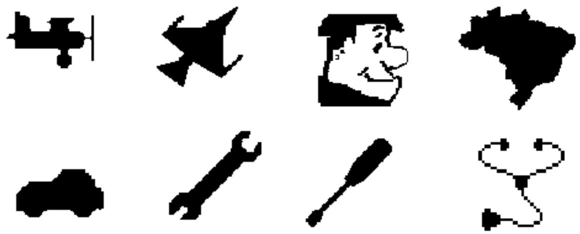
Shape Analysis and Classification

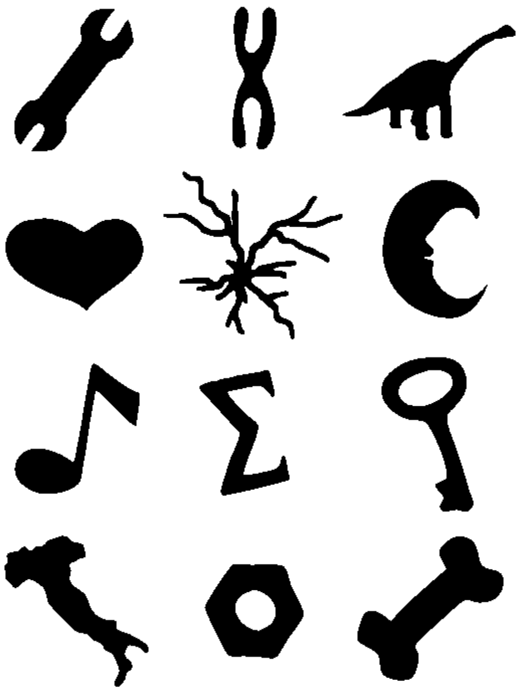


Luciano da Fontoura Costa

Roberto M. Cesar-Jr

http://www.vision.ime.usp.br/~cesar/shape/

Shape Analysis and Classification

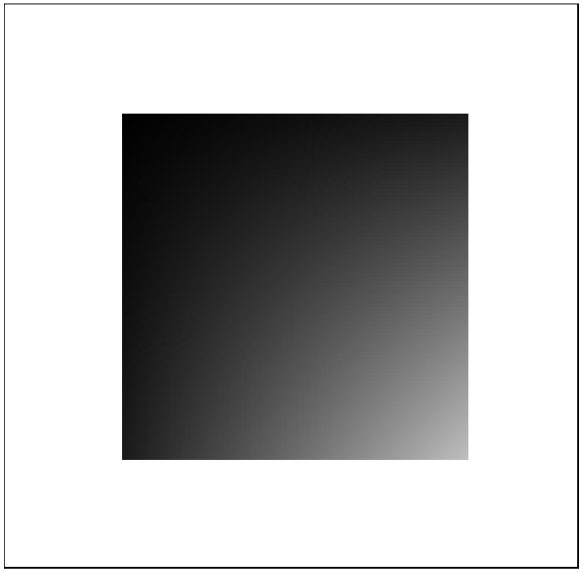


SHAPE ACQUISITION AND PROCESSING

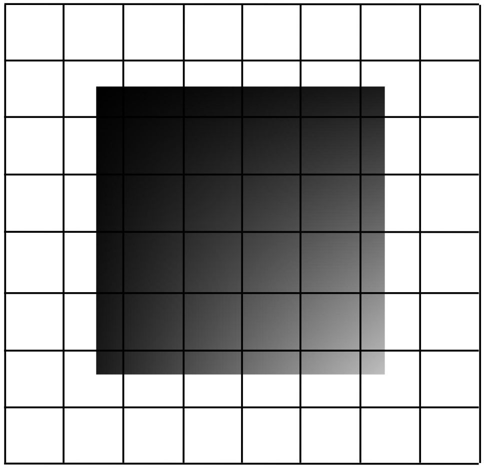
Introduction

- Image acquisition
- Image formation
- Image processing
  - Enhancement
  - Noise filtering
  - Edge detection
  - Image segmentation

Image formation

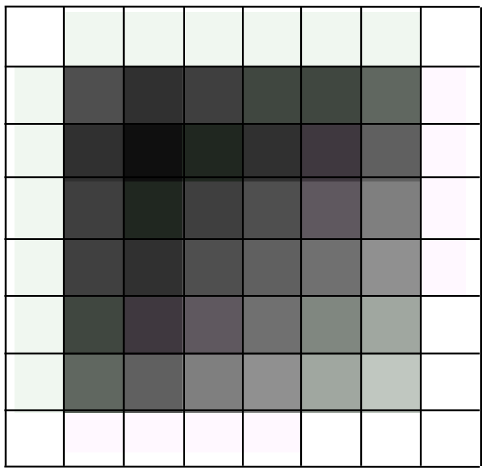


Original Image



Sampling

Image formation



Sampled image

255	255	255	255	255	255	255	255
255	82	26	40	49	50	98	255
255	26	0	31	40	42	76	255
255	40	31	56	59	77	106	255
255	49	40	59	68	103	125	255
255	50	42	77	103	124	146	255
255	98	76	106	125	146	177	255
255	255	255	255	255	255	255	255

