Digital Image and Video Processing

Marc Van Droogenbroeck

INTELSIG, Montefiore Institute, University of Liège, Belgium

September 2014

Practical informations

- Instructor: Marc Van Droogenbroeck
- Assistant: Philippe Latour
- Course Web page:
  `http://www.ulg.ac.be/telecom/assistant`
- Slides:
  `http://orbi.ulg.ac.be`
- Evaluation
  - compulsory project: 4 students, evaluated in December.
  - written individual examination: 20 minutes long, after the project presentation
  - written examination (open book) if necessary

General considerations

- Linear framework $\rightarrow$ non-linear frameworks
- A world of trade-offs (computational load $\leftrightarrow$framerate, etc).
- Never forget the acquisition step
- There is no unique, universal, solution
- More and more machine learning in computer vision

Outline

Outline

Image representation and fundamentals

- Elements of visual perception
  - Colors: representation and colorspaces
  - Transparency
- Data structure for images
- Resolution
- Examples of industrial applications:
  - Segmentation
  - Optical character recognition

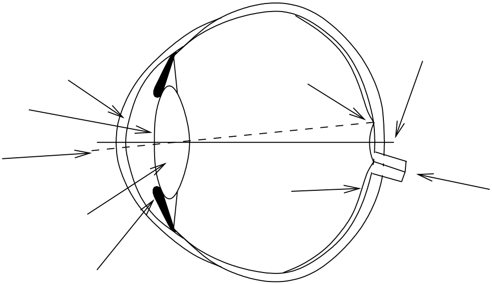


Figure : Lateral view of the eye globe (*rods* and *cones* are receptors located on the retina).

Human visual system, light and colors II

color	wavelength interval $\lambda\,[m]$	frequency interval $f\,[Hz]$
violet	$\sim$ 450–400 $[nm]$	$\sim$ 670–750 $[THz]$
blue	$\sim$ 490–450 $[nm]$	$\sim$ 610–670 $[THz]$
green	$\sim$ 560–490 $[nm]$	$\sim$ 540–610 $[THz]$
yellow	$\sim$ 590–560 $[nm]$	$\sim$ 510–540 $[THz]$
orange	$\sim$ 635–590 $[nm]$	$\sim$ 480–510 $[THz]$
red	$\sim$ 700–635 $[nm]$	$\sim$ 430–480 $[THz]$

Figure : Visible colors (remember that $\lambda = \frac{3\times10^8}{f}$).

Human visual system, light and colors III

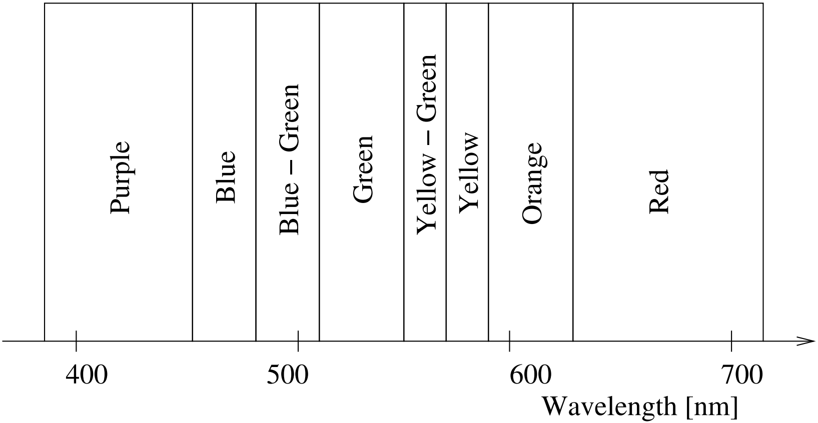


Figure : Colors on the visible spectrum.

Frequency representation of colors

$$\int_\lambda L(\lambda)\, d\lambda \tag{1}$$

Impossible from a practical perspective because this would require *one sensor for each wavelength.*

Solution: use colorspaces

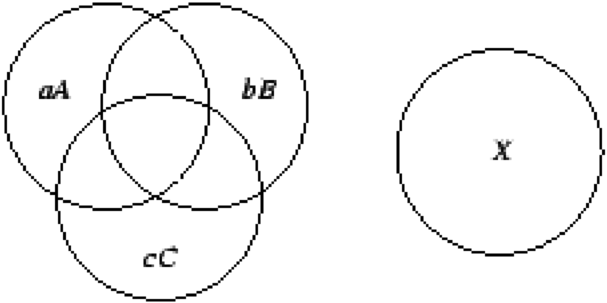


Figure : Equalization experiment of colors. The aim is to mix A, B, and C to get as close as possible to X.

Frequency representation of colors

$$\int_{\lambda} L(\lambda)\, d\lambda \tag{1}$$

Impossible from a practical perspective because this would require *one sensor for each wavelength.*
Solution: use colorspaces

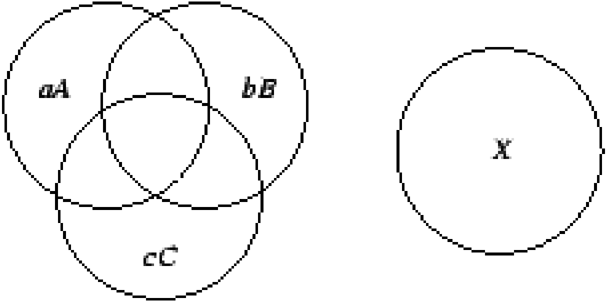


Figure : Equalization experiment of colors. The aim is to mix A, B, and C to get as close as possible to X.

The RGB additive colorspace

Three fundamental colors: red R $(700\,[nm])$, green G $(546,1\,[nm])$ and blue B $(435,8\,[nm])$,

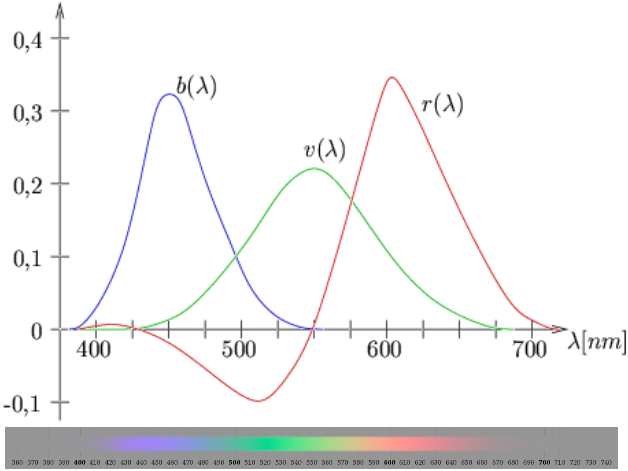
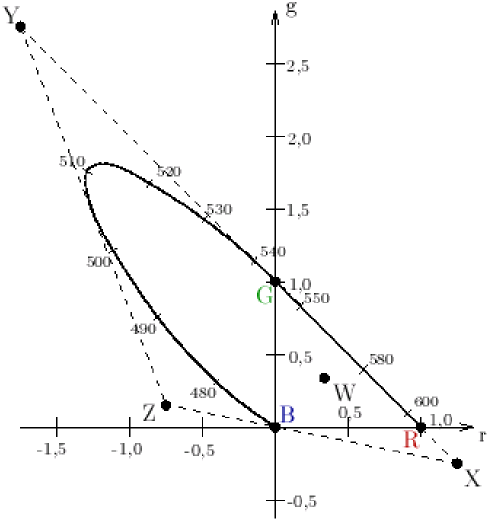


Figure : Equalization curves obtained by mixing the three fundamental colors to simulate a given color (wavelength).

CIE chromatic diagram for RGB



Notion of intensity

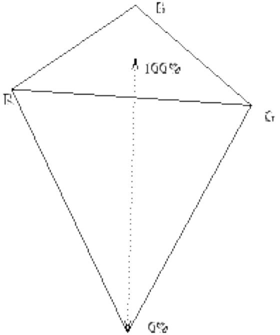


Figure : Pyramid derived from an RGB color representation.

Towards other colorspaces: the XYZ colorspace I

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2,769 & 1,7518 & 1,13 \\ 1 & 4,5907 & 0,0601 \\ 0 & 0,0565 & 5,5943 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2}$$

$$\begin{aligned} x &= \frac{X}{X+Y+Z} && (3) \\ y &= \frac{Y}{X+Y+Z} && (4) \\ z &= \frac{Z}{X+Y+Z} && (5) \\ x+y+z &= 1 && (6) \end{aligned}$$

Towards other colorspaces: the XYZ colorspace II

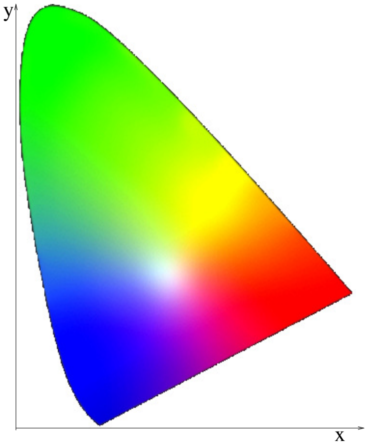


Figure : Approximative chromatic colorspace defined by two chrominance variables x and y.

Luminance

Luminance: $Y = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B$

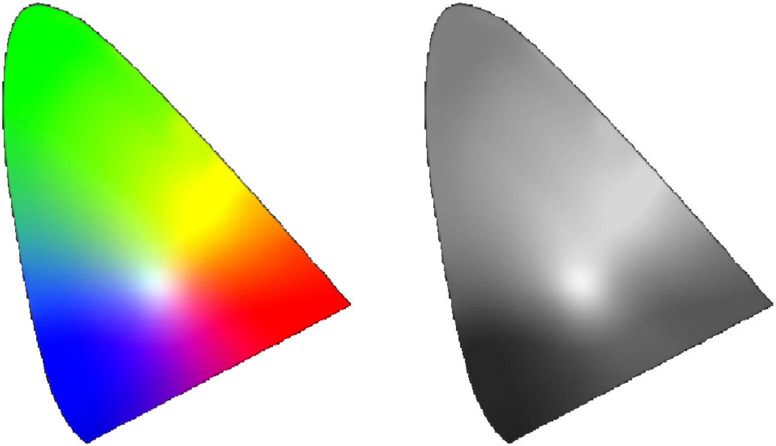


Figure : *xy* chromatic diagram and maximal luminance for each color.

Luminance + two chrominances

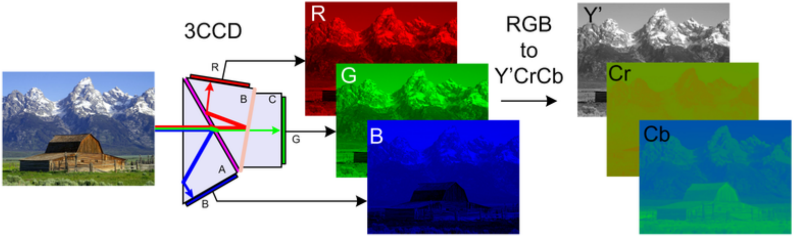


Figure : Acquisition of a $Y\,C_b\,C_R$ signal [Wikipedia]

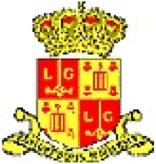
There are variations, such as the $Y\,U\,V$ colorspace, mainly developed for compression:

1. information concentration in the Y channel $\Rightarrow$ better compression.
2. better decorrelation between channels).

The HSI colorspace

Colorspace that has a better physical meaning:

- hue
- saturation
- intensity



Other colorspaces

- a subtractive colorspace: Cyan, Magenta, and Yellow (CMY)
- Luminance + chrominances (YIQ, YUV or YC_bC_r)

In practice,

Hexadecimal			R	G	B
00	00	00	0	0	0
00	00	FF	0	0	255
00	FF	00	0	255	0
00	FF	FF	0	255	255
FF	00	00	255	0	0
FF	00	FF	255	0	255
FF	FF	00	255	255	0
FF	FF	FF	255	255	255

Table : Definition of color values and conversion table between an hexadecimal and an 8-bits representation of colors.

Bayer filter I

A Bayer filter mosaic is a color filter array for arranging RGB color filters on a square grid of photo sensors.

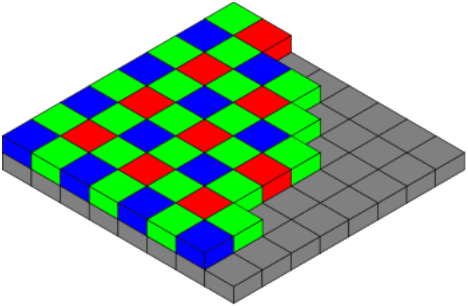


Figure : The Bayer arrangement of color filters on the pixel array of an image sensor. [Wikipedia]

Bayer filter II

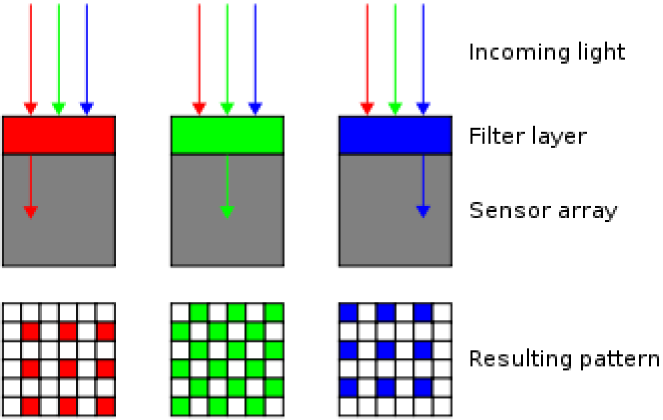


Figure : Profile/cross-section of sensor. [Wikipedia]

Bayer filter: practical considerations

- Most mono-sensor cameras use the Bayer pattern, except for professional 3CCD cameras (three sensor planes + prism to divide the incoming light)
- The filter pattern is 50% green, 25% red and 25% blue. Why?
- We only have one value per pixel. Other values are re-built by interpolation, but they might not even exist... !
- For compression or processing,
  - 1 sensor plane $\Rightarrow$ normally only one byte to process. Possible if the processing is very close to the sensor. Otherwise, there is no information about the real observed values.
  - 3 sensor planes $\Rightarrow$ 3 planes to process or to compress. Expected compression rate: $3 \times$ lower than for a single sensor.
- It might be wiser, for processing, to have a black-and-white (or a monochromatic, such as red) camera, instead of a color camera.

What defines the color of an object?

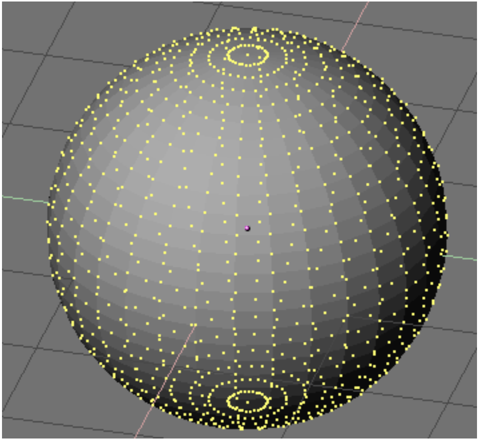


Figure : A synthetic 3D object. Shadows and surfaces with varying reflective coefficients model a 3D object.

Visual effects

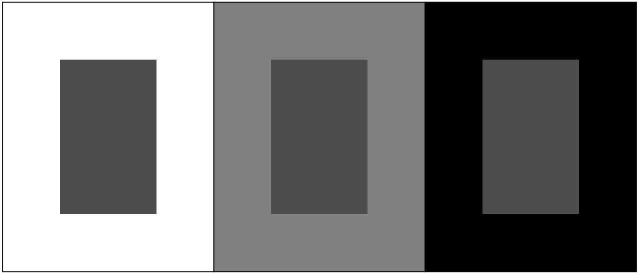


Figure : Illustration of a masking visual effect.

Transparency bits

Let

- $i(x, y)$ be the value of the image at location (x, y)
- $t(x, y)$ be the transparency (defined by 1 to 8 bits)
- $o(x, y)$ be the output value, after applying transparency

Applying transparency consists to calculate: $o(x, y) = \frac{t(x,y)}{255} i(x, y)$

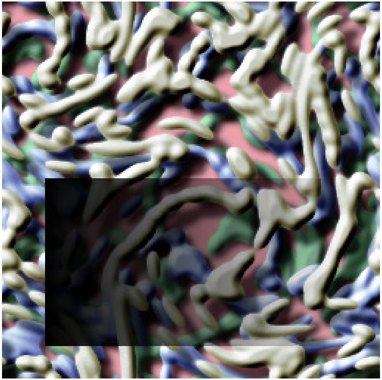


Figure : Transparency bits have been applied inside a rectangle.

Sampling grid and frame organization

- Each sample located on a grid is named a *pixel* (which stands for *picture element*).
- There are two common sampling grids and they induce certain types of connectivity.

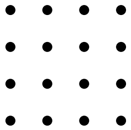
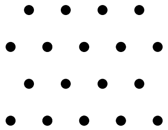
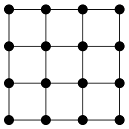
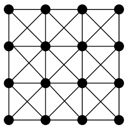
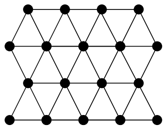
<table>
<tr><th colspan="2">Square grid</th><th>Hexagonal grid</th></tr>
<tr><td colspan="2"></td><td></td></tr>
<tr><td>4-connectivity</td><td>8-connectivity</td><td>6-connectivity</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

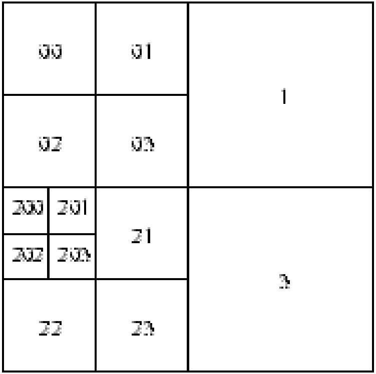
Table : Types of grid and associated connectivities.

