Image as a signal

Luc Brun

January 25, 2018

Introduction

Smoothing

Edge detection

Fourier Transform

Different way to "see" an image

- A stochastic process,
- A random vector $(I[0,0], I[0,1], \ldots, I[n,m])$ if the image I is an $n \star m$ array.
- **A 3D surface (for greyscale images).**
- A set of data connected by geometrical/topological constraints..
- **The sampling of a continuous signal.**
- ...

Sampling

An image is considered as the sampling of a continuous signal:

$$I = Q \circ f$$

- I is the discrete image
- f the continuous signal ($f \in C^2(\mathbb{R}^2, \mathbb{R})$: greyscale images),
- Q a sampling operator.

Convolution

- Combination of an image f with an other signal in order to attenuate or reinforce some properties.

$$\begin{aligned} f * g(x,y) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(u,v)g(x-u,y-v)dudv \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-u,y-v)g(u,v)dudv \end{aligned}$$

- Discrete version based on discrete masks:

$$M * I(i,j) = \sum_{k=-p}^{p}\sum_{l=-q}^{q} M[k][l]I[i-k][j-l]$$

Convolution

- Let us suppose that g is equal to $\frac{1}{R^2}$ on a square of side R and null everwhere else.

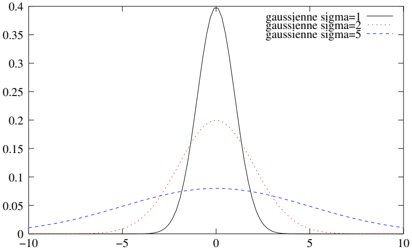
$$f * g(x,y) \;=\; \frac{1}{R^2}\int_{-R}^{+R}\int_{-R}^{+R} f(x-u,y-v)dudv$$

  Any valeur of f is replaced by its mean on the square.
- Let us suppose that the square C is divided into two halves C^+ and C^- with $g(u,v) = \frac{1}{R^2}$ over C^+ and $-\frac{1}{R^2}$ over C^-.

$$\begin{aligned} f * g(x,y) = \;& \frac{1}{R^2}\int\int_{C^+} f(x-u,y-v)dudv - \\ & \frac{1}{R^2}\int\int_{C^-} f(x-u,y-v)dudv \end{aligned}$$

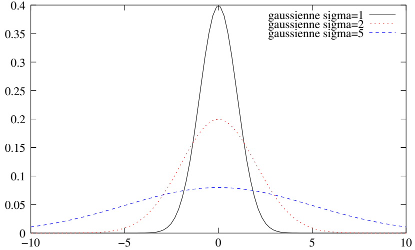
Gaussian function

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$



Parameter σ determine the flatness of the Gaussian and allows to control the strength of the filter. A high value of σ induces a strong smoothing and conversely.

Gaussian function



For practical reasons, the convolution of a signal with a Gaussian is performed by restricting it to a finite support $[-M_\epsilon, M_\epsilon]$ with:

$$\forall x \in [-M_\epsilon, M_\epsilon] \quad G(x) > \epsilon$$

M_ϵ : increasing function of σ
$\rightarrow$ The more we wish to smooth, the more we have to increase the support

Gaussian function

$$G(x,y) = \frac{1}{\sigma^2(2\pi)} e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} = G(x)G(y)$$

Convolution of the 2D function f with G provides thus:

$$\begin{aligned} f * G(x,y) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-u, y-v)G(u,v)dudv \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-u, y-v)G(u)G(v)dudv \\ &= \int_{-\infty}^{+\infty} G(u)\left(\int_{-\infty}^{+\infty} f(x-u, y-v)G(v)dv\right)du \\ &= \int_{-\infty}^{+\infty} G(u)\,(f *_y G)\,(x-u, y)du \end{aligned}$$

$$f * G(x,y) = (G *_x (f *_y G))\,(x,y)$$

where $*_x$ et $*_y$ denote convolutions according to variables x and y.

