

## TD3: Local filtering, gradient and Corner detection

In this TP, we focus on image filtering by local convolution masks. We illustrate these local operations with contour detection.

### Exercise 1 Smoothing

1. Write a small C program to filter an input PGM image with the following binomial filter:

$$b_{2,2} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

A technical problem to solve would be to deal with image boundaries: what can we do if the convolution mask support is not completely included in the image support. Among generic solutions (in the sense that we don't want to change the mask support or weights), you could:

- Fold the image boundaries: e.g. if  $N$  is the last column, " $N+1$ " would be the column  $N-1$ .
  - Make the image periodic: coordinates are given modulo  $N+1$
2. Consider now a 1D version of such binomial filter:  $b_x = \frac{1}{4}[1, 2, 1]^T$  and  $b_y = b_x^T$ . First, verify that that  $b_{2,2} = b_x \cdot b_y$  (separability principle).
  3. Implement the filtering of a PGM image using this separable filter.

**Bonus** Implement a function that constructs a 1D separable Gaussian filter of size  $k$  based on binomial coefficients.

### Exercise 2 Gradient Map

1. Implement the Sobel filter (cf lecture notes<sup>1</sup>), to estimate partial derivative along the  $x$  and  $y$  axis.
2. Display the gradient vector field as:
  - A scalar map of the gradient norm
  - A scalar map of the gradient orientation (rescale the angles in degree to  $[0, 255]$ )

**Bonus** Implement the 1D Gaussian derivative filter (cf lecture notes) and compare the maps with the results of Sobel filters.

---

<sup>1</sup><http://liris.cnrs.fr/david.coeurjolly/cours/ENS2012/html/c-ip-analyse.html>

### Exercise 3 Harris detector

In this exercise, we focus on corner detection instead of contour detection as discussed in the lecture. The idea behind Harris detector is to detect from partial derivative information local edges or corners. The idea is to define an energy function whose maxima should correspond to corners.

1. First, let us consider a basic detector so called Moravec detector. Let  $W = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$  be a smoothing convolution mask and  $V(x, y)$  be neighbourhood around pixel  $(x, y)$  (only in the 4 direct neighbours). We define

$$E_{x,y} = \sum_{(u,v) \in V(x,y)} W_{u,v} |I_{u+x,y+v} - I_{u,v}|^2 \quad (1)$$

where  $I_{u,v}$  is the image value at  $(u, v)$ .

**Implement such Moravec detector and threshold the map  $E$  to spot some corners. For example, you can output a PPM image with grayscale map for input image pixels and red crosses for corners detected by thresholding the map  $E/E_{max}$ .**

2. A better approach (less sensitive to noise) consist in doing a Taylor expansion of the differential term in the energy:

$$E_{x,y} = \sum_{u,v} W_{u,v} (x.dX + y.dY + O(x^2, y^2))^2 \quad (2)$$

where derivative are approximated for example by Sobel filters<sup>2</sup>:

$$dX = I * (-1, 0, 1) \quad (3)$$

$$dY = I * (-1, 0, 1)^t \quad (4)$$

Hence, in a small neighborhood, we have

$$E_{x,y} = (x, y)M(x, y)^t$$

where

$$M = \begin{pmatrix} A & C \\ C & B \end{pmatrix} \quad (5)$$

$$A = dX^2 * W \quad (6)$$

$$B = dY^2 * W \quad (7)$$

$$C = (dX.dY) * W \quad (8)$$

Let  $\alpha$  and  $\beta$  be the eigenvalues of  $M$ . According to such values, we can classify the point  $(x, y)$  into edge, corner or flat regions as depicted in Figure 1.

---

<sup>2</sup>keep in mind that '\*' refers the convolution product.

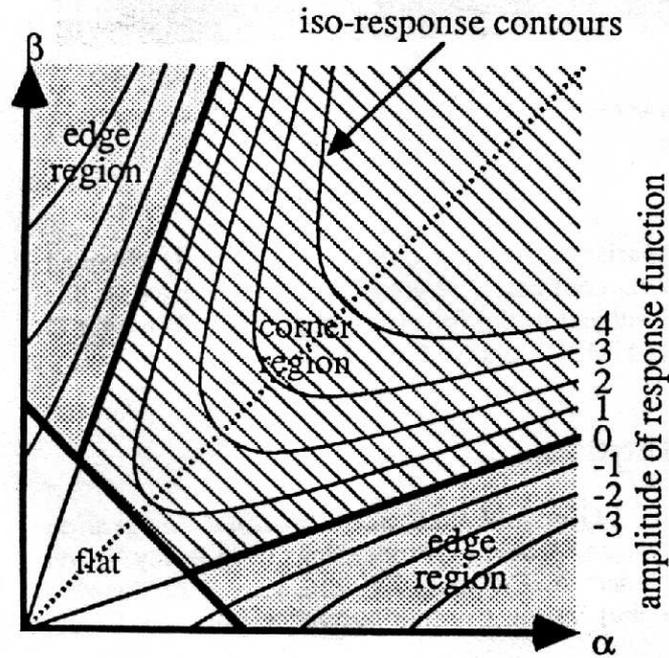


Figure 1: Nature the point as a function of  $\alpha$  and  $\beta$ .

We can simply define a single scalar value (parametrized by  $k$ , see below).

$$R = \det(M) - k \cdot \text{tr}^2(M) \quad (9)$$

$$= AB - C^2 - k(A + B)^2 \quad (10)$$

Now, corners will be defined as local maxima in the  $R$  map.

- (a) **From previous exercises, compute the maps  $A$ ,  $B$  and  $C$ .**
- (b) **Select corner points such that:  $k = 0.04$  and you simply apply a threshold on the normalized values  $R/R_{max} \geq \sigma$**
- (c) **Display the corners as red crosses superimposed on the original image.**

**BONUS:**

- What is the influence of the  $k$  parameter ?
- Add Gaussian and derivative of Gaussian filter instead of  $W$  and Sobel filters. Does it enhance the results ?
- Do some isotropy comparisons with Moravec: rotate your image a little bit, compute the corners using both approaches and compare.