VISIÓ PER

COMPUTADOR

Informe de laboratori 3

Marc Cervilla Rovira

10/03/2021

Introducció:

En aquesta sessió de laboratori aplicarem els coneixements obtinguts en el filtrat d’imatges per tal de dissenyar un senzill algorisme que tracti l’enfoc de les imatges. A partir de varies imatges d’un mateix objecte aplicant diferents enfocs voldrem calcular una espècie de índex que determinarà que tant enfocada al objecte està la imatge.

Partirem de les següents imatges:



on, d’esquerra a dreta, l’enfoc passa de davant de l’objecte a darrera.

Implementació:

La idea bàsica que voldrem implementar serà, partint de la premissa que quan una part d’una imatge està desenfocada presenta els contorns suavitzats, la imatge amb més derivada a la part central estarà millor enfocada. Així doncs, de les imatges presentades anteriorment, la que hauria de tenir major derivada i, per tant, un índex d’enfoc més elevat serà la primera.

*\*Durant tota l’explicació fico el codi aplicat a una imatge, ja que a les altres se’ls hi aplica exactament les mateixes instruccions, en l’algorisme aplicariem el codi a cada imatge i comparariem els resultats.*

En primer lloc, llegim les imatges i les convertim en escala de grisos.

A = imread('\_61A5845.jpg');

A = rgb2gray(A);

A continuació, generem els filtres derivatius que passarem a les imatges, en el meu cas utilitzo el filtre de sobel *hv* i el seu transposat *hh* per tal d'observar contorns en vertical i en horitzontal.

hv = fspecial('sobel');

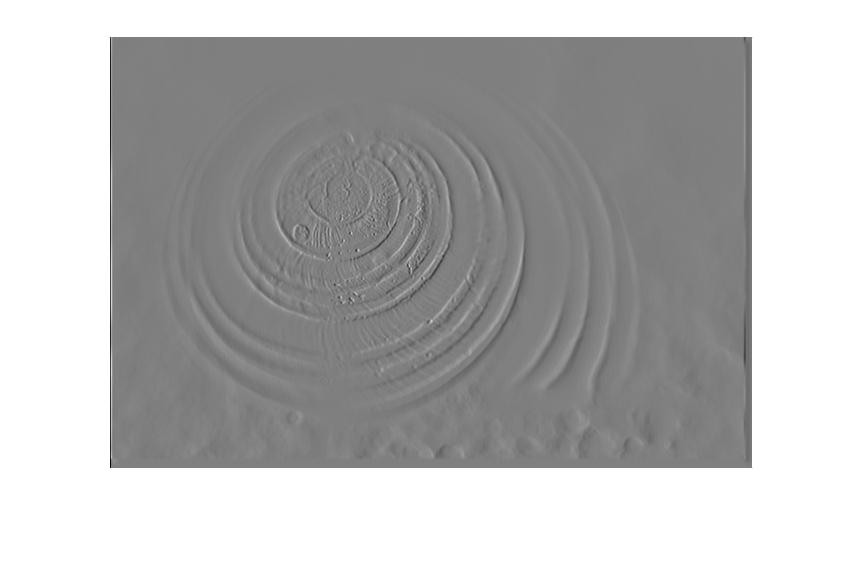
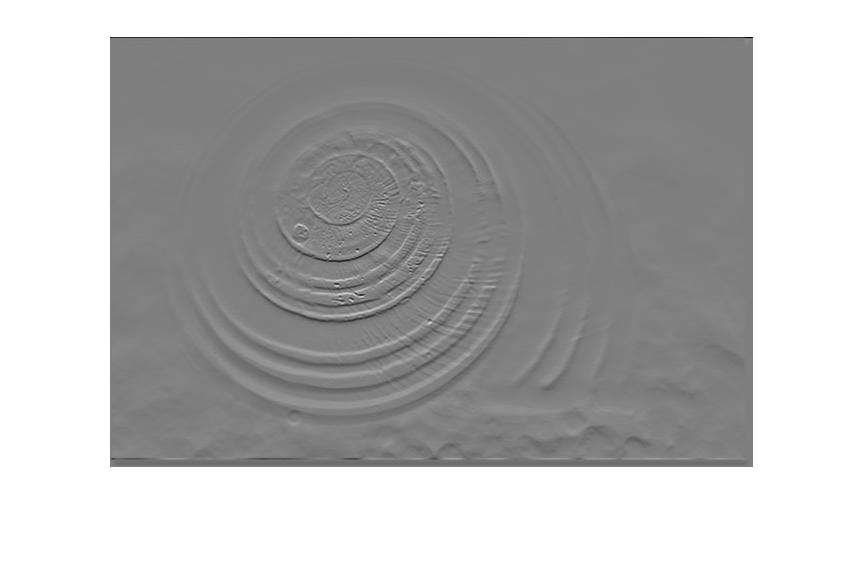
hh = hv';