Gaussian function

- We have thus:

$$f * G(x,y) = \left(G *_x (f *_y G)\right)(x,y)$$

- We use two 1D masks of size $[-M_\epsilon, M_\epsilon]$ rather than one 2D mask of size $[-M_\epsilon, M_\epsilon]^2$
  $\rightarrow$ complexity $\mathcal{O}(2|I||M|)$ instead of $\mathcal{O}(|I||M|^2)$.
  - $|I|$: Image's size
  - $|M|$: mask's size

Gaussian

$\sigma = 2$

$\sigma = 4$

$\sigma = 8$

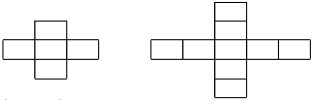
$\sigma = 16$

Gaussian

- Gaussian filtering is based on a linear combination of the initial values.
- $\rightarrow$ One initial impulsion modifies the filtered signal in any case.
- $\rightarrow$ boundaries are smoothed.

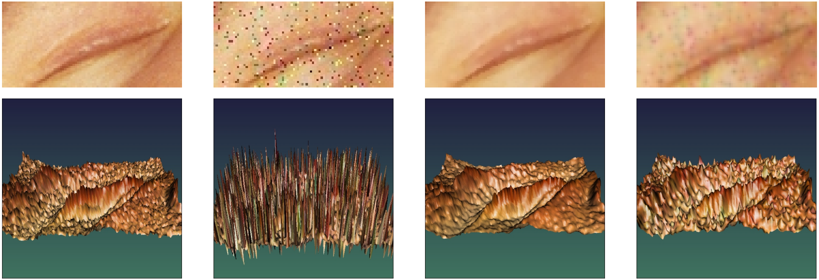
Median

- Map each pixel to the median value of the pixels contained in a given neighborhood.
  1. Define a neighborhood



  2. sort values in the neighborhood
  3. Map the median value to the central pixel
- Properties:
  1. Multiplicative law: $M[af] = aM[f]$,
  2. Filter weakly linear
  3. Efficient for impulsive noise
  4. Preserve boundaries,
  5. Remove extreme values $\rightarrow$ sparse non significant values have no influence.
  6. Can not be decomposed into a sequence of filters $M_x \circ M_y(f) \neq M(f)$.

Example 1



Original | Salt &Pepper | Median | Gaussian

Example 2

$size = 3$

$size = 5$

$size = 7$

$size = 9$

Edge detection

- Sudden change in the signal correspond to high values of its derivative
- In the 2D case the derivative corresponds to a differential:

$$Df(p).\vec{n} = \frac{\partial f}{\partial x}(p).n_x + \frac{\partial f}{\partial y}(p).n_y$$

$$= \lim_{h \to 0} \frac{f(p) - f(p + (hn_x, hn_y))}{h}$$

- Only the direction $\vec{n}$ matters $\rightarrow \|\vec{n}\| = 1$

Use of the gradient

$$\triangledown f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

- We have then: $Df(p).\vec{n} = \triangledown f(p) \bullet \vec{n}$
- Moreover:

$$\max_{\|\vec{n}\|=1} |Df(p).\vec{n}| = \max_{\|\vec{n}\|=1} |\triangledown f(p) \bullet \vec{n}| = \|\triangledown f(p)\|$$

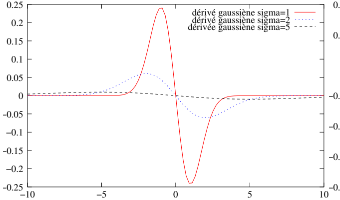
- The norm of the gradient provides the maximal variation of the differential. This maximum is reached for $\vec{n}$ colinear with $\triangledown f(p) \rightarrow$. The gradient's direction provides the direction of greatest variation of the function f.

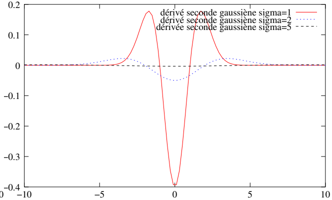
Edge detection

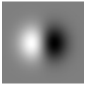
$$(f * g)^{(k)} = (f^{(k)}) * g = f * (g^{(k)})$$

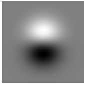
- Instead of smoothing f and then computing its gradient, we perform a convolution with the derivative of a Gaussian. (pre computed).
- In the same way, the computation of the Laplacian is performed by convoluing f with the Laplacian of a Gaussian function.