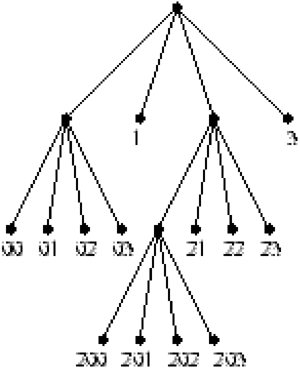
Typology of images and videos

- 2D. This types refers to a “classic” image and is usually expressed as a 2D array of values. It might represent the luminance, a color, depth, etc.
- 3D. 3D images are obtained with devices that produce 3D images (that is with x, y, z coordinates). Medical imaging devices produce this type of images.
- 2D+t. t refers to time. Therefore, $2D + t$ denotes a video composed over successive 2D images, indexed by t.
- 3D+t. $3D + \mathrm{t}$ images are in fact animated $3D$ images. A typical example is that of animated 3D graphical objects, like that produced by simulations.

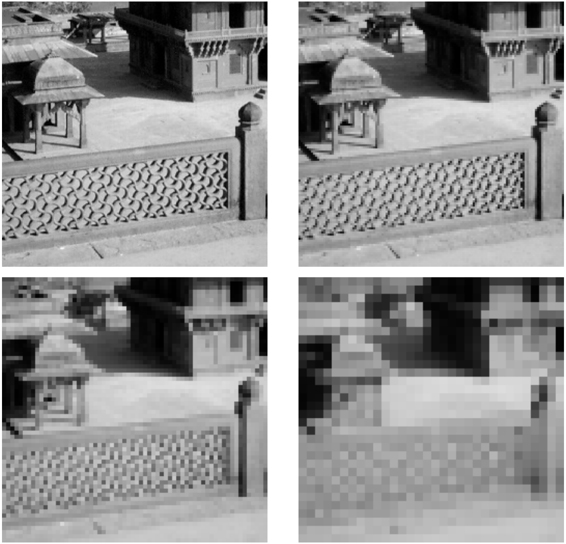
Data structure for dealing with images

- Typical data structure for representing images: matrices (or 2D tables), vectors, trees, lists, piles, ...
- A few data structures have been adapted or particularized for image processing, like the *quadtree*.





Resolution



The bitplanes of an image

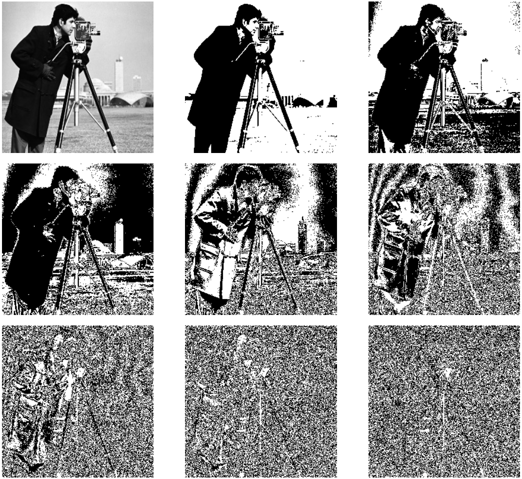


Table : An original image and its 8 bitplanes starting with the Most Significant Bitplane (MSB).

Objective quality measures and distortion measures

Let f be the original image (whose size is $N \times N$) and $\hat{f}$ be the image after some processing.

Definition

[Mean Square Error]

$$\mathrm{MSE} = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \left(f(j,k) - \hat{f}(j,k) \right)^2 \tag{7}$$

[Signal to noise ratio]

$$\mathrm{SNR} = \frac{\sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \left(f(j,k) \right)^2}{\sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \left(f(j,k) - \hat{f}(j,k) \right)^2} \tag{8}$$

[Peak signal to noise ratio]

$$\mathrm{PSNR} = \frac{N^2 \times 255^2}{\sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \left(f(j,k) - \hat{f}(j,k) \right)^2} \tag{9}$$

Images can be seen as topographic surfaces

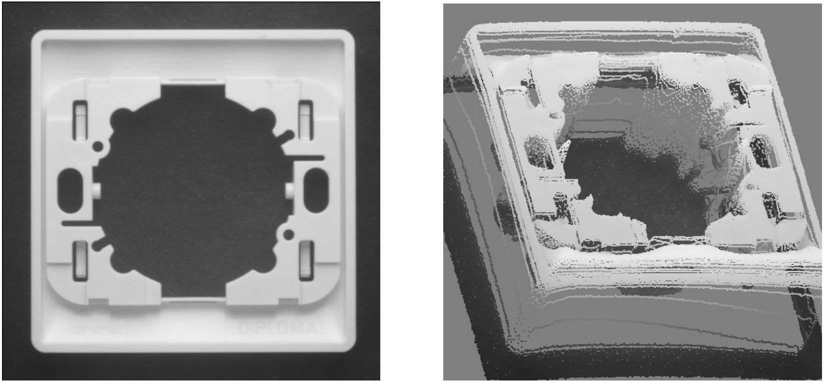


Figure : An image (left-hand side) and a view of its corresponding topographic surface (right-hand side).

Depth cameras

There are two *acquisition* technologies for depth-cameras, also called range- or 3D-cameras:

- measurements of the deformations of a pattern sent on the scene (*structured light*).

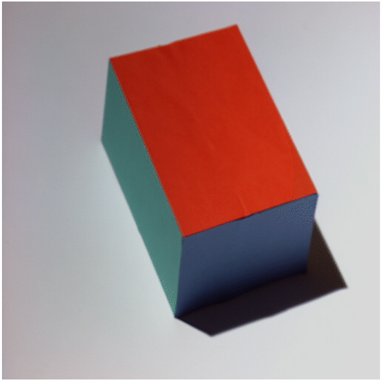
  - first generation of the Kinects

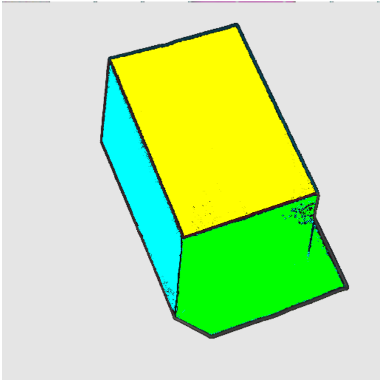
- measurements by *time-of-flight* (ToF). Time to travel forth and back between the source led (camera) and the sensor (camera).

  - Mesa Imaging, PMD cameras
  - second generation of Kinects

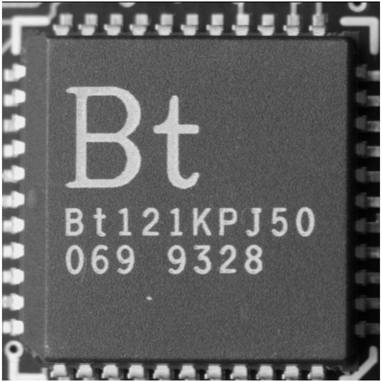
Illustration of a depth map acquired with a range camera

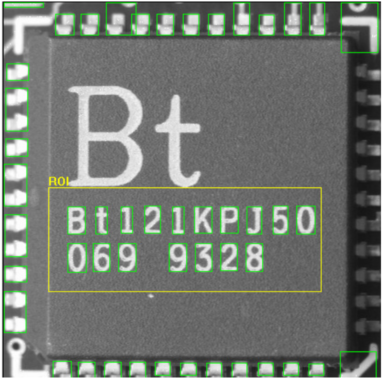
Image segmentation





Character recognition





Several successive stages:

- Selection of a Region of Interest (ROI). Processing is limited to that area.
- Detection of edges (contours).
- Identification and classification of characters.

Outline

Unitary transforms

- Computations on matrices
- Selection of well-known transforms
  - Discrete Fourier transform
  - Hadamard transform
  - Discrete Cosine Transform (DCT)
  - The continuous Fourier transform (only to express some properties)

What is the purpose of an image transform?

- Provide some *insight* on the image itself, by going into an equivalent space.
- Easy to compute and *interpretable*.
- Must be *reversible*. For example, the known continuous Fourier transform is reversible.

Let us model a sampled image f, as a matrix composed of $M \times N$ points or pixels

$$\underline{f} = \begin{bmatrix} f(0,0) & \cdots & f(0, N-1) \\ \vdots & & \vdots \\ f(M-1,0) & \cdots & f(M-1, N-1) \end{bmatrix}. \tag{10}$$

Let $\underline{P}$ et $\underline{Q}$ be two matrices of dimension $M \times M$ and $N \times N$ respectively:

$$\underline{F} = \underline{P}\,\underline{f}\,\underline{Q} \tag{11}$$

This might be expressed as

$$\underline{F}(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \underline{P}(u, m)\, \underline{f}(m, n)\, \underline{Q}(n, v). \tag{12}$$

Then, also,

$$\underline{f} = \underline{P}^{-1} \underline{F}\, \underline{Q}^{-1}. \tag{13}$$

Discrete Fourier transform I

The basis function for Fourier transforms is the imaginary exponential

$$W = e^{2\pi j/N} \tag{14}$$

We define a $J \times J$ transformation matrix, $\underline{\Phi}_{JJ}$, as

$$\underline{\Phi}_{JJ}(k,\ l) = \frac{1}{J}\exp\left(-j\frac{2\pi}{J}kl\right) \qquad k,\ l = 0,\ 1,\ \ldots,\ J-1. \tag{15}$$

Definition

Assuming the two following $\underline{P} = \underline{\Phi}_{MM}$ and $\underline{Q} = \underline{\Phi}_{NN}$ matrix transforms, then the Discrete Fourier Transform (DFT) is defined as

$$\underline{F} = \underline{\Phi}_{MM}\underline{f}\,\underline{\Phi}_{NN}. \tag{16}$$

Discrete Fourier transform II

Another, explicit, form of the DFT:

$$\underline{F}(u,\, v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \underline{f}(m,\, n) \exp\left[-2\pi j \left(\frac{mu}{M} + \frac{nv}{N}\right)\right], \quad (17)$$

where

$$u = 0,\, 1,\, \ldots,\, M-1 \qquad v = 0,\, 1,\, \ldots,\, N-1. \quad (18)$$

Inverse? Let us define the inverse matrix transformation, $\underline{\Phi}_{JJ}^{-1}$, as

$$\underline{\Phi}_{JJ}^{-1}(k, l) = \exp\left(j \frac{2\pi}{J} kl\right). \quad (19)$$

The, the inverse DFT is obtained as

$$\underline{f}(m,\, n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \underline{F}(u,\, v) \exp\left[2\pi j \left(\frac{mu}{M} + \frac{nv}{N}\right)\right] \quad (20)$$

Visualization

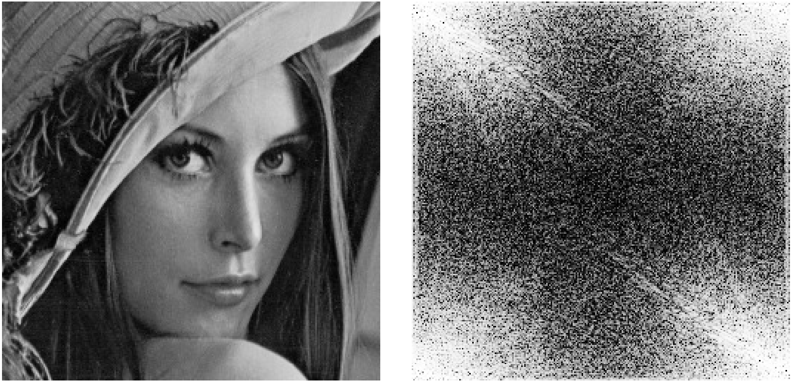


Figure : The Lena image and the module of its Discrete Fourier Transform.

Why is studying properties so important?

- They highlight the *behavior* of an operator.
- They are part of the *design* of a vision problem.
- They guide a practitioner towards a *solution*.

Periodicity

$$\underline{F}(u, -v) = \underline{F}(u, N - v) \qquad \underline{F}(-u, v) = \underline{F}(M - u, v) \tag{21}$$

$$\underline{f}(-m, n) = \underline{f}(M - m, n) \qquad \underline{f}(m, -n) = \underline{f}(m, N - n) \tag{22}$$

More generally,

$$\underline{F}(aM + u, bN + v) = \underline{F}(u, v) \qquad \underline{f}(aM + m, bN + n) = \underline{f}(m, n) \tag{23}$$

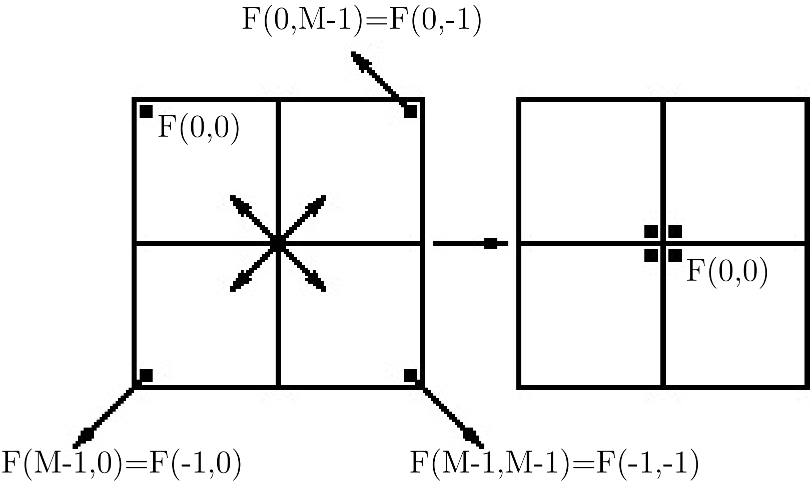


Figure : Reorganizing blocks (according to the periodicity) to center the spectrum.

Centered spectrum

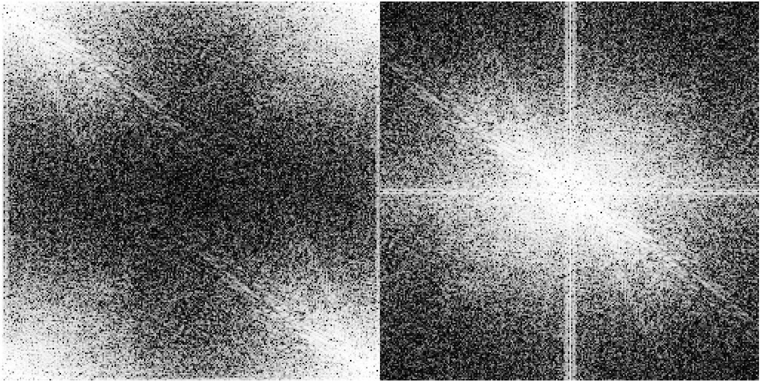


Figure : Spectrum of the Fourier transform before and after reorganizing the blocks to center the origin.

Hadamard transform

The second order Hadamard matrix is defined as

$$\underline{H}_{22} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{24}$$

The Hadamard matrix of order 2^k is generalized as

$$\underline{H}_{2J2J} = \begin{bmatrix} \underline{H}_{JJ} & \underline{H}_{JJ} \\ \underline{H}_{JJ} & -\underline{H}_{JJ} \end{bmatrix} \tag{25}$$

The inverse Hadamard matrix transform is given by

$$\underline{H}_{JJ}^{-1} = \frac{1}{J}\underline{H}_{JJ}. \tag{26}$$

Definition

The direct and inverse Hadamard transforms are defined respectively as

$$\underline{F} = \underline{H}_{MM}\underline{f}\,\underline{H}_{NN} \qquad \underline{f} = \frac{1}{MN}\underline{H}_{MM}\underline{F}\,\underline{H}_{NN}. \tag{27}$$

Discrete Cosine Transform (DCT) I

Let us define the following transform matrix $\underline{C}_{NN}(k,\ l)$

$$\underline{C}_{NN}(k,\ l) = \begin{cases} \frac{1}{\sqrt{N}} & l = 0, \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2k+1)l\pi}{2N}\right] & \textit{otherwise}. \end{cases} \qquad (28)$$

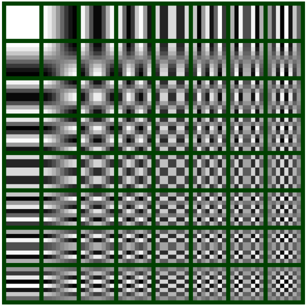


Figure : Basis functions of a Discrete Cosine Transform.

Discrete Cosine Transform (DCT) II

Definition

The Discrete Cosine Transform (DCT) and its inverse are defined respectively as

$$\underline{F} = \underline{C}_{NN} \underline{f} \, \underline{C}_{NN}^{T} \qquad \underline{f} = \underline{C}_{NN}^{T} \underline{F} \, \underline{C}_{NN}. \tag{29}$$

In an extended form, the DCT is expressed as

$$\underline{F}(u, v) = \frac{2c(u)\,c(v)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \underline{f}(m, n) \cos\left(\frac{2m+1}{2N} u\pi\right) \cos\left(\frac{2n+1}{2N} v\pi\right). \tag{30}$$

Why the DCT?

- a *nearly optimal* transform for compression
- used in the *JPEG* and *MPEG compression* standards

Definition

Assuming a continuous image $f(x, y)$, its Fourier transform is defined as

$$\mathcal{F}(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\, e^{-2\pi j(xu+yv)} dxdy. \tag{31}$$

We use this expression for studying the properties.

Inverse transform

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mathcal{F}(u, v)\, e^{2\pi j (ux + vy)} du dv \qquad (32)$$

Interpretation: the image is decomposed as a weighted sum of basic spectral components defined over $[-\infty, +\infty] \times [-\infty, +\infty]$. So $f(x, y)$ and $\mathcal{F}(u, v)$ form a pair of related representations of a same information:

$$f(x, y) \rightleftharpoons \mathcal{F}(u, v). \qquad (33)$$

In general, $\mathcal{F}(u, v)$ is a function of u and v with complex values. It can be expressed as

$$\mathcal{F}(u, v) = \|\mathcal{F}(u, v)\|\, e^{j\theta(u,v)}. \qquad (34)$$

$$\mathcal{F}(-u, -v) = \mathcal{F}^*(u, v) \tag{35}$$

Or,

$$\|\mathcal{F}(-u, -v)\| = \|\mathcal{F}(u, v)\| \tag{36}$$

$$\theta(-u, -v) = -\theta(u, v). \tag{37}$$

This leads to two important characteristics (for real-valued functions):

1. the spectrum is symmetric in the $u - v$ plane. Therefore, half the plane suffices.
2. the angle (phase) is anti-symmetric with respect to the origin of $u - v$.

