

ENTREGA LABORATORI 5

En aquesta entrega es demanava avaluar la precisió de l'operador “Sobel” contra el “Prewitt” en estimar l'orientació d'un contorn.

Per tal de realitzar aquesta avaluació s'ha creat una imatge de 77x77 on la meitat superior es blanc i la meitat inferior negra:

```

3
4 -      Im = zeros(77);
5 -      for j = 1:77
6 -          for i = 1:ceil(77/2)
7 -              Im(i,j) = 255;
8 -          end
9 -      end

```

Un cop tenim la imatge, s'ha creat un bucle que per cada iteració gira la imatge 1 grau respecte la iteració anterior, es a dir, el bucle va de 0 a 180 per rotar un grau més progressivament. Per cada rotació a l'hora de rotar s'ha fet un 'bicubic' i un 'crop' per tal de que la imatge sempre tingui el mateix tamany. Un cop rotada i havent aplicat un filtre gaussià per tal de suavitzar la rotació, s'ha avaluat el píxel central de la imatge fent l'arctangent del valor aplicant el filtre, tant per Sobel, guardant-lo al vector “vectors”, com per Prewitt, guardant-lo al vector “vectorp”.

```

10
11 -      vectors = zeros(1, 181);
12 -      Sobel = fspecial('sobel');
13 -      vectorp = zeros(1, 181);
14 -      Prewitt = fspecial('prewitt');
15 -      H = fspecial('gaussian', 15);
16
17      % Bucle: a cada iteració calcules l'arctangent del valor central.
18 -      for a = 0:180
19 -          % rotate de la imatge original segons alfa(a)
20 -          Irotate = imrotate(Im, a, 'bicubic', 'crop');
21 -          % filtre gaussia
22 -          Ifiltered = imfilter(Irotate, H);
23 -          [f, c] = size(Ifiltered);
24 -          % SOBEL
25 -          SX = imfilter(Ifiltered, Sobel);
26 -          SY = imfilter(Ifiltered, transpose(Sobel));
27 -          sx = SX(uint8(f/2), uint8(c/2));
28 -          sy = SY(uint8(f/2), uint8(c/2));
29 -          vectors(a+1) = (atan2(sy, sx)*180)/pi;
30
31          %PREWITT
32 -          PX = imfilter(Ifiltered, Prewitt);
33 -          PY = imfilter(Ifiltered, transpose(Prewitt));
34 -          px = PX(uint8(f/2), uint8(c/2));
35 -          py = PY(uint8(f/2), uint8(c/2));
36 -          vectorp(a+1) = (atan2(py, px)*180)/pi;
37
38 -      end

```

Un cop realitzades totes les iteracions del bucle tenim dos vectors que contenen tots els valors un avaluat el píxel central de la imatge, s'han creat les gràfiques, tant d'error com les calculades durant tot el bucle.

```
39 -  
40 -     errs = (0:180) - vectors;  
41 -     errp = (0:180) - vectorp;  
42 -  
43 -     figure('Name','Anàlisis Sobel'), plot(vectors);  
44 -     figure('Name','Error Sobel'), plot(errs);  
45 -     figure('Name','Anàlisis Prewitt'), plot(vectorp);  
46 -     figure('Name','Error Prewitt'), plot(errp);  
47 -  
48 - end
```