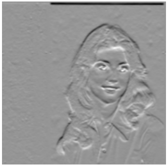
Edge Detection

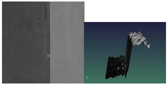


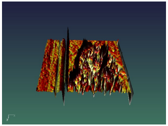


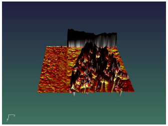


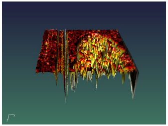






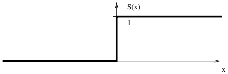






Optimal operators

- Modeling of an edge : $C(x) = S_0 S(x) + B(x)$



The optimal operator h convolved with C must have a maximum in 0 and must allow[Canny86]:

1. A good detection
2. A good localisation
3. a weak multiplicity of maxima induced by noise

$$\rightarrow 2h(x) - 2\lambda_1 h^{(2)}(x) + 2\lambda_2 h^{(3)}(x) + \lambda_3 = 0$$

Optimal operators

- Canny 86 : Finite support constraint

$$h(0) = 0 \quad ; \quad h(M) = 0 \quad ; \quad h'(0) = S \quad ; \quad h'(M) = 0$$

$\rightarrow$ Complex function to encode

- Deriche 87: Same equation but infinite support

$$h(0) = 0 \quad ; \quad h(+\infty) = 0 \quad ; \quad h'(0) = S \quad ; \quad h'(+\infty) = 0$$

$$\Rightarrow h(x) = ce^{-\alpha|x|} sin\omega x \underset{\omega \approx 0}{=} c\omega x e^{-\alpha|x|} = Cxe^{-\alpha|x|}$$

- Lissage:

$$I(x) = \int_0^x h(x)dx = b(\alpha|x| + 1)e^{-\alpha|x|}$$

α control the smoothing (same role than σ)

Optimal operators

- Use of the z transform $\rightarrow$ recursive implementation. Example : convolution of $x(i)$ with $l(i)$ (result in y)

$$
\begin{array}{ll}
\textit{Pour} & m = 1, \ldots, M \\
 & y^{+}(m) = a_0 x(m) + a_1 x(m-1) - b_1 y^{+}(m-1) - b_2 y^{+}(m-2) \\
\textit{Pour} & m = M, \ldots, 1 \\
 & y^{-}(m) = a_2 x(m+1) + a_3 x(m+2) - b_1 y^{-}(m+1) - b_2 y^{-}(m+2) \\
\textit{Pour} & m = 1, \ldots, M \\
 & y(m) = y^{+}(m) + y^{-}(m)
\end{array}
$$

  M : signal's size. Coefficients a_i and b_i deduced from parameter α. 8 operations for any α.
- Advantages of Deriche's filter:
  1. Formalisation of the notion of contour
  2. Computational times independent of α.
  3. No cutting of the filter

Optimal operators

original

$\alpha = 1$

$\alpha = 5$

$\alpha = 10$

Use of the second differential

- A high gradient does not obviously corresponds to a contour and conversely
- Detection of gradient''s local maxima: : zeros of the second differential.

$$D_2 f(p).\vec{n} = \frac{\partial^2 f}{\partial x^2}(p).n_x.n_x + \frac{\partial^2 f}{\partial y^2}(p).n_y.n_y + 2\frac{\partial^2 f}{\partial x \partial y}(p).n_x n_y$$

- Extrema of $Df(p).\vec{n}$ at $\vec{n} \Rightarrow$ zeros of $D_2 f(p).\vec{n}$ at $\vec{n}$.

Use of the Laplacian

$$\triangle f(p) = \frac{\partial^2 f}{\partial x^2}(p) + \frac{\partial^2 f}{\partial y^2}(p)$$

- Invariant by rotation$\rightarrow$ n does not play any role.
- Using the Laplacian hence avoid to compute the gradient additionally to the second differential
- Beware!!
  $\triangle f(p) = 0 \not\Rightarrow \exists \vec{n} \ / \ D_2 f(p).\vec{n} = 0$
- Coincide only if (Marr 1980): variations of intensity are linear on the zero crossing line and on lines parallel to it in a neighborhood of p.

Laplacian

- Zeros of the Laplacian are determined by its sign change.