Properties I

Why would we use linear transforms?

- to some extend, we may assume that some phenomena in image processing are linear (later we will challenge this assumption).
- Fourier transforms, and more generally linear transforms, are used because of their interesting properties.

1. **Separability**
   By permuting the integration order

$$\mathcal{F}(u,v) = \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} f(x,y)\, e^{-2\pi jxu} dx \right] e^{-2\pi jyv} dy \quad (38)$$

Properties II

2. **Linearity**
   Let $f_1(x,y) \rightleftharpoons \mathcal{F}_1(u,v)$ and $f_2(x,y) \rightleftharpoons \mathcal{F}_2(u,v)$. Then, for every constants c_1 and c_2,

$$c_1 f_1(x,y) + c_2 f_2(x,y) \rightleftharpoons c_1 \mathcal{F}_1(u,v) + c_2 \mathcal{F}_2(u,v) \tag{39}$$

3. **Zoom** [up to a factor, the output is scale-invariant]
   If $f(x,y) \rightleftharpoons \mathcal{F}(u,v)$, then

$$f(ax,by) \rightleftharpoons \frac{1}{|ab|}\mathcal{F}\left(\frac{u}{a},\frac{v}{b}\right) \tag{40}$$

4. **Spatial translation** [up to a phase, the norm of the output is translation-invariant]
   If $f(x,y) \rightleftharpoons \mathcal{F}(u,v)$, then

$$f(x-x_0,y-y_0) \rightleftharpoons \mathcal{F}(u,v)\,e^{-2\pi j(x_0 u + y_0 v)}$$

Properties III

5. **Spectral translation**
   If $f(x,y) \rightleftharpoons \mathcal{F}(u,v)$, then

$$f(x,y)\, e^{j2\pi(u_0 x + v_0 y)} \rightleftharpoons \mathcal{F}(u-u_0, v-v_0) \tag{41}$$

6. **Convolution**
   The convolution of $f(x,y)$ by $g(x,y)$ is defined as

$$(f \otimes g)(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\alpha,\beta)\, g(x-\alpha, y-\beta)\, d\alpha d\beta. \tag{42}$$

   If $f(x,y) \rightleftharpoons \mathcal{F}(u,v)$ and $g(x,y) \rightleftharpoons \mathcal{G}(u,v)$, then

$$(f \otimes g)(x,y) \rightleftharpoons \mathcal{F}(u,v)\, \mathcal{G}(u,v) \tag{43}$$

   The important result is that the convolution of two images can be obtained as the inverse Fourier transform of the product of the Fourier transform of both functions.

Consider the rectangular function $f\,(x,\ y)$ defined as

$$f\,(x,\ y) = ARect_{a,b}\,(x,\ y) \tag{44}$$

where

$$Rect_{a,b}\,(x,y) = \begin{cases} 1 & |x| < \frac{a}{2}, |y| < \frac{b}{2} \\ 0 & \text{elsewhere} \end{cases} \tag{45}$$

$$\begin{aligned} \mathcal{F}\,(u,v) &= \int_{-a/2}^{+a/2} dx \int_{-b/2}^{+b/2} dy A e^{-2\pi j(xu+yv)} && (46) \\ &= Aab \left(\frac{\sin\,(\pi a u)}{\pi a u}\right) \left(\frac{\sin\,(\pi b v)}{\pi b v}\right) && (47) \end{aligned}$$

Rectangular function II

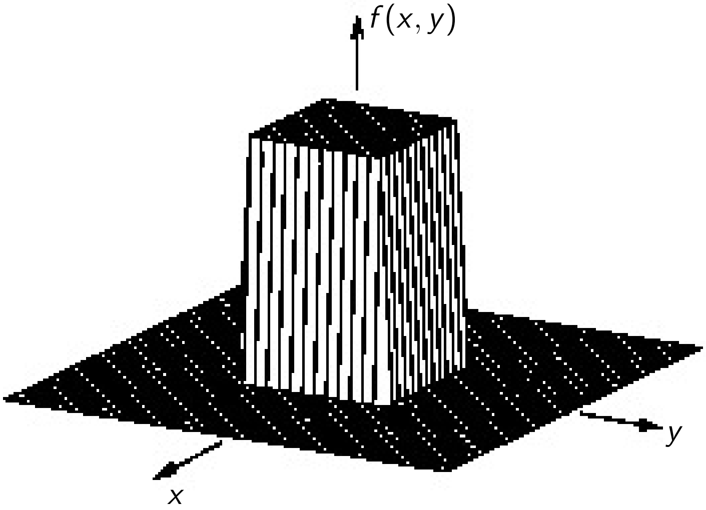


Figure : Rectangular function.

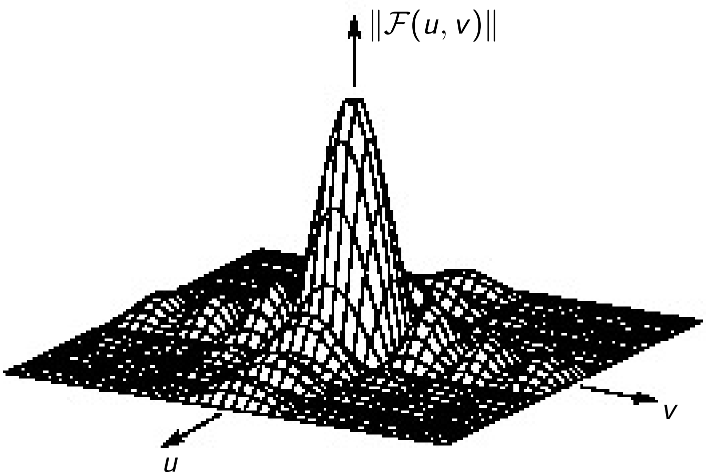


Figure : Module of the Fourier transform of the rectangular function.

Compression

Definition

The compression ratio is defined as

$$\frac{\text{bitrate prior to compression}}{\text{bitrate after compression}}. \tag{48}$$

Recent developments

H264 H.264/MPEG-4 Part 10 or AVC (Advanced Video Coding) is one of the most used format (Blu-ray compression for example).

H265 **High Efficiency Video Coding (HEVC)** is a video compression standard, a successor to H.264/MPEG-4 AVC (Advanced Video Coding) that was jointly developed by the ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) as ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.265. It can support resolutions up to 8192×4320!

Outline

- The notion of "ideal" filter
- Categories of ideal filters
- Typical filters:
  - Low-pass filters
  - High-pass filters
  - Gabor filters

Notion of "ideal" filter

A filter is said to be "ideal" if every transform coefficient is multiplied by 0 or 1.

Definition

[Ideal filter] An *ideal* filter is such that its transfer function is given by

$$\forall(u, v),\ \mathcal{H}(u, v) = 0 \ \text{ or } \ 1. \tag{49}$$

The notion of ideal filter is closely related to that of *idempotence*. The idempotence for a filter is to be understood such that, for an image $f(x, y)$,

$$\mathcal{F}(u, v)\mathcal{H}(u, v) = \mathcal{F}(u, v)\mathcal{H}(u, v)\mathcal{H}(u, v) \tag{50}$$

Typology of ideal filters I

For one-dimensional signals (such as the image function along an image line):

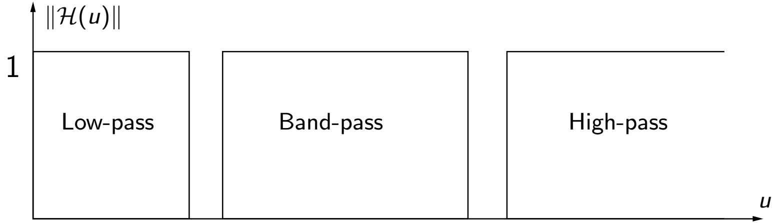
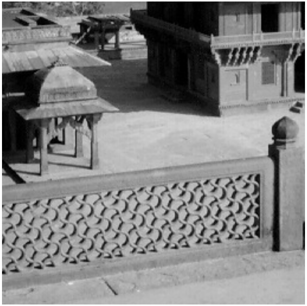


Figure : One-dimensional filters.

There are three types of circular ideal filters:

- *low-pass filters:*

$$\mathcal{H}(u,v) = \begin{cases} 1 & \sqrt{u^2+v^2} \leq R_0 \\ 0 & \sqrt{u^2+v^2} > R_0 \end{cases} \tag{51}$$



(a) Original image

(b) Low-pass filtered image

Typology of ideal filters III

- *High-pass filters:*

$$\mathcal{H}(u,v) = \begin{cases} 1 & \sqrt{u^2+v^2} \geq R_0 \\ 0 & \sqrt{u^2+v^2} < R_0 \end{cases} \tag{52}$$

- *pass-band filters*. There are equivalent to the complementary of a low-pass filter and a high-pass filter:

$$\mathcal{H}(u,v) = \begin{cases} 1 & R_0 \leq \sqrt{u^2+v^2} \leq R_1 \\ 0 & \text{otherwise} \end{cases} \tag{53}$$

Typology of ideal filters IV

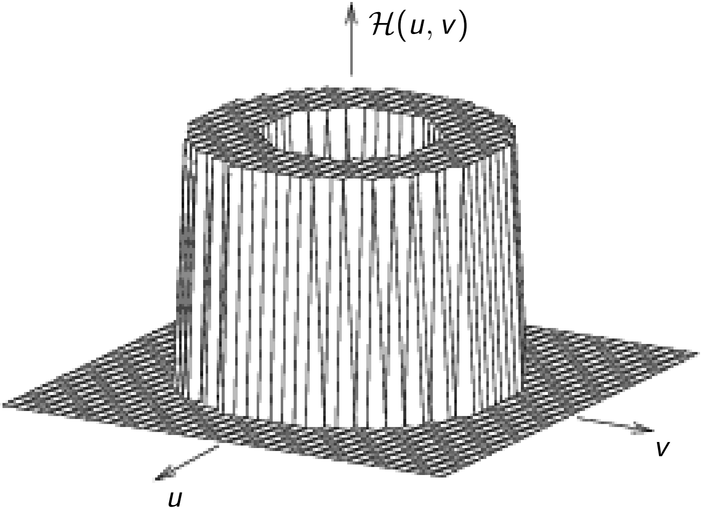


Figure : Transfer function of pass-band filters.

Effects of filtering

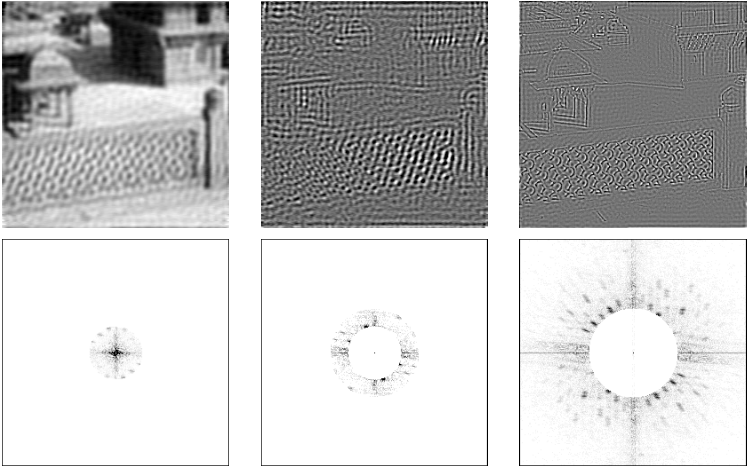


Figure : Fourier spectra of images filtered by three types of circular filters.

An image that passes through a linear filter

An two-dimensional filter is characterized by its two-dimensional *impulse function* $h(x, y)$. Theory shows that the filtered output image $g(x, y)$ can be computed as the convolution product between the impulse function and the input image $f(x, y)$

$$g(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\alpha, \beta)\, h(x - \alpha, y - \beta)\, d\alpha d\beta, \tag{54}$$

which is denoted as

$$g(x, y) = (f \otimes h)(x, y). \tag{55}$$

It can be shown that the convolution is the (x, y) plane is equivalent, in the Fourier domain, to

$$\mathcal{G}(u, v) = \mathcal{H}(u, v)\, \mathcal{F}(u, v).$$

A typical low-pass filter is the Butterworth filter (of order n) defined as

$$\mathcal{H}\left(u, v\right) = \frac{1}{1 + \left(\frac{\sqrt{u^2+v^2}}{R_0}\right)^{2n}}. \tag{56}$$

Low-pass filters II

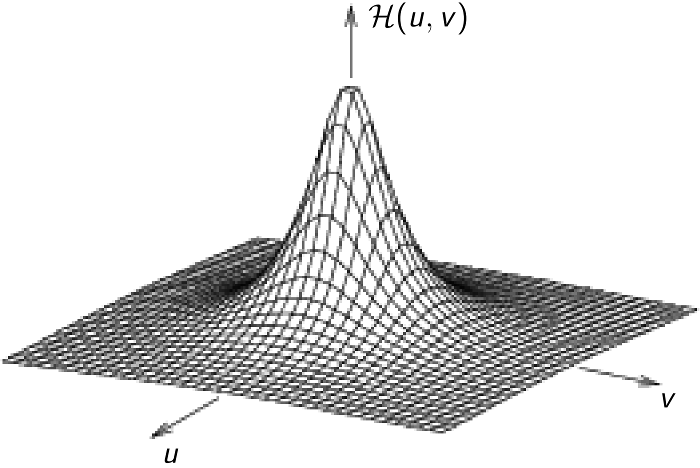
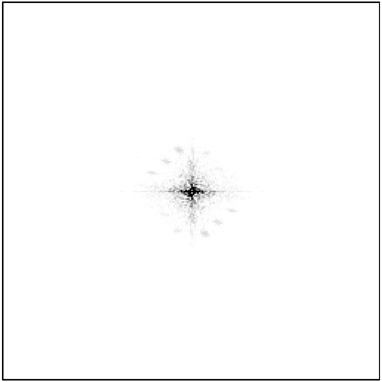


Figure : Transfer function of a low-pass Butterworth filter (wih $n = 1$).

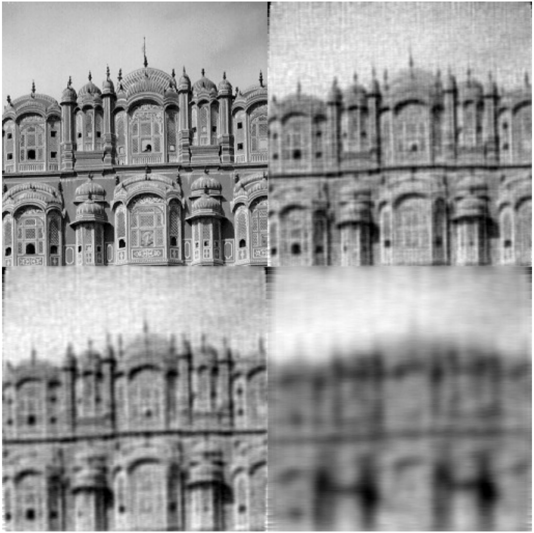
(a) Input image



(b) Spectrum of the filtered image

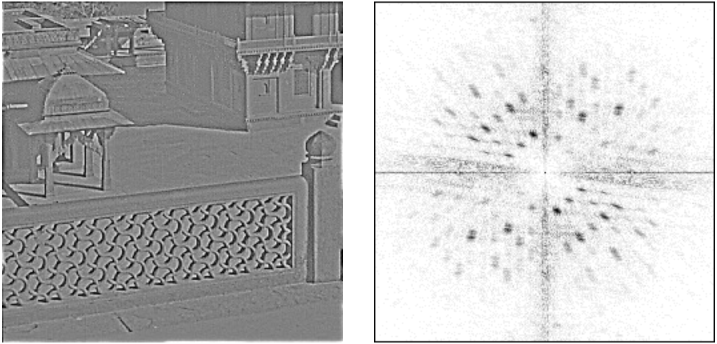
Figure : Effects of an order 1 Butterworth filter (cut-off frequency: $f_c = 30$).

Effect of a low-pass filter (with decreasing cut-off frequencies)



High-pass filters I

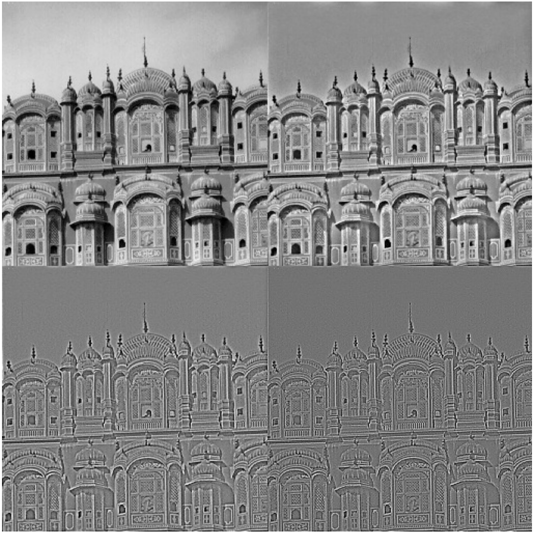
$$\mathcal{H}\left(u,v\right) = \frac{1}{1 + \left(\frac{R_0}{\sqrt{u^2+v^2}}\right)^{2n}} \tag{57}$$



(a) Filtered image (b) Spectrum of (a)

Figure : Effects of an order 1 Butterworth filter 1 (cut-off frequency: $f_c = 50$).

Effect of a high-pass filter (with increasing cut-off frequencies)



Definition

Gabor filters are particular class of linear filter. There are directed filters with a Gaussian-shaped impulse function:

$$h(x, y) = g(x', y')e^{2\pi j(Ux+Vy)} \tag{58}$$

- $(x', y') = (x\cos\phi + y\sin\phi\,, -x\sin\phi + y\cos\phi)$, these are the (x, y) coordinates rotated by an angle ϕ, and
- $g(x', y') = \frac{1}{2\pi\sigma^2}e^{(-(x'/\lambda)^2 + y'^2)/2\sigma^2}$.