Aplicant aquests filtres obtindrem per a cada imatge un parell d’imatges amb els filtratges en vertical i en horitzontal.

AV = imfilter(double(A),hv);

AH = imfilter(double(A),hh);

Per exemple, en el cas de la primera imatge se’ns generen el parell d’imatges:



filtratge vertical filtratge horitzontal

Com s’observa en el codi, quan filtrem, hem passat les imatges a double, per a que al aplicar el filtre les dades no surtin del rang. Després de fer això per tal de mostrar les imatges sempre utilitzarem el reescalat dels píxels de la següent manera:

imshow(AV,[]);

Tot seguit, com que volem calcular el contorn en general no fa falta que diferenciem entre el vertical i horitzontal. Utilitzem els dos ja que si ara apliquem el valor absolut de les imatges anteriors i les sumem, ens donarà un resultat dels contorns bastant bo:

AF = abs(AV)+abs(AH);

Ara podem comparar les diferents imatges per tal de veure com actua l’enfoc en els contorns:



AF



BF



CF

En les anteriors imatges es veu clarament la idea del algorisme, ja que a mesura que l’enfoc passa més cap a darrere ( i suposant que la imatge està centrada) els contorns passen del centre de la imatge cap a les bandes.

Per tant, a continuació, voldrem calcular un valor **m** que ens estimi que tant enfocada està la imatge en funció dels contorns centrals de la imatge, per exemple, en les imatges anteriors el valor m més elevat seria a la imatge AF.

La manera que proposo per realitzar aquest càlcul sería:

En primer lloc, normalitzar la imatge ja filtrada a valors entre 0 i 1 (dividint-la pel seu màxim).

MAX = max(max(AF));

ANorm = AF./MAX;

Després creo una matriu gaussiana del tamany [files columnes] de la imatge inicial, la qual tindrà un seguit de pesos distribuïts de manera que com més al centre de la matriu se situen més alts son, per tant, com més lluny del centre estigui un pixel menys pes tindrà.

Un punt molt important és aplicar una bona desviació estàndard tenint en compte el tamany de la matriu, ja que aquesta marca en quin nivell disminueixen els pesos conforme s’allunyen del centre i podriem estar donant bastant pes a la perifèria de la imatge, per tant, no ha de ser un valor molt elevat i depèn del tamany de la imatge, algo així com la dimensió més gran entre 10, assegurant-nos així que disminueixen ràpid els pesos.

files = size(AF,1);

columnes = size(AF,2);

sigma = max([files columnes])/10;

GAUS = fspecial('gaussian', [files columnes], sigma);

Per últim, creem una matriu final on cada píxel tindrà un valor determinat per:

pes del pixel i,j \* color del píxel en la imatge filtrada,

així doncs, aquests valors seràn més elevats com més contorn siguin (color més blanc) i com més al centre estiguin.

F = zeros(files,columnes);

for i = 1:files

for j = 1:columnes

F(i,j) = AF(i,j)\*GAUS(i,j);

end

end

Queda només evaluar el valor **m** en funció d’aquesta matriu, es podria fer de moltes maneres, amb mediana, mitjana, suma, etc.

Com que amb la matriu gaussiana queden valors molt petits la que més m’agrada és la suma dels elements de la matriu:

m = sum(sum(F));

El resultat final de l’algorisme aplicat a les imatges anteriors seria el següent:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **AF** | **BF** | **CF** |
| **m** | 120.15 | 94.56 | 46.71 |

Que com ja havíem dit abans, el valor de m més elevat és en AF on la imatge està més ben enfocada i disminueix conforme es desenfoca en BF i CF.