Color Gradient

$$f\left(\begin{array}{ccc} \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (x,y) & \mapsto & (f_1(x,y), f_2(x,y), f_3(x,y)) \end{array}\right.$$

- Differential of f in p:

$$Df(p).(n_x, n_y) = \left(\begin{array}{c} \frac{\partial f_1}{\partial x}(p).nx + \frac{\partial f_1}{\partial y}(p).n_y \\ \frac{\partial f_2}{\partial x}(p).nx + \frac{\partial f_2}{\partial y}(p).n_y \\ \frac{\partial f_3}{\partial x}(p).nx + \frac{\partial f_3}{\partial y}(p).n_y \end{array}\right)$$

- Squared norm of the differential:

$$S(p, \vec{n}) = \|Df(p).(n_x, n_y)\|^2 = E{n_x}^2 + 2Fn_x n_y + G{n_y}^2$$
$$E = \sum_{i=1}^{3} \left(\frac{\partial f_i}{\partial x}\right)^2 \quad F = \sum_{i=1}^{3} \frac{\partial f_i}{\partial x}\frac{\partial f_i}{\partial y} \quad G = \sum_{i=1}^{3} \left(\frac{\partial f_i}{\partial y}\right)^2$$

Color Gradient

$$S(p,\vec{n}) = E{n_x}^2 + 2Fn_xn_y + G{n_y}^2$$

If $\vec{n} = (cos(\theta), \sin(\theta))$, maximum of $S(p,\vec{n})$ for:

$$\theta_0 = \frac{1}{2}arctan\left(\frac{2F}{E-G}\right)$$

Associated Maximum
$\lambda(x,y) =_{not} S(p,(cos(\theta_0), sin(\theta_0)))$:

$$\lambda(x,y) = \frac{E+G+\sqrt{(E-G)^2+4F^2}}{2}$$

λ equal to the squared norm of the gradient in the case of a mono dimensional signal.

Color Laplacian

- Differential of $S(p, \vec{n})$:

$$\begin{aligned} D_S(p).\vec{n} = \quad & E_x(p){n_x}^3 + (2F_x(p) + E_y(p))n_x^2 n_y \\ & +(G_x(p) + 2F_y(p))n_x n_y^2 + G_y(p)n_y^3 \end{aligned}$$

  E_x, E_y,F_x, F_y et G_x, G_y : partial derivatives of E, F and G according to x and y.

- $D_S(p).\vec{n} \approx$ Laplacian

References:
Zenzo 86, Cumani 89, 91

Determination of contours

1. Compute the gradient in each point
2. Compute the image of gradient's norm
3. Extract the local maxima in the direction of the gradient
4. Perform an hysteresis thresholding of the image of maxima. locaux.

Hysteresis thresholding only preserve:

1. Points whose norm is greater than a high threshold (sh)
2. Points whose norm is greater than a low threshold (sb with $sb < sh$) and belonging to a contour having at least one point whose norm is greater than sh.

Determination of contours

- Compute $D_2 f(p).\vec{n}$ for any point p of the image ($\vec{n} = \frac{\nabla f}{\|\nabla f\|}$)
- Search for zero crossings of $D_2 f(p).\vec{n}$ in the direction $\vec{n}$.
- Create the images of zeros crossings and grandient's norm.
- Perform an hysteresis thresholding on the image of local maxima.

Determination of contours

- Compute the Laplacian.
- Search for zero crossings
- Create the images of zeros crossings and grandient's norm.
- Perform an hysteresis thresholding on the image of local maxima.

Fourier Transform

- Allows to switch from the spatial domain to the frequency domain
- For any integrable function:

$$F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j(ux+vy)}dxdy$$

- Inverse transform:

$$f(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)e^{j(ux+vy)}dudv$$

Fourier Transform

	Function	Fourier Transform
Linearity	$af(x,y) + bg(x,y)$	$aF(u,v) + bG(u,v)$
Contraction	$f(ax, by)$	$\frac{1}{\lvert a \rvert \lvert b \rvert} F(\frac{u}{a}, \frac{u}{b})$
Translation	$f(x - x_0, y - y_0)$	$e^{-2j\pi(ux_0 + vy_0)} F(u,v)$
Convolution	$f * g(x,y)$	$F(u,v)G(u,v)$
Separability	$f(x)g(y)$	$F(u).F(v)$
Rotation	θ	$-\theta$

Fourier Transform

- Low frequencies near the origin encode flat zones of the image
- High frequencies encode abrupt changes (textures/contours),
- This switch of domain of representation allows to transform global properties (or operations) into local ones.

Fourier Transform