The corresponding Fourier transform is given by

$$\mathcal{H}(u, v) = e^{-2\pi^2\sigma^2[(u'-U')^2\lambda^2 + (v'-V')^2]} \tag{59}$$

- $(u', v') = (u\cos\phi + v\sin\phi\,, -u\sin\phi + v\cos\phi)$, and
- (U', V') is obtained by rotating (U, V) with the same angle ϕ.

Gabor filters II

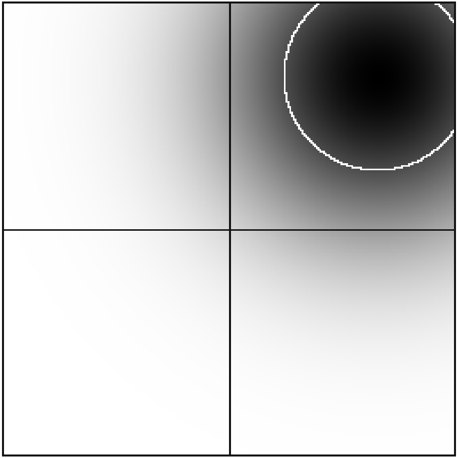


Figure : Transfer function of Gabor filter. The white circle represents the $-3\,[dB]$ circle ($=$ half the maximal amplitude).

Gabor filters III

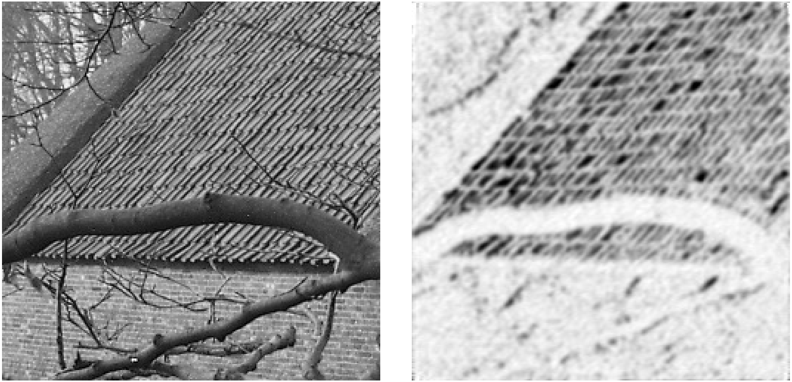


Figure : Input image and filtered image (with an filter oriented at 135^o).

Implementation

There are mainly 4 techniques to implement a Gaussian filter:

1. *Convolution* with a restricted Gaussian kernel. One often choose $N_0 = 3\sigma$ or 5σ

$$g_{1D}[n] = \begin{cases} \frac{1}{\sqrt{2}\sigma} e^{-(n^2/2\sigma^2)} & |n| \leq N_0 \\ 0 & |n| > N_0 \end{cases} \tag{60}$$

2. *Iterative convolution* with a uniform kernel:

$$g_{1D}[n] \simeq u[n] \otimes u[n] \otimes u[n] \tag{61}$$

   where

$$u[n] = \begin{cases} \frac{1}{(2N_0+1)} & |n| \leq N_0 \\ 0 & |n| > N_0 \end{cases} \tag{62}$$

3. Multiplication in the Fourier domain.
4. Implementation as a *recursive filter*.

Outline

- Reminders of the set theory
- Basic morphological transforms
- Neighboring transformations
- Geodesy and reconstruction
- Grayscale morphology

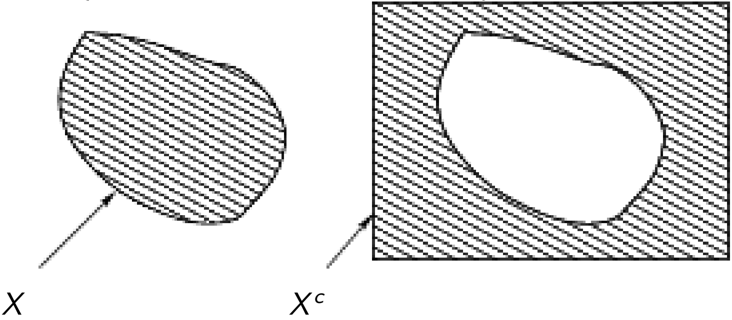
Sets will be denoted with capital letters, such as A, B, ..., and elements of these sets by lowercase letters a, b, ...

- **Set equality**
  Two sets are *equal* if they contain the same elements:
  $X = Y \Leftrightarrow (x \in X \Rightarrow x \in Y \text{ and } x \in Y \Rightarrow x \in X)$. The empty set is denoted as $\emptyset$.
- **Inclusion**
  X is a *subset* of Y (that is, X is included in Y) if all the elements of X also belong to Y: $X \subseteq Y \Leftrightarrow (x \in X \Rightarrow x \in Y)$.
- **Intersection**
  The *intersection* between X and Y is the set composed of the elements that belong to both sets:
  $X \cap Y = \{x \text{ such that } x \in X \text{ and } x \in Y\}$.

Reminders of the set theory II

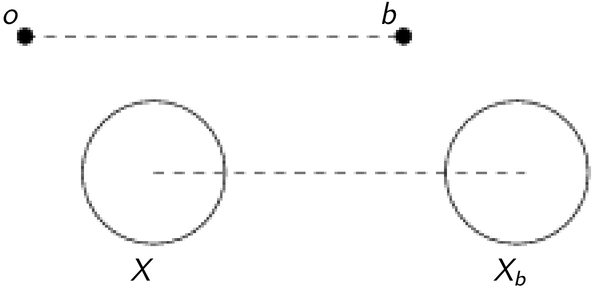
- **Union**
  The *union* between two sets is the set that gathers all the elements that belong to at least one set:
  $X \cup Y = \{x \text{ such that } x \in X \text{ or } x \in Y\}$.
- **Difference**
  The *set difference* between X and Y, denoted by $X - Y$ or $X \backslash Y$ is the set that contains the elements of X that are not in Y : $X - Y = \{x | x \in X \, and \, x \notin Y\}$.

- **Complementary**
  Assume that X is a subset of a $\mathcal{E}$ space, the *complementary* set of X with respect to $\mathcal{E}$ is the set, denoted X^c, given by $X^c = \{x \text{ such that } x \in \mathcal{E} \text{ and } x \notin X\}$.



- **Symmetric**
  The *symmetric* set, $\check{X}$, of X is defined as $\check{X} = \{-x | x \in X\}$.

- **Translated set**
  The *translate* of X by b is given by $\{z \in \mathcal{E} | z = x + b,\, x \in X\}$.



Basic morphological operators I

Erosion

Definition

Morphological erosion

$$X \ominus B = \{z \in \mathcal{E} | B_z \subseteq X\}. \tag{63}$$

The following algebraic expression is equivalent to the previous definition:

Definition

$$X \ominus B = \bigcap_{b \in B} X_{-b}. \tag{64}$$

B is named "**structuring element**".

Basic morphological operators II

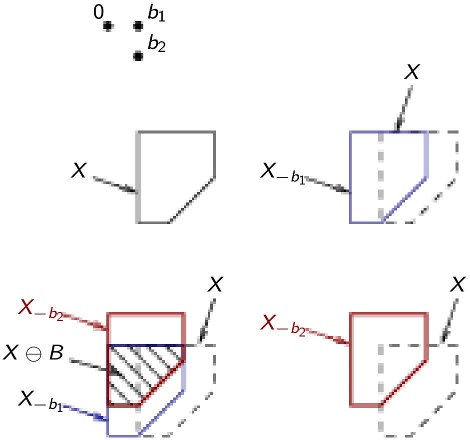


Figure : Algebraic interpretation of the erosion.

Erosion with a disk

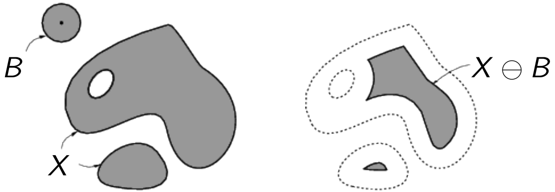


Figure : Erosion of X with a disk B. The origin of the structuring element is drawn at the center of the disk (with a black dot).

Dilation I

Definition

From an algebraic perspective, the *dilation* (*dila***ta***tion* in French!), is the union of translated version of X:

$$X \oplus B = \bigcup_{b \in B} X_b = \bigcup_{x \in X} B_x = \{x + b | x \in X,\ b \in B\}. \tag{65}$$

Dilation II

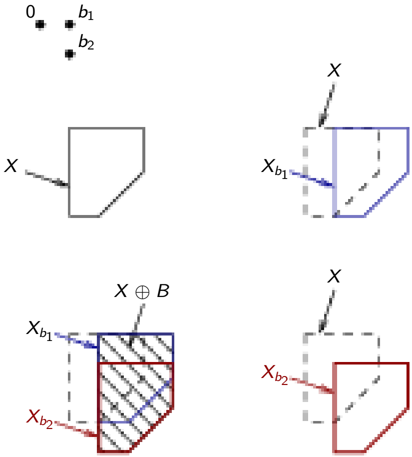


Figure : Illustration of the algebraic interpretation of the dilation operator.

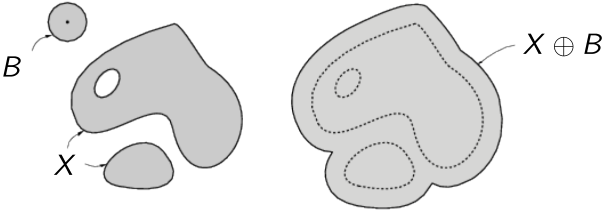


Figure : Dilation of X with a disk B.

Duality

Erosion and dilation are two dual operators with respect to complementation:

$$X \ominus \check{B} = (X^c \oplus B)^c \qquad (66)$$

$$X \ominus B = (X^c \oplus \check{B})^c \qquad (67)$$

Erosion and dilation obey the principles of "ideal" morphological operators:

1. erosion and dilation are invariant to translations: $X_z \ominus B = (X \ominus B)_z$. Likewise,$X_z \oplus B = (X \oplus B)_z$;
2. erosion and dilation are compatible with scaling: $\lambda X \ominus \lambda B = \lambda(X \ominus B)$ and $\lambda X \oplus \lambda B = \lambda(X \oplus B)$;
3. erosion and dilation are local operators (if B is bounded);
4. it can be shown that erosion and dilation are continuous transforms.

Algebraic properties

- erosion and dilation are *increasing* operators: if $X \subseteq Y$, then $(X \ominus B) \subseteq (Y \ominus B)$ and $(X \oplus B) \subseteq (Y \oplus B)$;
- if the structuring element contains the origin, then the erosion is *anti-extensive* and the dilation is *extensive*, that is $X \ominus B \subseteq X$ and $X \subseteq X \oplus B$.

Morphological opening I

Definition

The *opening* results from cascading an erosion and a dilation with the same structuring element:

$$X \circ B = (X \ominus B) \oplus B. \tag{68}$$

The interpretation of the opening operator (which can be seen as an alternative definition) is based on

$$X \circ B = \bigcup \{B_z | z \in \mathcal{E} \,\mathrm{and}\, B_z \subseteq X\}. \tag{69}$$

In other words, the opening of a set by structuring element B is the set of all the elements of X that are covered by a translated copy of B when it moves inside of X.

Definition

A *closing* is obtained by cascading a dilation and an erosion with a unique structuring element:

$$X \bullet B = (X \oplus B) \ominus B. \tag{70}$$

Opening and closing are dual operators with respect to set complementation: indeed,

$$(X \circ B)^c = X^c \bullet \check{B} \tag{71}$$

and

$$(X \bullet B)^c = X^c \circ \check{B}. \tag{72}$$

Illustration

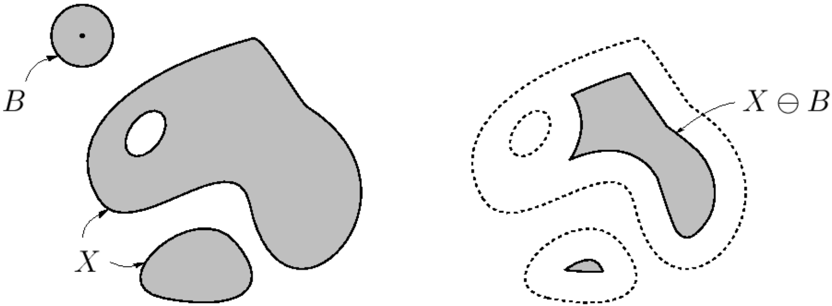


Figure : Opening and closing of X with a disk B.

By construction, the opening and closing follow the "ideal" principles of morphological operators.
The most important algebraic properties of $X \circ B$ and $X \bullet B$ are

1. opening and closing are *increasing*. If $X \subseteq Y$, then

$$(X \circ B) \subseteq (Y \circ B) \text{ and } (X \bullet B) \subseteq (Y \bullet B) \tag{73}$$

2. opening is *anti-extensive*, and closing is extensive

$$X \circ B \subseteq X, \quad X \subseteq X \bullet B \tag{74}$$

3. opening and closing are *idempotent* operators (projective operators). This means that

$$(A \circ B) \circ B = A \circ B \text{ and } (A \bullet B) \bullet B = A \bullet B \tag{75}$$

- Dilation is *commutative* and *associative*

$$X \oplus B = B \oplus X \tag{76}$$

$$(X \oplus Y) \oplus C = X \oplus (Y \oplus C) \tag{77}$$

- Dilation distributes the union

$$(\bigcup_j X_j) \oplus B = \bigcup_j (X_j \oplus B) \tag{78}$$

- The erosion distributes the intersection

$$(\bigcap_j X_j) \ominus B = \bigcap_j (X_j \ominus B) \tag{79}$$

- *Chain rule* ($\equiv$ *cascading rule*):

$$X \ominus (B \oplus C) = (X \ominus B) \ominus C \tag{80}$$

General properties II

- The opening and closing are not related to the exact location of the origin (so they do not depend on the location of the origin when defining B). Let $z \in \mathcal{E}$

$$X \circ B_z = X \circ B \tag{81}$$
$$X \bullet B_z = X \bullet B \tag{82}$$

A practical problem: dealing with borders I

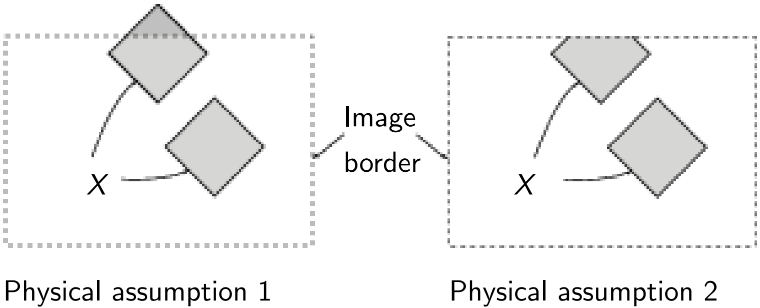
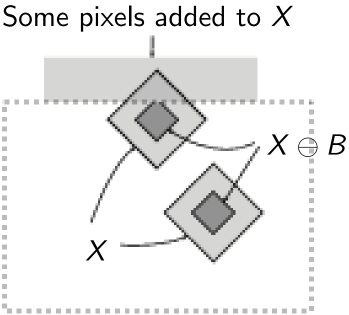
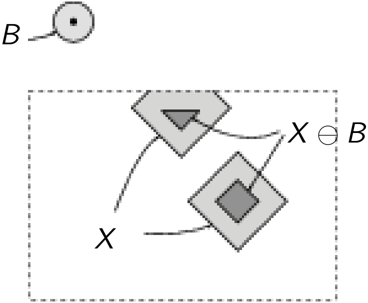


Figure : Two possible physical assumptions for borders.





Physical assumption 1

Physical assumption 2

Figure : Comparison of the effects of two physical assumptions on the computation of the erosion of X.

The *Hit or Miss transform* *is defined such as*

$$X \Uparrow (B, C) = \{x | B_x \subseteq X,\ C_x \subseteq X^c\} \tag{83}$$

If $C = \emptyset$ the transform reduces to an erosion of X by B.

Geodesic dilation

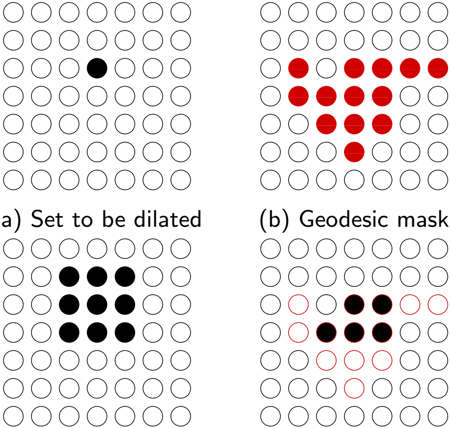
A geodesic dilation is always based on two sets (images).

Definition

The *geodesic dilation* of size 1 of X conditionally to Y, denoted $D_Y^{(1)}(X)$, is defined as the intersection of the dilation of X and Y:

$$\forall X \subseteq Y, \;\; D_Y^{(1)}(X) = (X \oplus B) \cap Y \tag{84}$$

where B is usually chosen according to the frame connectivity (a 3×3 square for a 8-connected grid).



(a) Set to be dilated (b) Geodesic mask

(c) Elementary dilation (d) Geodesic dilation

Figure : Geodesic dilation of size 1.

Definition

The size n *geodesic dilation* of a set X conditionally to Y, denoted $D_Y^{(n)}(X)$, is defined as n successive geodesic dilation of size 1:

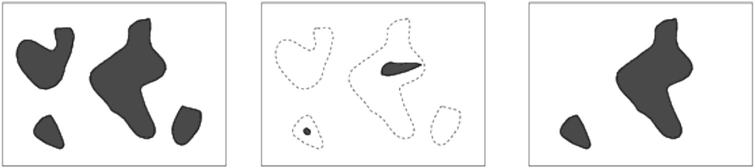
$$\forall X \subseteq Y, \;\; D_Y^{(n)}(X) = \underbrace{D_Y^{(1)}(D_Y^{(1)}(\dots D_Y^{(1)}}_{\text{n times}}(X))) \qquad (85)$$

where B is usually chosen according to the frame connectivity.