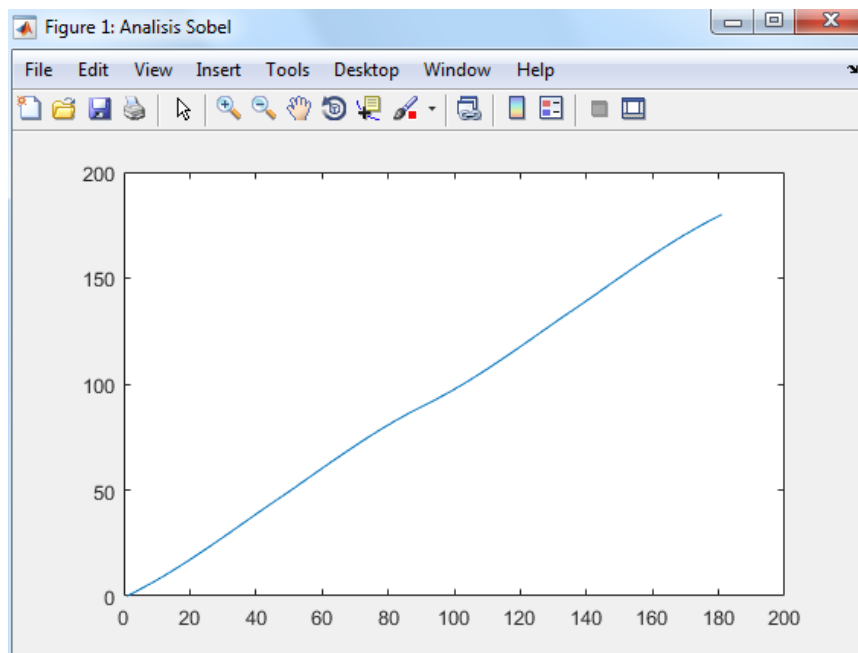


Figura 1: Funció beta per Sobel

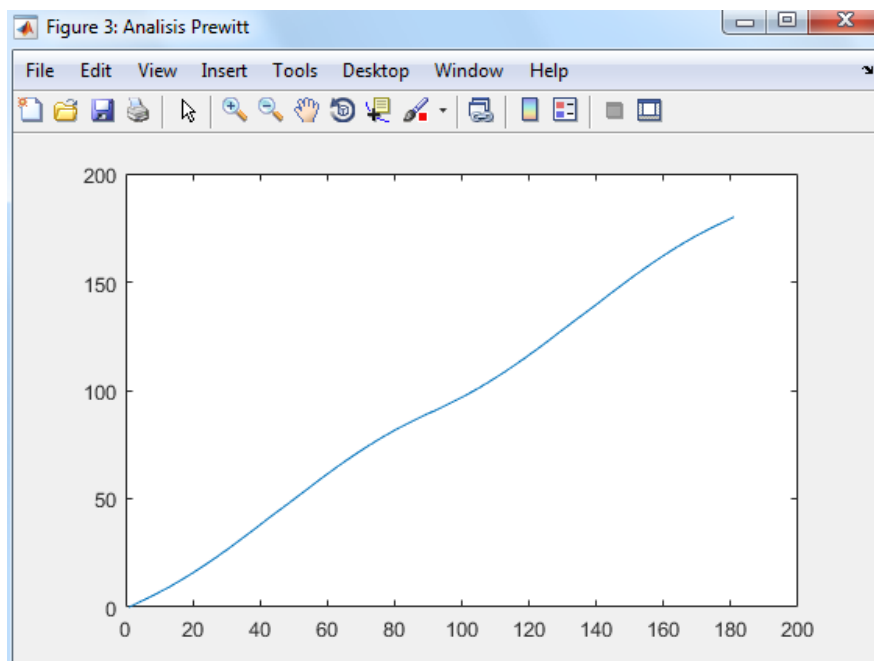


Figura 2: Funció beta per Prewitt

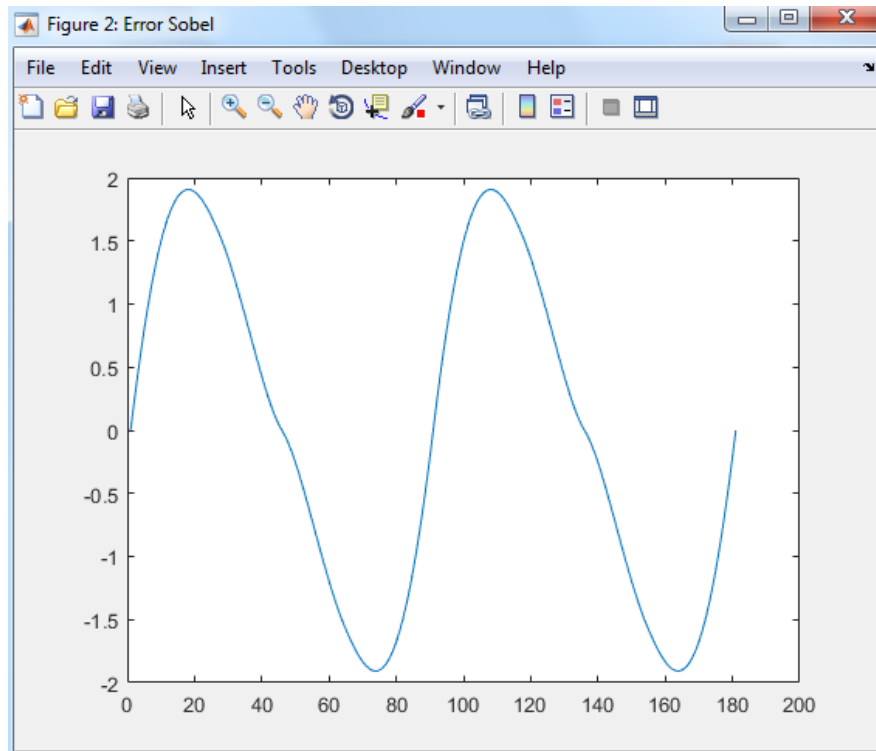


Figura 3: Funció error per Sobel

