

Visió per Computador
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili

Detecció de contorns

Josep Bello Curto



18-04-2022

Índex

| | | |
|----------|--|----------|
| 1 | Exercici 1: Identificació de models de contorns | 1 |
| 1.1 | Respostes | 1 |
| 2 | Exercici 2: Aplicant detectors de contorns | 2 |
| 2.1 | Respostes | 2 |
| 2.2 | Codi | 3 |
| 3 | Exercici 3: Estudi de l'efecte de suavitzat sobre la imatge de contorns | 4 |
| 3.1 | Respostes | 4 |
| 3.2 | Codi | 5 |
| 4 | Exercici 4: Comparació entre mètodes d'obtenció de línies de contorns | 6 |
| 4.1 | Respostes | 6 |
| 4.2 | Codi | 8 |

1. Exercici 1: Identificació de models de contorns

1.1 Respostes

Els contorns de les imatges del paquet `imatges_contorns` són:

- `camisas_bn`: Roof edge
- `d95`: Ramp edge
- `d95a2`: Step edge
- `frase_bn`: Ramp edge

2. Exercici 2: Aplicant detectors de contorns

2.1 Respostes

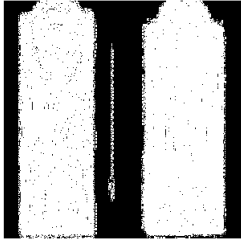


Figura 2.1: Contorns camisas_bn amb operador sobel



Figura 2.2: Contorns frase amb operador sobel



Figura 2.3: Contorns d95 amb operador sobel

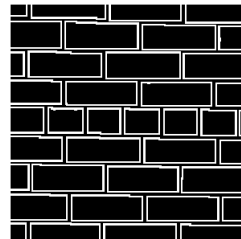


Figura 2.4: Contorns d95a2 amb operador sobel

Podem veure com la imatge a la qual millor hem detectat els contorns es la figura 2.4, la imatge d95a2 que és tracta d'uns contorns ramp edge i que a més no te soroll.

Per les imatges camisas_bn i frase, figures 2.1 i 2.2 respectivament, queden prou bé, encara que a les dues seria necessari un pre processat per eliminar el soroll, que sobretot a frase es nota molt el soroll sal i pebre.

Finalment a d95 no podem detectar res, ja que necessitaria un preprocessat per elevar el contrast i traure soroll.

2.2 Codi

```
1 d95a2 = imread('..\in_img\Imatges_contorns/d95a2.bmp');
2 camisas_bn = imread('..\in_img\Imatges_contorns/camisas_bn.bmp');
3 d95 = imread('..\in_img\Imatges_contorns/d95.bmp');
4 frase_bn = imread('..\in_img\Imatges_contorns/frase_bn.bmp');
5
6 detect_edges(d95a2, '..\out_img\ex2\d95a2_edges.bmp');
7
8 detect_edges(camisas_bn, '..\out_img\ex2/camisasbn_edges.bmp');
9
10 detect_edges(d95, '..\out_img\ex2\d95_edges.bmp');
11
12 detect_edges(frase_bn, '..\out_img\ex2/frase_edges.bmp');
13
14 function dst = detect_edges(src, dst_path)
15     if numel(size(src))>=3
16         X = sprintf('Img is RGB');
17         disp(X);
18         src = rgb2gray(src);
19     end
20
21     x_kernel = [-1 0 1; -2 0 2; -1 0 1];
22     y_kernel = [-1 -2 -1; 0 0 0; 1 2 1];
23
24     x_gradient = conv2(src, x_kernel);
25     y_gradient = conv2(src, y_kernel);
26
27     dst = abs(x_gradient) + abs(y_gradient);
28     imwrite(dst, dst_path);
29 end
```

3. Exercici 3: Estudi de l'efecte de suavitzat sobre la imatge de contorns

3.1 Respostes

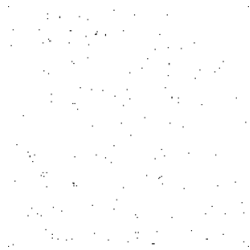


Figura 3.1: Contorns d95 amb filtre mitjana 3x3



Figura 3.2: Contorns d95 amb filtre gaussià 3x3

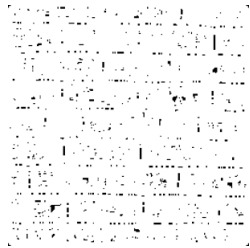


Figura 3.3: Contorns d95 amb filtre mitjana 7x7



Figura 3.4: Contorns d95 amb filtre gaussià 7x7

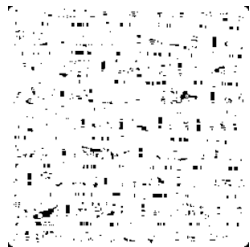


Figura 3.5: Contorns d95 amb filtre mitjana 11x11



Figura 3.6: Contorns d95 amb filtre gaussià 11x11

Podem observar com aplicant els filtres comencem a detectar algunes coses, i que el millor filtre que funciona per detectar els contorns en aquest cas és el de la mitjana per una mida de 7x7.