Definition

The *reconstruction* of X conditionally to Y is the geodesic dilation of X until idempotence. Let i be the iteration during which idempotence is reached, then the reconstruction of X is given by

$$R_Y(X) = D_Y^{(i)}(X) \text{ with } D_Y^{(i+1)}(X) = D_Y^{(i)}(X). \tag{86}$$



(a) Blobs (b) Marking blobs (c) Reconstructed blobs

Figure : Blob extraction by marking and reconstruction.

Notion of a function

Let $\mathcal{G}$ be the range of possible grayscale values. An image is represented by a function $f : \mathcal{E} \rightarrow \mathcal{G}$, which projects a location of a value of $\mathcal{G}$. In practice, an image is not defined over the entire space $\mathcal{E}$, but on a limited portion of it, a compact D.

We need to define an order between functions.

Definition

[Partial ordering between functions] Let f and g be functions. f is inferior to g,

$$f \leq g \text{ if } f(x) \leq g(x), \forall x \in \mathcal{E} \tag{87}$$

Definition

[Infimum and supremum] Let f_i be a family of functions, $i \in I$. The *infimum* (respectively the *supremum*) of this family, denoted $\wedge_{i \in I} f_i$ (resp. $\vee_{i \in I} f_i$) is the largest lower bound (resp. the lowest upper bound).

In the practical case of a finite family I, the supremum and the infimum correspond to the maximum and the minimum respectively. In that case,

$$\forall x \in \mathcal{E}, \begin{cases} (f \vee g)(x) = \max(f(x), g(x)) \\ (f \wedge g)(x) = \min(f(x), g(x)) \end{cases} \tag{88}$$

Grayscale morphology III

Definition

The translate of *a function f* by b, denoted by f_b, is defined as

$$\forall x \in \mathcal{E}, \quad f_b(x) = f(x - b). \tag{89}$$

Additional definitions related to operators I

Definition

[Idempotence] An operator ψ is *idempotent* if, for each function, a further application of it does not change the final result. That is, if

$$\forall f,\ \psi(\psi(f)) = \psi(f) \tag{90}$$

Definition

[Extensivity] An operator is *extensive* if the result of applying the operator is larger that the original function

$$\forall f,\ f \leq \psi(f) \tag{91}$$

Additional definitions related to operators II

Definition

[Anti-extensivity] An operator is *anti-extensive* if the result of applying the operator is lower that the original function

$$\forall f,\ f \geq \psi(f) \tag{92}$$

Definition

[Increasingness] An increasing operator is such that it does not modify the ordering between functions:

$$\forall f, g,\ f \leq g \Rightarrow \psi(f) \leq \psi(g) \tag{93}$$

By extension, an operator ψ_1 is lower that an operator ψ_2 if, for every function f, $\psi_1(f)$ is lower to $\psi_2(f)$:

$$\psi_1 \leq \psi_2 \Leftrightarrow \forall f,\ \psi_1(f) \leq \psi_2(f) \tag{94}$$

Definition

Let B be the domain of definition of a structuring element. The *grayscale dilation* and *erosion* (with a flat structuring element) are defined, respectively as,

$$f \oplus B = \bigvee_{b \in B} f_b(x) \quad (95)$$

$$f \ominus B = \bigwedge_{b \in B} f_{-b}(x) \quad (96)$$

Numerical example ($B = \{-1, 0, 1\}$)

$f(x)$	25	27	30	24	17	15	22	23	25	18	20
$f(x-1)$		25	27	30	24	17	15	22	23	25	18
$f(x)$	25	27	30	24	17	15	22	23	25	18	20
$f(x+1)$	27	30	24	17	15	22	23	25	18	20	
$f \ominus B(x) = \min$		25	24	17	15	15	15	22	18	18	
$f \oplus B(x) = \max$		30	30	30	24	22	23	25	25	25	

Typical questions:

- best algorithms? (note that there is some redundancy between neighboring pixels)
- how do proceed around borders?

Algorithms

- Based on the decomposition of the structuring element:
  - $f \ominus (H \oplus V) = (f \ominus H) \ominus V$
  - $f \ominus (B \oplus B) = (f \ominus B) \ominus \partial(B)$
- Appropriate structure for storing and propagating the local min and max
  - queues
  - histogram

Illustration I

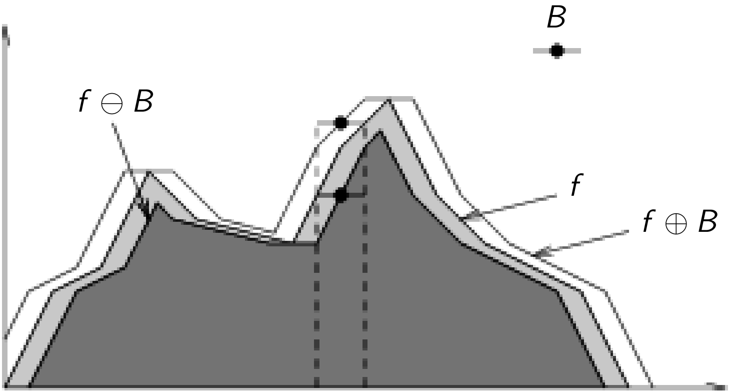


Figure : Erosion of a function.

Illustration II

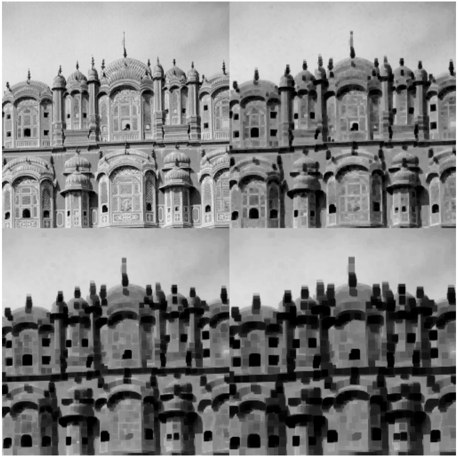


Figure : Erosions with squares of increasing sizes.

Illustration III

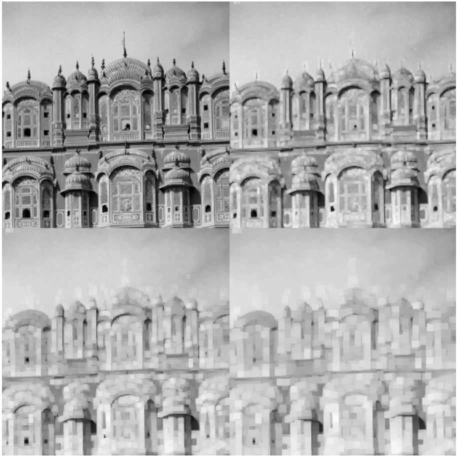


Figure : Dilations with squares of increasing sizes.

The opening $f \circ B$ is obtained by cascading an erosion followed by a dilation. The closing $f \bullet B$ is the result of a dilation followed by an erosion.

Definition

[Morphological opening and closing]

$$
\begin{aligned}
f \circ B &= (f \ominus B) \oplus B && (97) \\
f \bullet B &= (f \oplus B) \ominus B && (98)
\end{aligned}
$$

Morphological opening and closing II

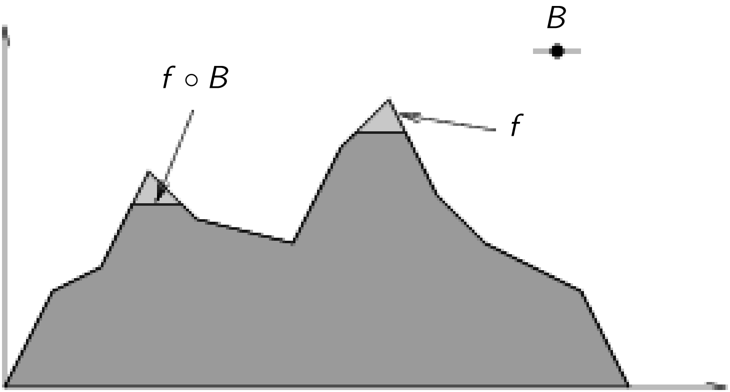


Figure : Opening of a function.

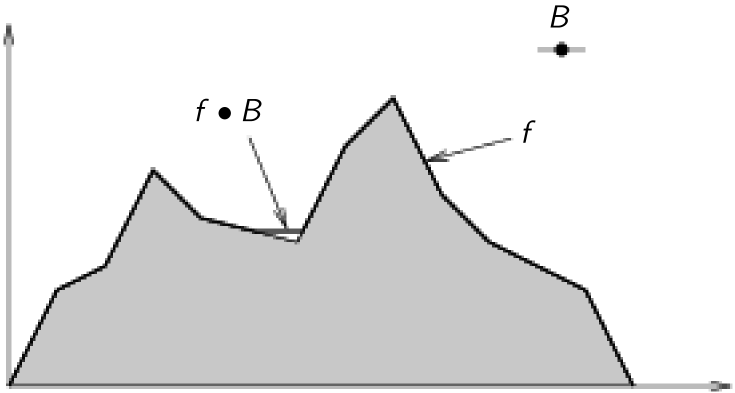


Figure : Closing of a function.

Morphological opening and closing IV

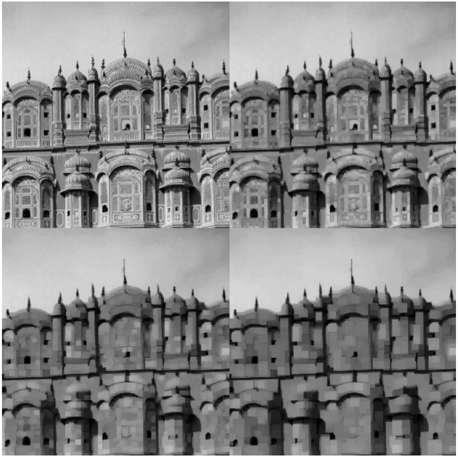


Figure : Opening with squares of increasing sizes.

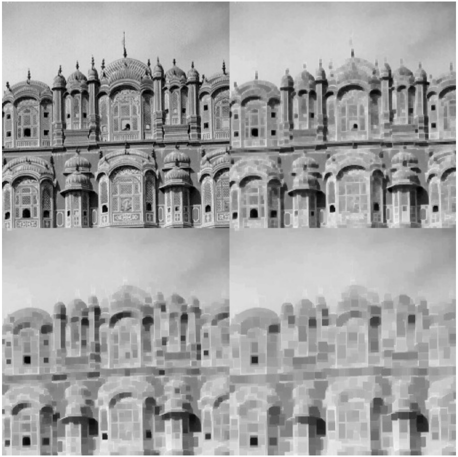
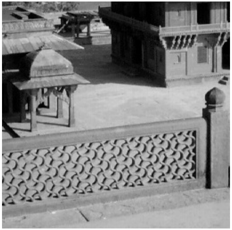


Figure : Closing with squares of increasing sizes.

Morphological opening and closing VI



(a) Original image f

(b) Erosion with a square

(c) Dilation with a square

(d) Opening with a square

Figure : Morphological operators on a grayscale image.

Properties of grayscale morphological operators

Erosion and dilation are increasing operators

$$f \leq g \Rightarrow \begin{cases} f \ominus B \leq g \ominus B \\ f \oplus B \leq g \oplus B \end{cases} \tag{99}$$

Erosion distributes the infimum and dilation distributes the supremum

$$(f \wedge g) \ominus B = (f \ominus B) \wedge (g \ominus B) \tag{100}$$

$$(f \vee g) \oplus B = (f \oplus B) \vee (g \oplus B) \tag{101}$$

Opening and closing are idempotent operators

$$(f \circ B) \circ B = f \circ B \tag{102}$$

$$(f \bullet B) \bullet B = f \bullet B \tag{103}$$

Opening and closing are anti-extensive and extensive operators respectively

$$f \circ B \leq f \tag{104}$$

$$f \leq f \bullet B \tag{105}$$

Reconstruction of grayscale images I

Definition

The *reconstruction* of f, conditionally to g, is the geodesic dilation of f until idempotence is reached. Let i, be the index at which idempotence is reached, the reconstruction of f is then defined as

$$R_g(f) = D_g^{(i)}(f) \text{ with } D_g^{(i+1)}(f) = D_g^{(i)}(f). \tag{106}$$

Reconstruction of grayscale images II

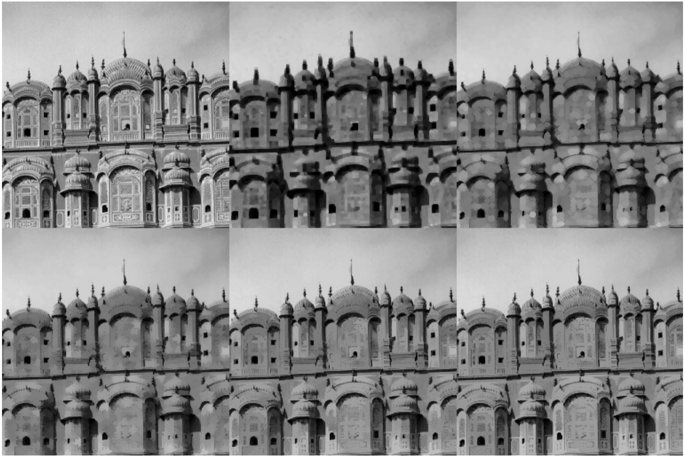


Figure : Original image, eroded image, and several successive geodesic dilations.

Reconstruction of grayscale images III

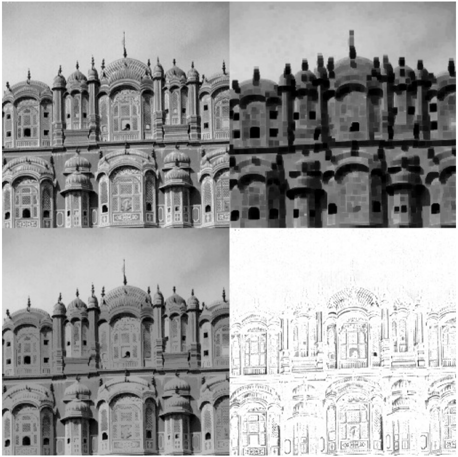


Figure : Original image, eroded image, reconstructed image starting from the eroded image, and difference image (reverse video).

Reconstruction of grayscale images IV

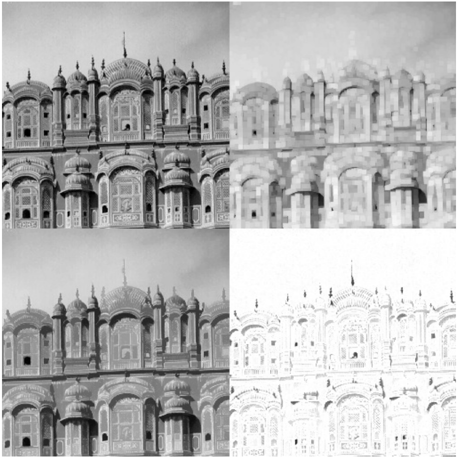


Figure : Original image, dilated image, reconstructed image starting from the dilated image (dual reconstruction), and difference image (reverse video).

Outline

Non-linear filtering

- Rank filters
  - Median
- Morphological filters
  - Algebraic definition
  - How to build a filter?
  - Examples of filters
    - Alternate sequential filters
    - Morphological filter

$f(x)$	25	27	30	24	17	15	22	23	25	18	20

$f(x-1)$	?	25	27	30	24	17	15	22	23	25	18
$f(x)$	25	27	30	24	17	15	22	23	25	18	20
$f(x+1)$	27	30	24	17	15	22	23	25	18	20	?
$f \ominus B(x) = \min$		25	24	17	15	15	15	22	18	18	
$f \oplus B(x) = \max$		30	30	30	24	22	23	25	25	25	

We could order the values

$f(x)$	25	27	30	24	17	15	22	23	25	18	20

1		25	24	17	15	15	15	22	18	18	18
2	25	27	27	24	17	17	22	23	23	20	20
3	27	30	30	30	24	22	23	25	25	25	
$f \ominus B(x) = \min$		25	24	17	15	15	15	22	18	18	
$f \oplus B(x) = \max$		30	30	30	24	22	23	25	25	25	

Definition of rank filters

Let $k \in \mathbb{N}$ be a threshold.

Definition

[Rank filter] The operator or k-order rank filter, denoted as $\rho_{B,k}(f)(x)$, defined with respect to the B structuring element, is

$$\rho_{B,k}(f)(x) = \bigvee \{t \in \mathcal{G} | \sum_{b \in B} [f(x+b) \geq t] \geq k\} \tag{107}$$

The simplest interpretation is that $\rho_{B,k}(f)(x)$ is the k-est value when all the $f(x+b)$ values are ranked in decreasing order. Rank filters are ordered. Let $\sharp(B)$, be the surface of B, then

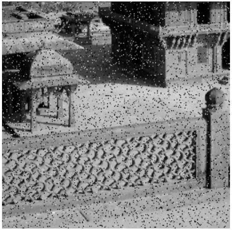
$$\rho_{B,\sharp(B)}(f)(x) \leq \rho_{B,\sharp(B)-1}(f)(x) \leq \ldots \leq \rho_{B,1}(f)(x) \tag{108}$$

Median filter I

If n is odd, the $k = \frac{1}{2}(\sharp(B) + 1)$ choice leads to the definition of a self-dual operator, that is a filter that produces the same result as if applied on the dual function. This operator, denoted med_B, is the *median filter*.

$f(x)$	25	27	30	24	17	15	22	23	25	18	20
1		25	24	17	15	15	15	22	18	18	18
med_B	25	27	27	24	17	17	22	23	23	20	20
3	27	30	30	30	24	22	23	25	25	25	
$f \ominus B(x) = \min$		25	24	17	15	15	15	22	18	18	
$f \oplus B(x) = \max$		30	30	30	24	22	23	25	25	25	

Median filter II



(a) Original image f + noise

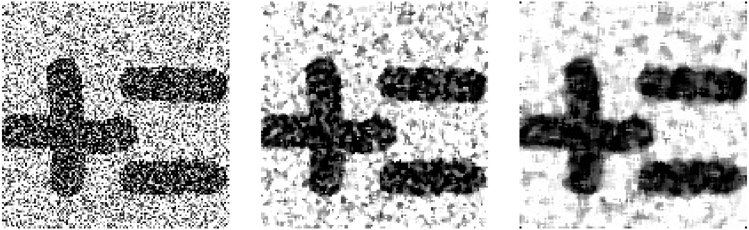
(b) Opening with a 5×5 square

(c) Low-pass Butterworth ($f_c = 50$)

(d) Median with a 5×5 square

Figure : Comparison of filters on a noisy image.

