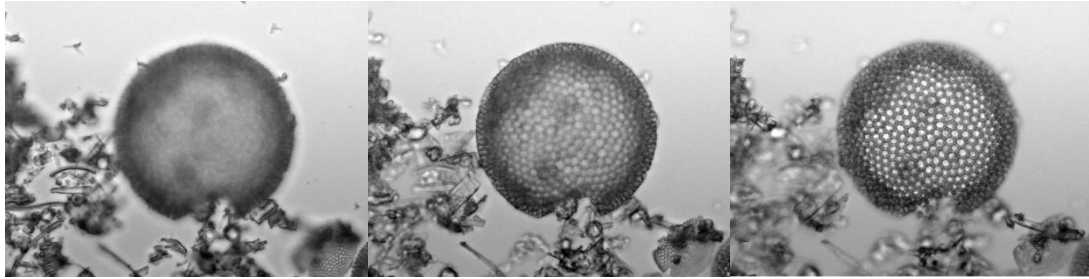


Figura 7: Imatges de diatomees sota el microscopi preses amb diferents distàncies focals. Autor: Richard Wheeler

Així que no ens sorprèn obtenir uns valors  $m_1 = 19.72$ ,  $m_2 = 61.73$  i  $m_3 = 113.21$  (novament amb  $\sigma = 0.5$ ) que ens indiquen que la imatge més ben centrada és la tercera, com clarament s'aprecia amb una ràpida inspecció dels contorns obtinguts amb el filtre de Sobel (Figura 8).

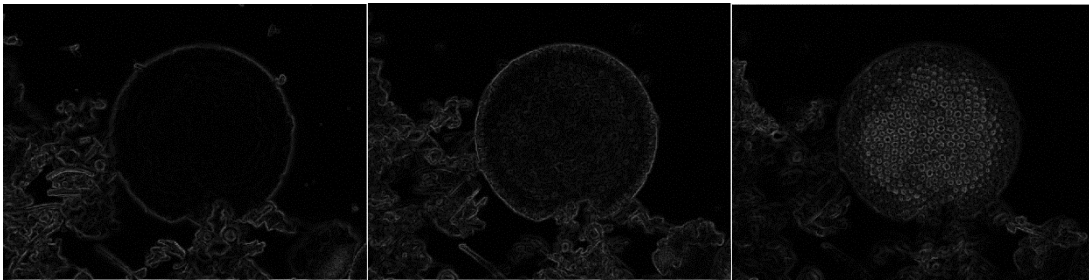


Figura 8: Contorns de les diatomees de les tres imatges anteriors