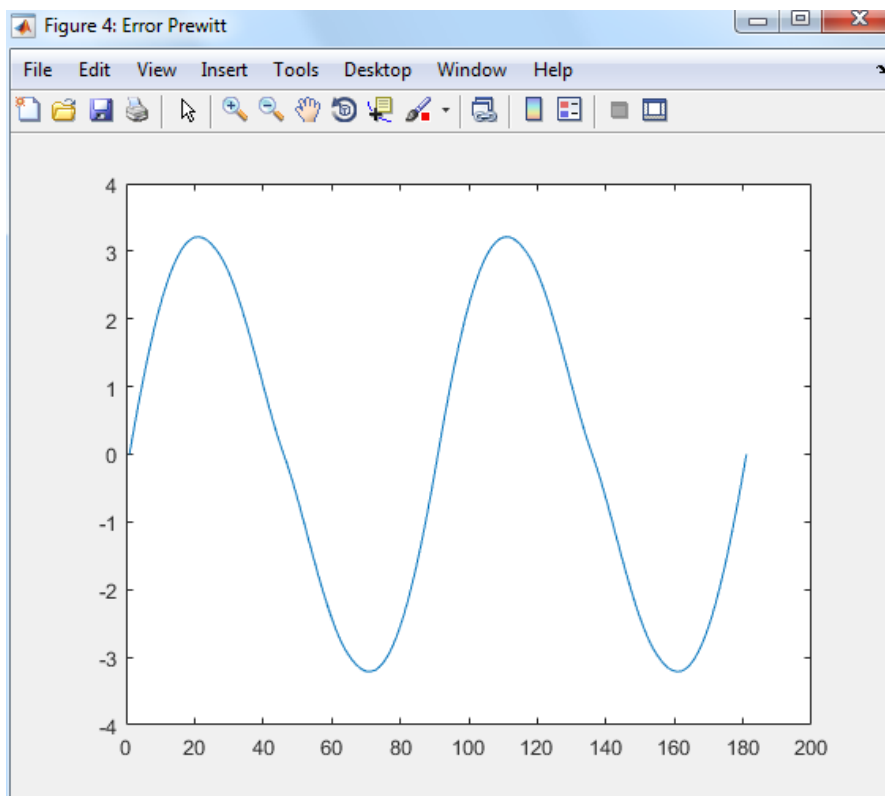


Figura 4: Funció error per Prewitt

Podem veure que la diferencia entre un i l'altre es gairebé nul·la, però podríem dir que Sobel es millor ja que té un rang d'error menor a Prewitt.