3.2 Codi

```
1 d95 = imread(' ../in_img/Imatges_contorns/d95.bmp');
2
3 % 3.1 3x3 filter
4 d95_mean_3 = mean_filter(d95, 3);
5 detect_edges(d95_mean_3, ' ../out_img/ex3/d95_mean_3x3.bmp');
6
7 d95_gaussian_3 = gaussian_filter(d95, 3);
8 detect_edges(d95_gaussian_3, ' ../out_img/ex3/d95_gaussian_3x3.bmp');
9
10 % 3.2 7x7 and 11x11 filters
11 d95_mean_7 = mean_filter(d95, 7);
12 detect_edges(d95_mean_7, ' ../out_img/ex3/d95_mean_7x7.bmp');
13
14 d95_gaussian_7 = gaussian_filter(d95, 7);
15 detect_edges(d95_gaussian_7, ' ../out_img/ex3/d95_gaussian_7x7.bmp');
16
17
18 d95_mean_11 = mean_filter(d95, 11);
19 detect_edges(d95_mean_11, ' ../out_img/ex3/d95_mean_11x11.bmp');
20
21 d95_gaussian_11 = gaussian_filter(d95, 11);
22 detect_edges(d95_gaussian_11, ' ../out_img/ex3/d95_gaussian_11x11.bmp');
23
24 function dst = detect_edges(src, dst_path)
25     if numel(size(src))>=3
26         X = sprintf('Img is RGB');
27         disp(X);
28         src = rgb2gray(src);
29     end
30
31     x_kernel = [-1 0 1; -2 0 2; -1 0 1];
32     y_kernel = [-1 -2 -1; 0 0 0; 1 2 1];
33
34     x_gradient = conv2(src, x_kernel);
35     y_gradient = conv2(src, y_kernel);
36
37     dst = abs(x_gradient) + abs(y_gradient);
38     imwrite(dst, dst_path);
39 end
40
41 function dst = mean_filter(src, size)
42     dst = medfilt2(src, [size, size]);
43     %figure, imshow(dst);
44     %title(size)
45 end
46
47 function dst = gaussian_filter(src, window_size)
48     dst = imgaussfilt(src, "FilterSize", window_size);
49     %figure, imshow(dst);
50     %title(window_size)
51 end
```

4. Exercici 4: Comparació entre mètodes d'obtenció de línies de contorns

4.1 Respostes

A nivell teòric el filtre de canny a més del càlcul del gradient i thresholding fa una non-maximum Suppression, per tant, computacionalment ja és més costós.

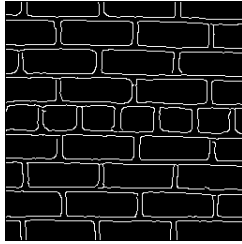


Figura 4.1: Contorns d95 canny filter

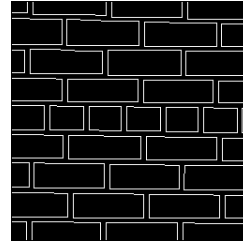


Figura 4.2: Contorns d95a2 canny filter



Figura 4.3: Contorns lena low canny filter



Figura 4.4: Contorns lena low bn canny filter

Podem observar com amb el filtre de canny i un valor de threshold adequat detectem per totes les imatges contorns, per tant, aquest filtre pareix el més adequat.