Effect of the size of the median filter



(a) Image f (b) 3×3 median (c) 5×5 median

Notes about the implementation

The median filter is not idempotent. Successive applications can result in oscillations (theoretically if the domain of the function is infinite)

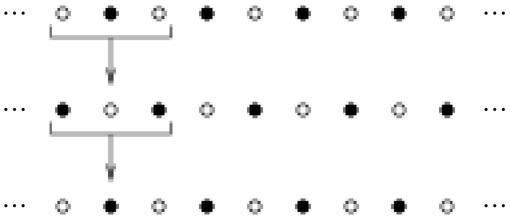


Figure : Repeated application of a median filter.

Also,

$$med_{5\times5}(f) \neq med_{1\times5}(med_{5\times1}(f)) \tag{109}$$

but it's an acceptable *approximation*.

Morphological filters

Definition

By definition, a filter is an *algebraic filter* iff the operator is increasing and idempotent:

$$\psi \text{ is an algebraic filter } \Leftrightarrow \forall f, g \begin{cases} f \leq g \Rightarrow \psi(f) \leq \psi(g) \\ \psi(\psi(f)) = \psi(f) \end{cases} \tag{110}$$

Definition

An *algebraic opening* is an operator that is increasing, idempotent, and anti-extensive. Formally,

$$\forall f, g, \; f \leq g \Rightarrow \psi(f) \leq \psi(g) \tag{111}$$

$$\forall f, \; \psi(\psi(f)) = \psi(f) \tag{112}$$

$$\forall f, \; \psi(f) \leq f \tag{113}$$

An *algebraic closing* is defined similarly, except that the operator is extensive.

By combining know filters!

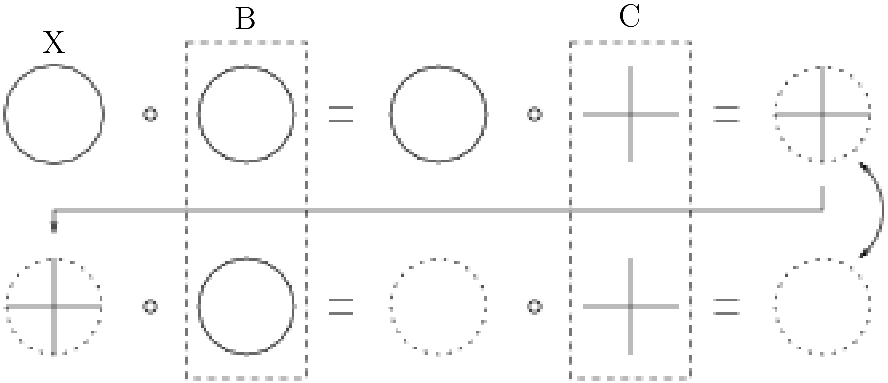


Figure : The composition of two openings is not an opening.

How to build a filter? II

New filters can be built starting from openings, denoted α_i, and closings, denoted ϕ_i. The rules to follow are:

1. the supremum of openings is an opening: $(\bigvee_i \alpha_i)$ is an opening;
2. the infimum of closings is a closing: $(\bigwedge_i \phi_i)$ is a closing.

Composition rules: structural theorem

Let ψ_1 and ψ_2 be two filters such that $\psi_1 \geq I \geq \psi_2$ (for example, ψ_1 is a closing and ψ_2 an opening).

Theorem

[Structural theorem] *Let ψ_1 and ψ_2 be two filters such that $\psi_1 \geq I \geq \psi_2$, then*

$$\psi_1 \geq \psi_1\psi_2\psi_1 \geq (\psi_2\psi_1 \vee \psi_1\psi_2) \geq (\psi_2\psi_1 \wedge \psi_1\psi_2) \geq \psi_2\psi_1\psi_2 \geq \psi_2 \tag{114}$$

$$\psi_1\psi_2,\ \psi_2\psi_1,\ \psi_1\psi_2\psi_1,\ \psi_2\psi_1\psi_2 \text{ are all filers} \tag{115}$$

Note that there is no ordering between $\psi_1\psi_2$ and $\psi_2\psi_1$.

Alternate Sequential Filters (ASF)

Let γ_i (ϕ_i) be an opening (resp. a closing) of size i and I be the identity operatog (i.e. $I(f) = f$). We assume that there is the following order:

$$\forall i, j \in \mathbb{N}, \quad i \leq j, \quad \gamma_j \leq \gamma_i \leq I \leq \phi_i \leq \phi_j, \tag{116}$$

For each index i, we define these operators:

$$\begin{aligned} m_i &= \gamma_i \phi_i, \qquad r_i = \phi_i \gamma_i \phi_i, \\ n_i &= \phi_i \gamma_i, \qquad s_i = \gamma_i \phi_i \gamma_i. \end{aligned}$$

Examples of filters II

Definition

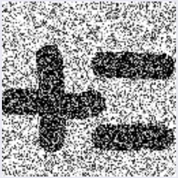
[Alternate Sequential Filters (ASF)] For each index $i \in \mathbb{N}$, the following operators are the alternate sequential filters of index i

$$M_i = m_i m_{i-1} \ldots m_2 m_1 \qquad R_i = r_i r_{i-1} \ldots r_2 r_1 \tag{117}$$
$$N_i = n_i n_{i-1} \ldots n_2 n_1 \qquad S_i = s_i s_{i-1} \ldots s_2 s_1 \tag{118}$$

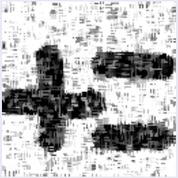
Theorem

[Absorption law]

$$i \leq j \Rightarrow M_j M_i = M_j \;\; but \;\; M_i M_j \leq M_j \tag{119}$$



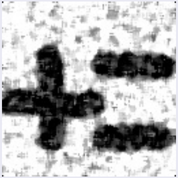
(a) Image f



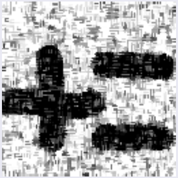
(b) $M_1(f)$

(c) $M_2(f)$

(d) $M_3(f)$



(e) 5×5 median



(f) $N_1(f)$

(g) $N_2(f)$

(h) $N_3(f)$

Toggle mappings I

The *morphological center* is a typical example of toggle mapping.

Definition

[Morphological center] Let ψ_i be a family of operators. The *morphological center* β of a function f with respect to the ψ_i family is defined, for each location x of the domain of f as follows:

$$\beta(f)(x) = (f(x) \vee (\bigwedge_i \psi_i(x))) \wedge (\bigvee_i \psi_i(x)) \tag{120}$$

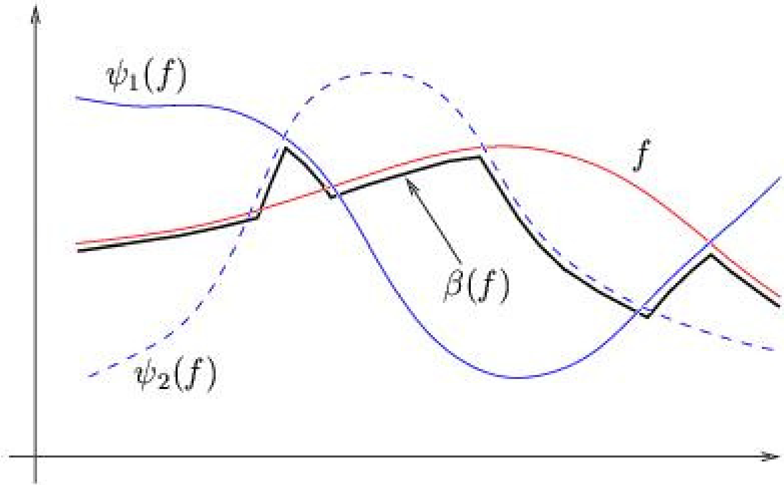


Figure : Morphological center of a one-dimensional signal.

Outline

- Linear operators
  - First derivate operators
  - Second derivate operators
  - Sampling the derivate
    - Residual error
    - Synthesis of operators for a fixed error
  - Practical expressions of gradient operators and convolution masks
- Non-linear operators
  - Morphological gradients

What's a border/contour/edge?

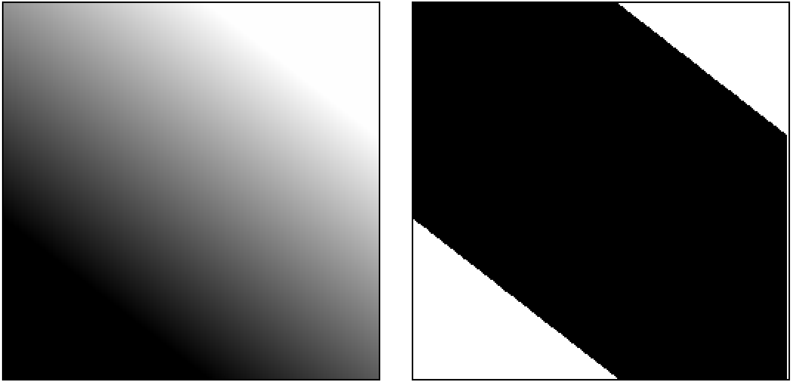


Figure : An image (diagonal ramp) and its contours (in black).

Can we locate edge points?

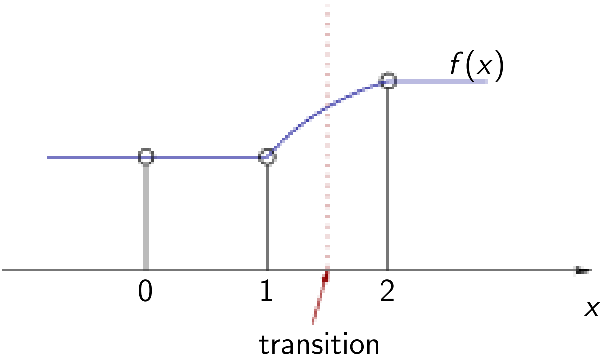


Figure : Problem encountered to locate an edge point.

Linear operators I

For derivate operators, we have to address two problems:

1. find the best approximate for the derivate
2. avoid an excessive amplification of the noise

These are two apparent contradictory requirements $\Rightarrow$ trade-offs

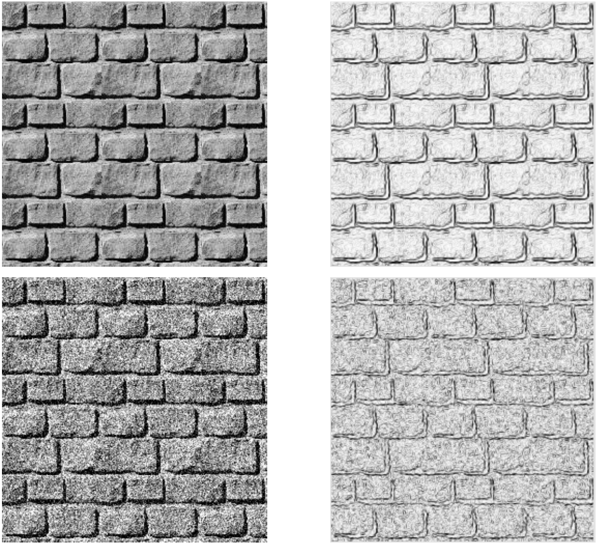


Figure : Images (left-hand side) and gradient images (right-hand side)

First derivate operator I

Let us consider the *partial* derivate of a function $f(x, y)$ with respect to x. Its Fourier transform is given by

$$\frac{\partial f}{\partial x}(x, y) \rightleftharpoons 2\pi ju\,\mathcal{F}(u, v) \tag{121}$$

In other words, deriving with respect to x consists of multiplying the Fourier transform of $f(x, y)$ by the following transfer function $\mathcal{H}_x(u, v) = 2\pi ju$, or of filtering $f(x, y)$ with the following impulse function:

$$h_x(x, y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}(2\pi ju)\,e^{2\pi j(xu+yv)}\,dudv \tag{122}$$

If we adopt a vectorial notation of the derivate, we define the *gradient* ∇f of image f by

$$\nabla f = \frac{\partial f}{\partial x}\overrightarrow{e_x} + \frac{\partial f}{\partial y}\overrightarrow{e_y} = (h_x \otimes f)\overrightarrow{e_x} + (h_y \otimes f)\overrightarrow{e_y} \tag{123}$$

First derivate operator II

Definition

[Gradient amplitude]

$$|\nabla f| = \sqrt{(h_x \otimes f)^2 + (h_y \otimes f)^2} \tag{124}$$

The amplitude of the gradient is sometimes approximated by

$$|\nabla f| \simeq |h_x \otimes f| + |h_y \otimes f| \tag{125}$$

which introduces a still acceptable error (in most cases) of 41%!

Definition

[Gradient orientation]

$$\varphi_{\nabla f} = tan^{-1}\left(\frac{h_y \otimes f}{h_x \otimes f}\right) \tag{126}$$

Definition

[Laplacian]

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = (h_{xx} \otimes f) + (h_{yy} \otimes f) \tag{127}$$

As the first derivate, it can be shown that in the Fourier domain, the Laplacian consists to apply the following filter

$$\nabla^2 f \rightleftharpoons -4\pi^2(u^2 + v^2)\mathcal{F}(u, v) \tag{128}$$

As can be seen, high frequencies tend to be amplified.

In order to derive practical expressions for the computation of a derivate, we adopt the following approach:

- develop some approximations and compute the resulting error,
- study the spectral behavior of these approximations, and
- discuss some practical approximations expressed in the terms of convolution masks.

Centered approximations?

An approximation of the first derivate is given by

$$f_a'(x) = \frac{f(x+h) - f(x-h)}{2h} \tag{129}$$

where h is the distance between two samples and index a denotes that it is an approximation. Please note that his approximation consists to filter $f(x)$ by the following convolution mask

$$\frac{1}{2h}\left[\begin{array}{ccc} -1 & 0 & 1 \end{array}\right] \tag{130}$$

For the second derivate, one possible approximation is

$$f_a''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{131}$$

Computation of the residual error. Let's consider the following Taylor extensions

$$f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2} f''(x) + \ldots + \frac{h^n}{n!} f^{(n)}(x) + \ldots \tag{132}$$

$$f(x-h) = f(x) - h\,f'(x) + \frac{h^2}{2} f''(x) + \ldots + (-1)^n \frac{h^n}{n!} f^{(n)}(x) + \ldots \tag{133}$$

First derivate. By subtraction, member by member, these two equalities, one obtains

$$f(x+h) - f(x-h) = 2h\,f'(x) + \frac{2}{3!} h^3 f^{(3)}(x) + \ldots = 2h\,f'(x) + O(h^3) \tag{134}$$

After re-ordering,

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2) \tag{135}$$

Second derivatee. Like for the first derivate, we use the Taylor extension by add them this time (so we sum up (132) and (133)),

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \frac{2}{4!} h^4 f^{(4)}(x) + \ldots \tag{136}$$

As a result:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2) \tag{137}$$

The $f_a''(x)$ approximation is also of the second order in h.
Synthesis of expressions with a pre-defined error. Another approximation, of order $O(h^4)$, can be built. It corresponds to

$$f_a'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} \tag{138}$$

Spectral behavior of discrete gradient operators I

Consider the one-dimensional continuous function $f(x)$ and the following first derivate:

$$f_a'(x) = \frac{f(x+h) - f(x-h)}{2h} \tag{139}$$

Its Fourier is given by

$$\frac{f(x+h) - f(x-h)}{2h} \rightleftharpoons \frac{e^{2\pi juh} - e^{-2\pi juh}}{2h} \mathcal{F}(u) \tag{140}$$

which can be rewritten as

$$\frac{f(x+h) - f(x-h)}{2h} \rightleftharpoons (2\pi ju) \frac{\sin(2\pi hu)}{2\pi hu} \mathcal{F}(u) \tag{141}$$

where the $(2\pi ju)$ factor corresponds to the ideal (continuous) expression of the first derivate.

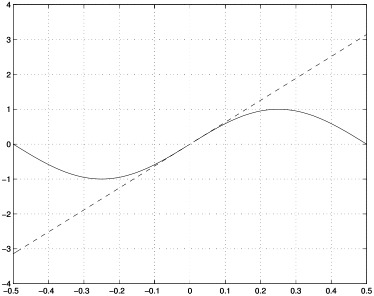
Let us now consider the approximation of the second derivate

$$f_a''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{142}$$

Its Fourier is given by

$$\frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \rightleftharpoons (-4\pi^2 u^2) \left(\frac{\sin(\pi h u)}{(\pi h u)}\right)^2 \mathcal{F}(u) \tag{143}$$

Spectral behavior of discrete gradient operators III



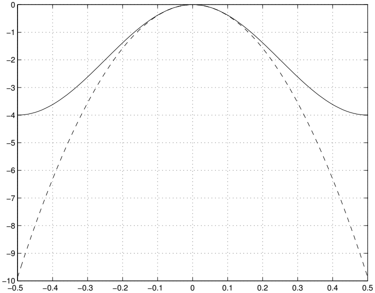


Figure : Spectral behavior of the derivate approximations (for $h = 1$). Left: first derivate, right: second derivate.