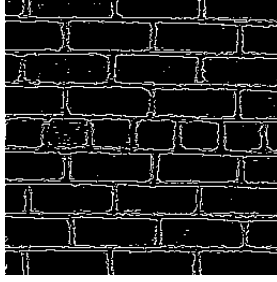


Figura 4.5: Contorns d95 Roberts filter

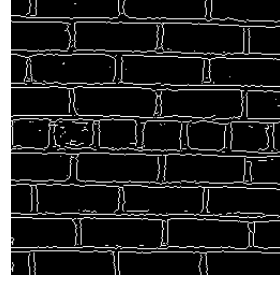


Figura 4.6: Contorns d95 Prewitt filter

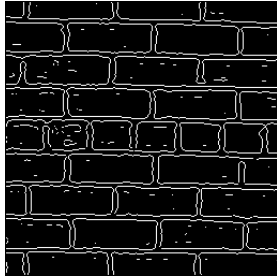


Figura 4.7: Contorns d95 LoG filter

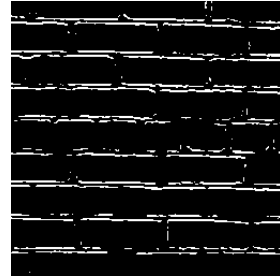


Figura 4.8: Contorns d95 nothinning

Per qualsevol dels 3 filtres, Roberts, Prewitt i LoG podem observar com, en aquest cas, és més difícil obtenir una imatge amb els contorns definits i eliminar el soroll, almenys sense fer un preprocessat.

Finalment per l'opció nothinning on no hi ha thresholding podem veure com també els contorns quedem quasi eliminats.

4.2 Codi

```
1 d95a2 = imread('.../in_img/Imatges2/d95a2.tif');
2 d95 = imread('.../in_img/Imatges2/d95.bmp');
3 lena_low = imread('.../in_img/Imatges2/lena_low.png');
4 lena_low_bn = imread('.../in_img/Imatges2/Lenna_bn.png');
5
6 % 4.1 Canny
7 detect_edges(d95, '.../out_img/ex4/d95_canny.png', 'canny', 0.5);
8 detect_edges(d95a2, '.../out_img/ex4/d95a2_canny.png', 'canny', 0.2);
9 detect_edges(lena_low, '.../out_img/ex4/lena_low_canny.png', 'canny', 0.1);
10 detect_edges(lena_low_bn, '.../out_img/ex4/lena_low_bn_canny.png', 'canny', 0.2);
11
12 % 4.2 Roberts, Prewitt, LoG, nothinning
13 detect_edges(d95, '.../out_img/ex4/d95_roberts.png', 'Roberts', 0.06);
14 detect_edges(d95, '.../out_img/ex4/d95_prewitt.png', 'Prewitt', 0.05);
15 detect_edges(d95, '.../out_img/ex4/d95_log.png', 'log', 0.005);
16
17 detect_edges_nothinning(d95, '.../out_img/ex4/d95_nothinning.png');
18
19 function dst = detect_edges(src, dst_path, f, threshold)
20     if numel(size(src))>=3
21         X = sprintf('Img is RGB');
22         disp(X);
23         src = rgb2gray(src);
24     end
25
26     dst = edge(src, f, threshold);
27     figure, imshow(dst);
28     imwrite(dst, dst_path);
29 end
30
31 function dst = detect_edges_nothinning(src, dst_path)
32     if numel(size(src))>=3
33         X = sprintf('Img is RGB');
34         disp(X);
35         src = rgb2gray(src);
36     end
37
38     dst = edge(src, 'nothinning');
39     figure, imshow(dst);
40     imwrite(dst, dst_path);
41 end
```

Índex de figures

| | | |
|-----|--|---|
| 2.1 | Contorns camisas_bn amb operador sobel | 2 |
| 2.2 | Contorns frase amb operador sobel | 2 |
| 2.3 | Contorns d95 amb operador sobel | 2 |
| 2.4 | Contorns d95a2 amb operador sobel | 2 |
| 3.1 | Contorns d95 amb filtre mitjana 3x3 | 4 |
| 3.2 | Contorns d95 amb filtre gaussià 3x3 | 4 |
| 3.3 | Contorns d95 amb filtre mitjana 7x7 | 4 |
| 3.4 | Contorns d95 amb filtre gaussià 7x7 | 4 |
| 3.5 | Contorns d95 amb filtre mitjana 11x11 | 4 |
| 3.6 | Contorns d95 amb filtre gaussià 11x11 | 4 |
| 4.1 | Contorns d95 canny filter | 6 |
| 4.2 | Contorns d95a2 canny filter | 6 |
| 4.3 | Contorns lena low canny filter | 6 |
| 4.4 | Contorns lena low bn canny filter | 6 |
| 4.5 | Contorns d95 Roberts filter | 7 |
| 4.6 | Contorns d95 Prewitt filter | 7 |
| 4.7 | Contorns d95 LoG filter | 7 |
| 4.8 | Contorns d95 nothinning | 7 |