Practical expressions of gradient operators and convolution masks I

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \tag{144}$$

corresponds to the following non-centered approximation of the first derivate:

$$\frac{f(x+h,y) - f(x,y)}{h} \tag{145}$$

This "convolution mask" has an important drawback. Because it is not centered, the result is shifted by half a pixel. One usually prefers to use a centered (larger) convolution mask such as

$$\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \tag{146}$$

Practical expressions of gradient operators and convolution masks II

In the y direction, this becomes

$$\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \tag{147}$$

But then, it is also possible to use a diagonal derivate:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{148}$$

Practical expressions of gradient operators and convolution masks III

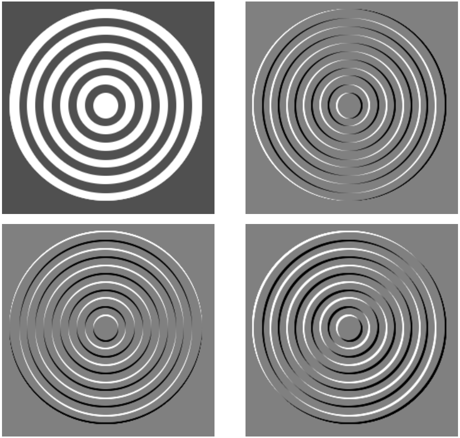


Figure : (a) original image, (b) after the application of a horizontal mask, (c) after the application of a vertical mask, and (d) mask oriented at 135^0.

Practical problems

The use of convolution masks has some drawbacks:

- **Border effects**. Solutions:
  - (i) put a *default* value *outside* the image;
  - (ii) *mirroring extension*: copy inside values starting from the border;
  - (iii) *periodization* of the image –pixels locate on the left are copied on the right of the image,
  - (iv) *copy* border values to fill an artificial added border.
- **The range (dynamic) of the possible values is modified.**
- **It might be needed to apply a normalization factor**.

Prewitt gradient filters

$$[h_x] = \frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \tag{149}$$

$$[h_y] = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{150}$$

Figure : Original image, and images filtered with a horizontal and vertical Prewitt filter respectively.

$$[h_x] = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \qquad (151)$$

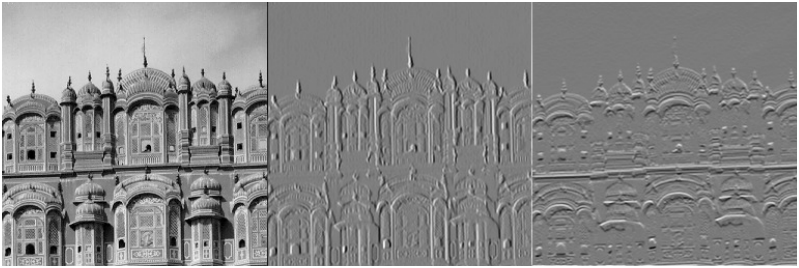


Figure : Original image, and images filtered with a horizontal and vertical Sobel filter respectively.

Second derivate: basic filter expressions

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

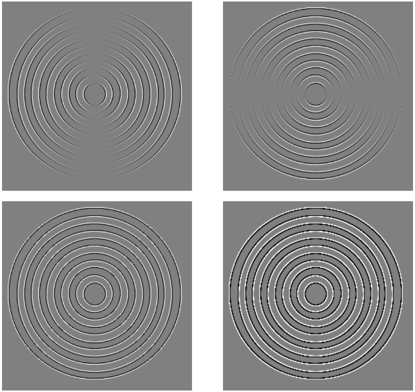


Figure : Results after filtering with the second derivate mask filters.

Morphological gradients

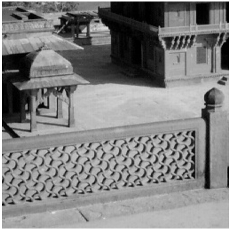
- *Erosion gradient* operator:

$$GE(f) = f - (f \ominus B) \tag{152}$$

- *Dilation gradient* operator:

$$GD(f) = (f \oplus B) - f \tag{153}$$

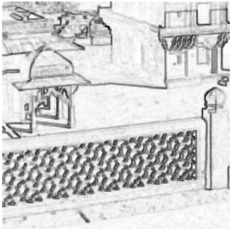
- *Morphological gradient* of Beucher: $GE(f) + GD(f)$.
- *Top-hat* operator: $f - f \circ B$;
- min/max gradient operators: $min(GE(f), GD(f))$, $max(GE(f), GD(f))$
- Non-linear Laplacian: $GD(f) - GE(f)$.



(a) Original image f

(b) $f \oplus B$

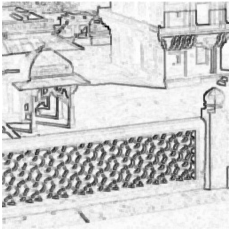
(c) $f \ominus B$



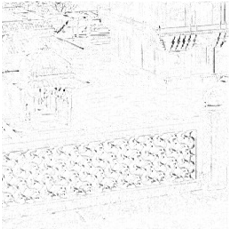
(d) $(f \oplus B) - (f \ominus B)$ (reverse video)

Figure : Gradient of Beucher.

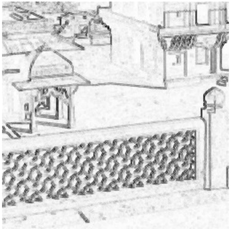
Different non-linear border detectors



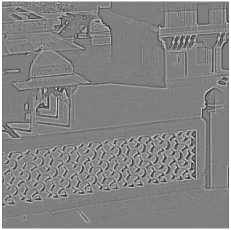
(a) $(f \oplus B) - (f \ominus B)$



(b) $f - f \circ B$ (*top-hat*)



(c) $max(GE(f), GD(f))$



(d) $GD(f) - GE(f)$

Outline

Texture analysis

- Definition?
- Statistical characterization of textures
  - Local mean
  - Local standard deviation
  - Local histogram
  - Co-occurrence matrix of a grayscale image
- Geometrical characterization of textures
  - Spectral approach
  - Texture and energy

Goals of texture analysis?

The major question related to texture are:

- *texture analysis*. The purpose is to characterize a texture by a set of parameters called “texture descriptors”.
- *texture recognition*.
- *image segmentation*.

Definition

Definition

A texture is a signal than can be *extended naturally* outside of his domain.

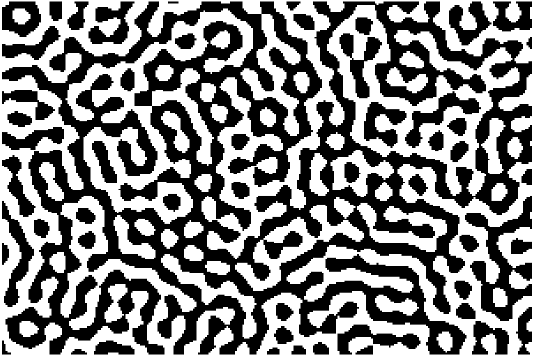
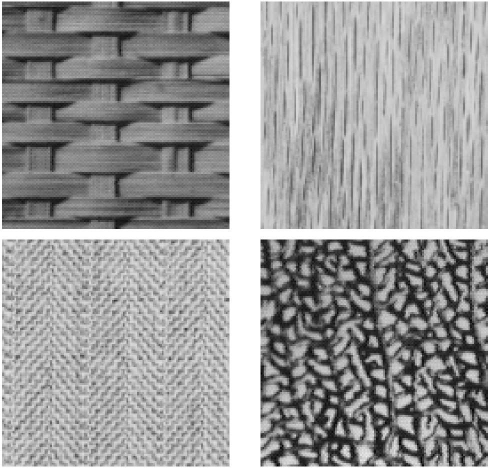


Figure : One possible texture (following Lantuéjoul).

Examples of “real” textures



Simple analysis of a grayscale image

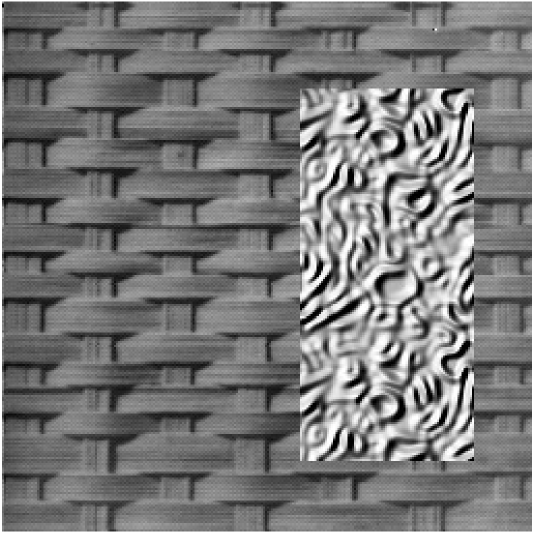


Figure : Example of an image with two textures.

Statistical descriptors of textures

Simple descriptors:

- mean
- variance

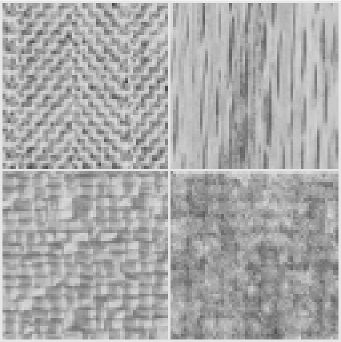
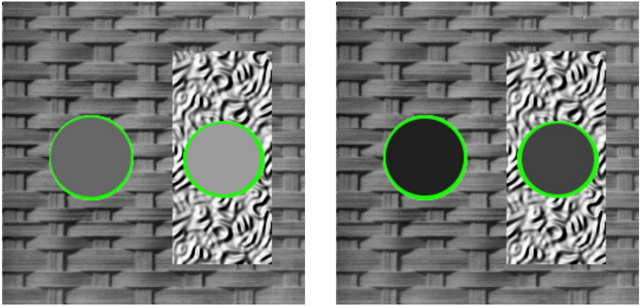


Figure : Textures with identical means and variances.

Statistics defined inside of a local window I



(a) (b)

Figure : Illustration of texture statistics computed over a circle. (a) grayscale mean (103 and 156 respectively) (b) standard deviation (32 and 66 respectively).

Local statistics

Definition

The *local mean* over a spatial window B is defined as

$$\mu_f = \frac{1}{\sharp(B)} \sum_{(x,y)\in B} f(x,y) \tag{154}$$

Definition

The *standard deviation* over a spatial window B is defined as

$$\sigma_f = \sqrt{\frac{\sum_{(x,y)\in B} \left[f(x,y) - \mu_f\right]^2}{\sharp(B)}} \tag{155}$$

Global and local histograms I

Definition

The *histogram* of an image is the curve that displays the frequency of each grayscale level.

Let us consider this image:

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 & 2 \\ 3 & 2 & 1 & 0 & 0 & 0 \\ 3 & 3 & 3 & 3 & 2 & 0 \end{matrix}$$

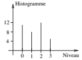


Figure : Non-normalized histogram of an image.

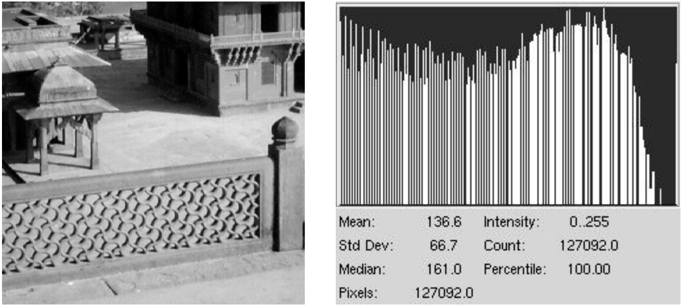


Figure : An image and its global histogram (here B accounts for the whole image domain).

Global and local histograms III

Definition

With a smaller window B, it is possible to define a local (normalized) histogram $p(l)$ as

$$p(l) = \frac{\sharp\left\{(x, y) \in B \,|\, f(x, y) = l\right\}}{\sharp(B)} \tag{156}$$

Histogram statistics

- **Mean**

$$\mu_L = \sum_{l=0}^{L-1} l\, p(l) \tag{157}$$

where L denotes the number of possible grayscale levels inside the B window.

- **Standard deviation**

$$\sigma_L = \sqrt{\sum_{l=0}^{L-1} (l - \mu_L)^2 p(l)} \tag{158}$$

- **Obliquity**

$$S_s = \frac{1}{\sigma_L^3} \sum_{l=0}^{L-1} (l - \mu_L)^3\, p(l) \tag{159}$$

- "**Kurtosis**"

$$S_k = \frac{1}{\sigma^4} \sum_{l=0}^{L-1} (l - \mu_L)^4\, p(l) - 3 \tag{160}$$

Co-occurrence matrix of a grayscale image I

Definition. A co-occurrence matrix is defined by means of a geometrical relationship R between two pixel locations (x_1, y_1) and (x_2, y_2). An example of such a geometrical relationship is

$$x_2 = x_1 + 1 \tag{161}$$

$$y_2 = y_1 \tag{162}$$

for which (x_2, y_2) is at the right of (x_1, y_1).
The co-occurrence matrix $C_R(i, j)$ is squared, with the $L \times L$ dimensions, where L is the range of all possible grayscale values inside of B. Indices of the co-occurrence matrix then indicates the amount of grayscale level value pairs as defined by R.

Co-occurrence matrix of a grayscale image II

Construction of the $C_R(i,j)$ matrix:

1. Matrix initialization: $\forall i,j \in [0,L[\ :\ C_R(i,j) = 0$.
2. Filling the matrix. If the relationship R between two pixels (x_1,y_1) and (x_2,y_2) is followed, then

$$C_R\left(f(x_1,y_1), f(x_2,y_2)\right) \leftarrow C_R\left(f(x_1,y_1), f(x_2,y_2)\right) + 1$$

Let us consider an image with four grayscale levels ($L = 4$, and $l = 0, 1, 2, 3$):

$$f(x,y) = \begin{matrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 3 \\ 2 & 2 & 3 & 3 \end{matrix} \tag{163}$$

$$\begin{aligned} P_{0^o,d}(i,j) &= \sharp\{(x_1,y_1),(x_2,y_2) \in B \mid y_1 = y_2, |x_2 - x_1| = d, \\ & \quad f(x_1,y_1) = i \text{ and } f(x_2,y_2) = j\} \end{aligned} \tag{164}$$

The $P_{0^o,1}$ and $P_{90^o,1}$ matrices are 4×4 matrices respectively given by

$$P_{0^o,1} = \begin{bmatrix} 6 & 2 & 1 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 4 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix} \qquad P_{90^o,1} = \begin{bmatrix} 6 & 1 & 2 & 0 \\ 1 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \tag{165}$$

Use of the Fourier transform

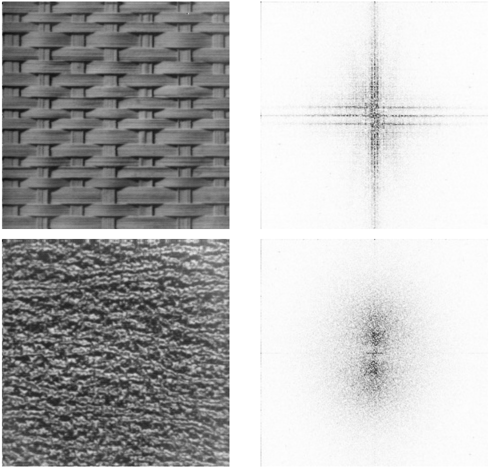


Figure : Spectral characterization of a texture. Right-hand images are the modules of the Fourier transforms.

Textures and energy I

Measures are derived from three simple vectors: (1) $L_3 = (1, 2, 1)$ that computes the *mean*, (2) $E_3 = (-1, 0, 1)$ that detects edges, and (3) $S_3 = (-1, 2, -1)$ which corresponds to the second derivate. By convolving these symmetric vectors, Laws has derived 9 basic convolution masks:

$$\frac{1}{36}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Laws 1

$$\frac{1}{12}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Laws 2

$$\frac{1}{12}\begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix}$$

Laws 3

$$\frac{1}{12}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Laws 4

$$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Laws 5

$$\frac{1}{4}\begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$

Laws 6

$$\frac{1}{12}\begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix}$$

Laws 7

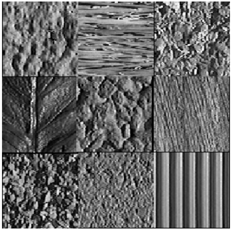
$$\frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix}$$

Laws 8

$$\frac{1}{4}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Laws 9

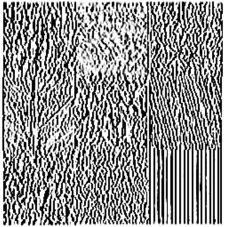
Textures and energy II



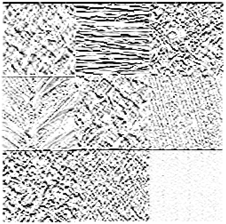
Textures

Laws 3

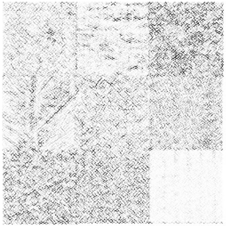
Laws 5



a typical 5 $\times$ 5 filter



Laws 4



Laws 9

Figure : Laws "residues" (reverse video).

Outline

- Problem statement
- Segmentation by thresholding
- Segmentation by region detection (region growing)
    - Watershed

General considerations:

- a very specific problem statement is not always easy.
- chicken-and-egg problem; maybe segmentation is an ill-conditioned problem.

Image segmentation

- Problem statement
- Segmentation by thresholding
- Segmentation by region detection (region growing)
  - Watershed

General considerations:

- a very specific problem statement is not always easy.
- chicken-and-egg problem; maybe segmentation is an ill-conditioned problem.

Problem statement

Definition

Generally, the problem of segmentation consists in finding a set of non-overlapping regions $R_1, \ldots, R_n$ such that

$$\mathcal{E} = \bigcup_{i=1}^{n} R_n \quad \text{and} \quad \forall i \neq j,\, R_i \cap R_j = \emptyset \tag{166}$$

Definition

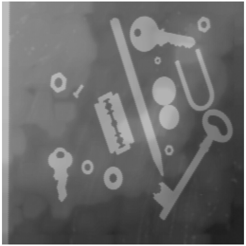
More formally, the segmentation process is an operator ϕ on an image I that outputs, for example, a binary image $\phi(I)$ that differentiates regions by selecting their borders.
An alternative consists to attribute a different label to each pixel of different regions (this is called *region labeling*).

As any similar operator, segmentation can be *local* or *global*. For local segmentation techniques, the results for one given pixel does not impact the segmentation result outside a close neighborhood.

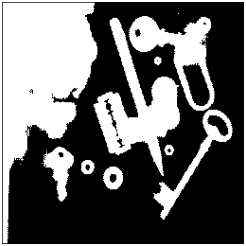
A first typology of segmentation techniques and comparisons

Family of segmentation techniques	input	local/global	markers
Thresholding	image	local (pixel)	no
Watershed	image, gradient, etc	global	yes

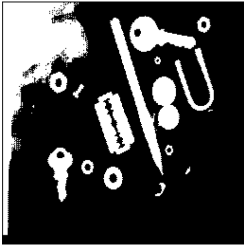
Segmentation by thresholding I



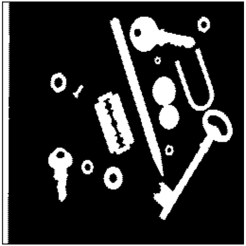
(a) Original image



(b) Thresholding at 110



(c) Thresholding at 128



(d) Thresholding after background equalization

Segmentation by thresholding II

Rationale

There are two contents:

1. "background" pixels
2. "foreground" pixels

Assumptions to solve the segmentation problem:

1. the probability density functions of the two content types are different.
2. one threshold or two thresholds (Otsu's method) are sufficient.

Segmentation by thresholding III

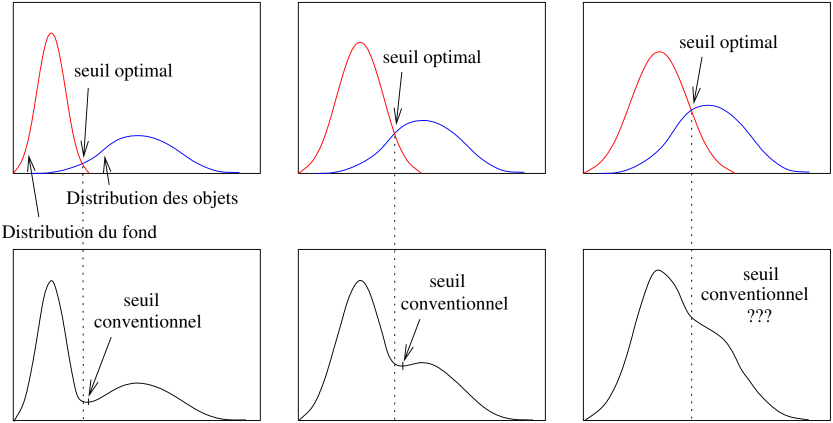


Figure : Optimal threshold.

Segmentation by watershed

In the terms of a topographic surface, a catchment basin $\mathcal{C}(M)$ is associated to every minimum M.

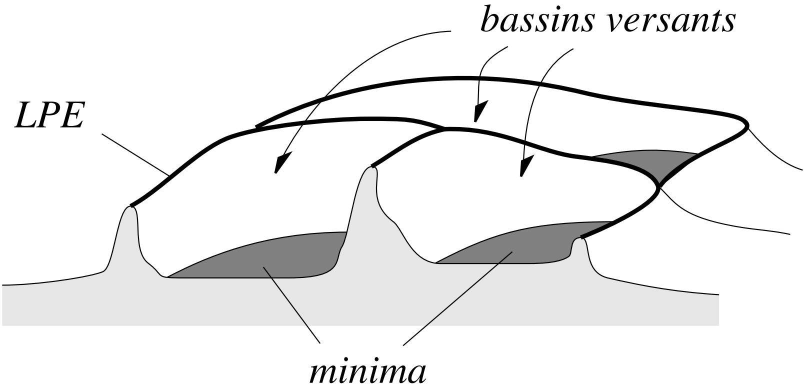


Figure : Minimums, catchment basins, and watershed.

Formal expression and principles of a segmentation algorithm based on the watershed

Approach:

- first, we introduce the case of binary images.
- definition of geodesic path and distance.
- description of an algorithm that handles a stack of thresholded images

Geodesic path

Let X be a binary image.

Definition

[Geodesic path] A geodesic path, of length l, between two points s and t is a series of $l+1$ pixels $x_0 = s, x_1, \ldots, x_l = t$ such that

$$\forall i \in [0, l],\ x_i \in X \ \text{ and } \ \forall i \in [0, l],\ x_{i-1}, x_i \text{ are neighbors} \tag{167}$$

Geodesic distance

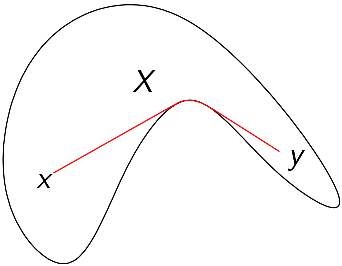


Figure : The shortest path between x and y.

Definition

[Geodesic distance] The geodesic distance between two points s and t is the length of the shortest geodesic path linking s to t; the distance is infinite if such a path does not exist.

Algorithm for the construction of the geodesic skeleton by growing the zone of influence

Notations

The zone of influence of a set Z_i, is denoted by $ZI(domain = \text{X}, center = Z_i)$ and its frontier by $FR(domain = \text{X}, center = Z_i)$.

The skeleton by zone of influence (SZI) is obtained via the following algorithm:

- first, one delineates the Z_i zones of each region;
- for remaining pixels, an iterative process is performed until stability is reached: if a pixel has a neighbor with an index i, then this pixel gets the same index; pixels with none or two different indices in their neighborhood are left unchanged;
- after all the iterations, all the pixels (except pixels at the interface) are allocated to one region of the starting regions Z_i.

Example

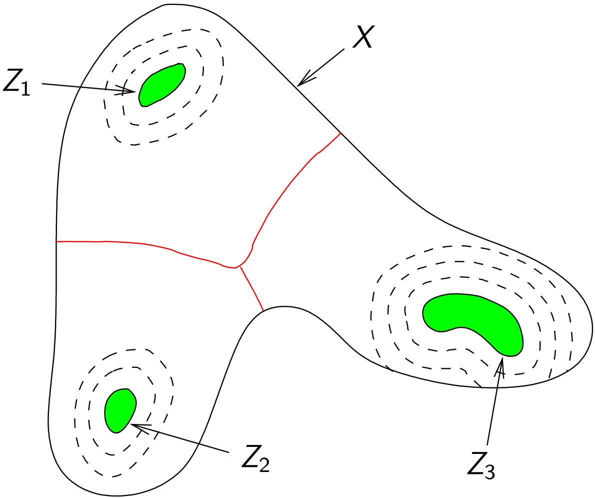


Figure : Geodesic skeleton.

The case of grayscale images (a gradient image for example) I

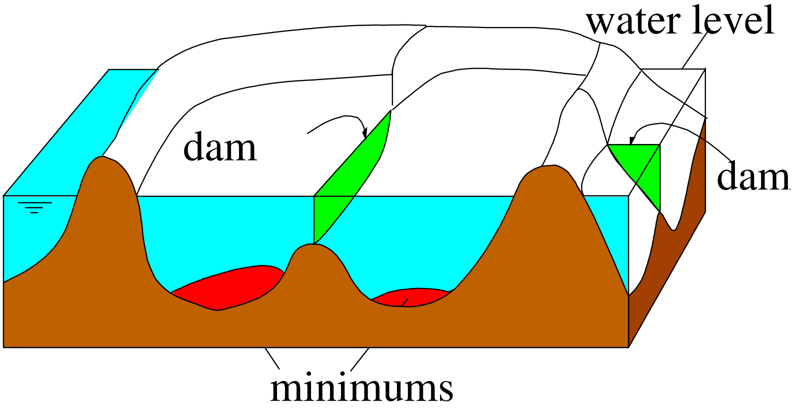


Figure : A dam is elevated between two neighboring catchment basins.

The case of grayscale images (a gradient image for example) II

Notations:

- f is the image.
- h_{min} and h_{max} are the limits of the range values of f on the function support (typically, $h_{min} = 0$ and $h_{max} = 255$).
- $T_h(f) = \{x \in dom\, f : f(x) \leq h\}$ is a set obtained by thresholding f with h. For h growing, we have a stack of decreasing sets.
- M_i are the minimums and $\mathcal{C}(M_i)$ are the catchment basins.

Let $\mathcal{C}_h(M_i)$ be the subset of the M_i basin filled at "time" (or "height") h. Then

$$\mathcal{C}_h(M_i) = \mathcal{C}(M_i) \cap T_h(f) \tag{168}$$

In this expression, $\mathcal{C}(M_i)$ is unknown.

Initialization:

- $\mathcal{C}_{h\ min}(M) = T_{h\ min}(f)$; the initialization considers that all the local minimums are valid catchment basin originators.

Construction

$$\forall h \in [h_{min} + 1, h_{max}] : \mathcal{C}_h(M) = ZI_h \cup Min_h \tag{169}$$

with

- $ZI_h =$ influence zone (with domain $T_h(f)$);
- Min_h is the set of all the points of $T_h(f)$ that have no label after the growing process of influence zones. They correspond to minimums that are introduced at level h.

Markers

Marking is a process that allows to select only some of the local minimums.
Watershed has the following advantages with respect to other techniques (such as thresholding):

- the possibility to be applicable to *any sort of input image* (original image, gradient, etc),
- the flexibility to put some **markers** to select only a few local minimums. With markers, the amount of regions is exactly equal to the number of markers put in the image.

Outline

Motion analysis

There are basically two "pure" approaches to motion analysis in a video sequence:

1. Approach by *tracking* (= *motion estimation* based techniques):
   - detects some particular points in a video frame.
   - find the corresponding points in the next frame.
   - based on a model, interpret the trajectories of the points (usually at the object level).
2. Approach by *background subtraction*:
   - build a reference frame or model with no foreground in it.
   - compare a next frame to the reference.
   - update the reference.

Motion analysis by tracking: principles

There are several techniques but, usually, they involve the following steps:

1. detect features in successive frames.
2. make some correspondences between the features detected in consecutive frames
3. based on a model, regroup some features to facilitate tracking objects.

Feature detection

Some known feature detectors:

- Harris's corner detector
- Scale Invariant Feature Transform (SIFT)
- Speeded Up Robust Features from an image (SURF)
- Features from Accelerated Segment Test (FAST)
- ...

Original image

Features detected by SURF

Feature correspondence

Difficulties for feature correspondence

Typical questions/difficulties for tracking approaches (targeting *motion estimation*):

- How to filter the features? (remove some useless features)
- How do we regroup features?
  - we need a *model*. But this introduces a **bias towards the model**.
- How do we ensure continuity over time?
- What happens when occlusions occur?
- ...

Introduction to motion analysis by background subtraction I

- Objective: separate the foreground (pixels "in motion") from the background ("static" pixels).
- Steps:

[Initialization] build a *reference frame* or a *model* for the background.

[Subtraction] *compare* the current frame to the reference frame or model, and "*subtract*" the frame to get a binary image indicating pixels who have changed.

[Updating] *update* the reference frame or model.

Introduction to motion analysis by background subtraction II

One frame in the sequence

Built reference frame

Figure : Building a reference frame or a model.

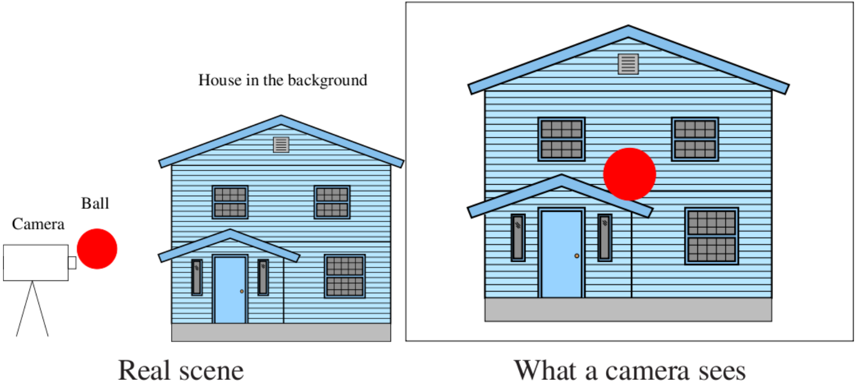
Introduction to motion analysis by background subtraction III

Original image

Features detected by ViBe

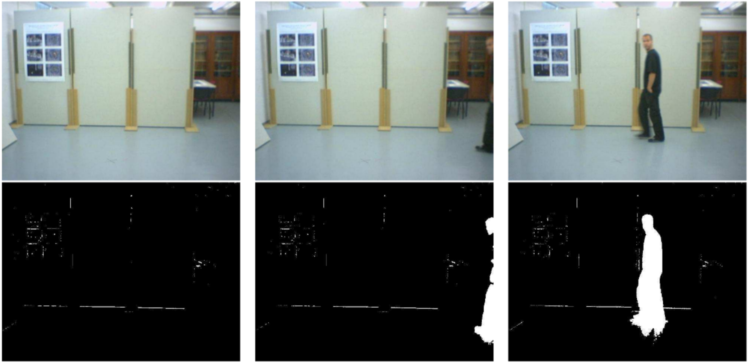
Figure : Segmentation by background subtraction.

Principles



- Major assumption: fixed camera

Example



- Any application with moving objects
- Video-surveillance

Elementary method

Naive approach (static background)

Foreground is detected, pixel by pixel, as the difference between the current frame and a static reference image (background):

$$|I_t - B| > \text{threshold} \tag{170}$$

where

- I_t is the current pixel value (at time t),
- B is the reference background value for that pixel.

Problems:

- How do we choose the reference image?
- What's the best threshold?

Problem with the threshold

Classical processing chain

pre-processing of an input frame

$\Downarrow$

segmentation map after background processing

$\Downarrow$

post-processing of the segmentation map

Typical post-processing operations are:

- morphological filtering (cleaning): erosion, dilation, opening, area opening
- median filtering
- analysis of connected components
- shadow removal
- ...

Classical processing chain

pre-processing of an input frame

$\Downarrow$

segmentation map after background processing

$\Downarrow$

post-processing of the segmentation map

Typical post-processing operations are:

- morphological filtering (cleaning): erosion, dilation, opening, area opening
- median filtering
- analysis of connected components
- shadow removal
- ...

Choice for a reference image

Simple techniques

- One “good” image is chosen (usually a frame empty of foreground objects).
- Exponentially updated reference image

$$B_t = \alpha I_t + (1 - \alpha) B_{t-1} \tag{171}$$

  Typical value for α: 0.05
- Median of the last N frames.

Important choice:

- *conservative update* (only when the pixel belongs to the background) or not (the pixel belongs to the foreground or the background).

Advanced techniques

The background is modelled as a probability density function to be estimated

- One gaussian distribution per pixel
- *Mixture of gaussians for each pixel*
- *Kernel based estimation of the probability* density function

One gaussian per pixel

For each pixel, the probability density function of observed values is modeled by a **single** gaussian.

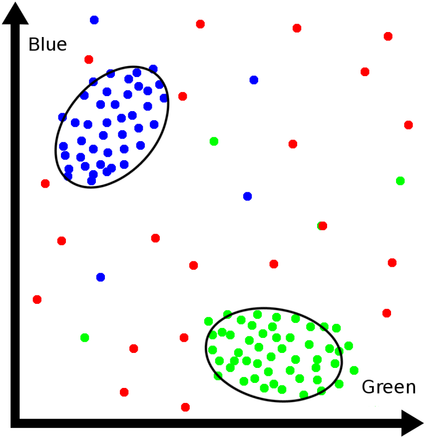
Once the model is built (here it means that we need to estimate the mean and variance), we evaluate the distance to the mean. If

$$|I_t - \mu| \leq \text{threshold} \times \sigma \tag{172}$$

then the pixel belongs to the background.

Mixture of Gaussians Model

Motivation: the probability density function of the background is multi-modal



[Stauffer, 1999, 2000] [Power, 2002] [Zivkovic, 2006]

Mixture of Gaussians Model

For each pixel:

$$P(X) = \sum_{i=1}^{N} \alpha_i N(\mu_i, \sigma_i) \tag{173}$$

Typical values for n: 3 or 5

Fundamental assumptions

- The background has a low variance.
- The background is more frequently visible than the foreground.

Kernel Density Estimation (KDE) methods

For each pixel:

$$P(X) = \sum_{i=1}^{N} \alpha_i K_\sigma(X - X_i) \quad (174)$$

where $\{X_i\}_{i=1,\ldots,N}$ are the N last values observed (samples) for that pixel, and $K_\sigma()$ is a kernel probability function centered at X_i.

Decision rule

The pixel belongs to the background if $P(X) > threshold$.

[Elgammal, 2000] [Zivkovic, 2006]

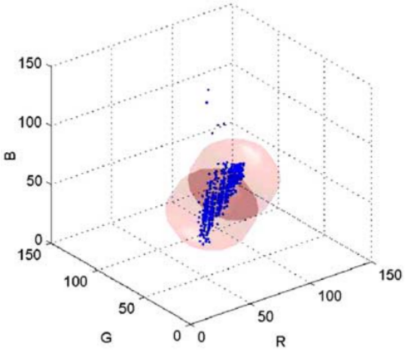
Parameters of KDE techniques

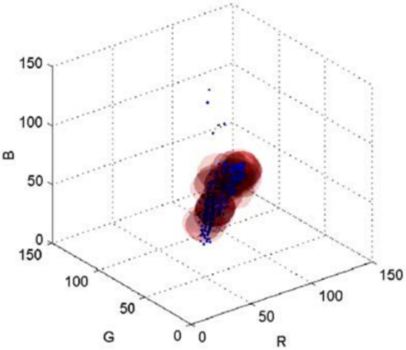
Typical values

$P(X) = \sum_{i=1}^{N} \alpha_i K_\sigma(X - X_i)$ with

- Number of samples: $N = 100$
- Weight: $\alpha_i = \alpha = \frac{1}{N}$
- Spreading factor: $\sigma = \text{Variance}(X_i)$
- Probability density function chosen to be gaussian: $K_\sigma(X - X_i) = N(X_i, \sigma^2)$

GMM techniques vs KDE techniques





Challenges

- Input related issues
  - lighting changes
    - slow (day/night cycles)
    - fast (light switch, clouds in the sky, ...)
  - unwanted motions
    - camera shaking (wind)
    - in the background (tree leaves, waving grass, water)
  - appearance changes of foreground objects (reflections), shadows, camouflage, ...
- Implementation related issues
  - Robustness
  - Real time

Outline

Outline