Quantization

Pixels

Image formation

$$g = \begin{bmatrix} 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 82 & 26 & 40 & 49 & 50 & 98 & 255 \\ 255 & 26 & 0 & 31 & 40 & 42 & 76 & 255 \\ 255 & 40 & 31 & 56 & 59 & 77 & 106 & 255 \\ 255 & 49 & 40 & 59 & 68 & 103 & 125 & 255 \\ 255 & 50 & 42 & 77 & 103 & 124 & 146 & 255 \\ 255 & 98 & 76 & 106 & 125 & 146 & 177 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

Digital image representation as an array

Image formation: sampling

Image formation: sampling

Image formation: sampling

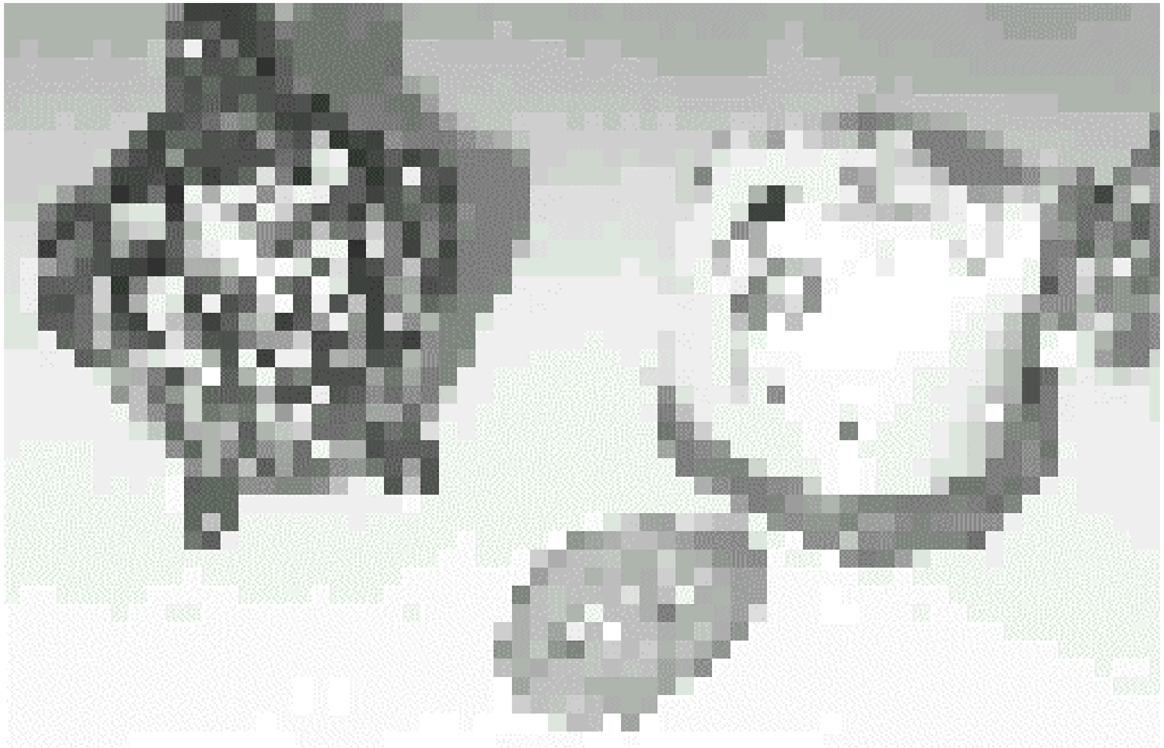
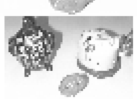


Image formation: sampling

(a)

(b)

(c)

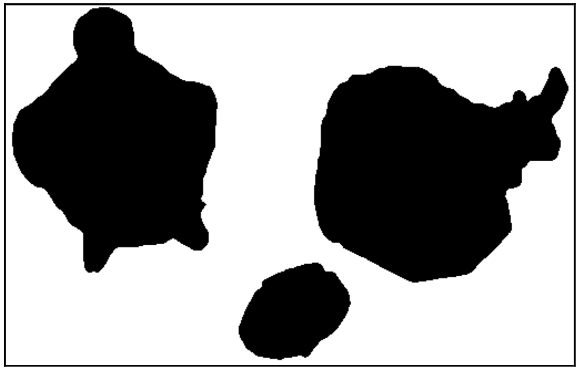


(d)

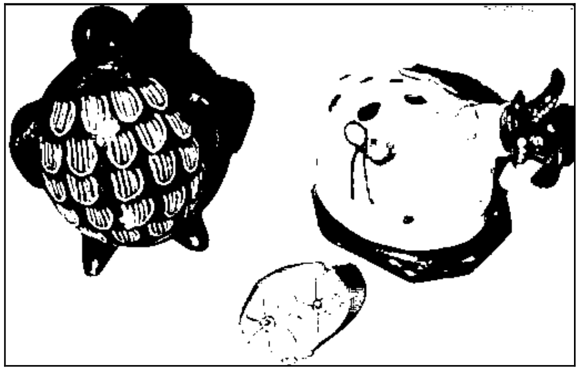
Image pyramid

Image segmentation

(a)



(b)

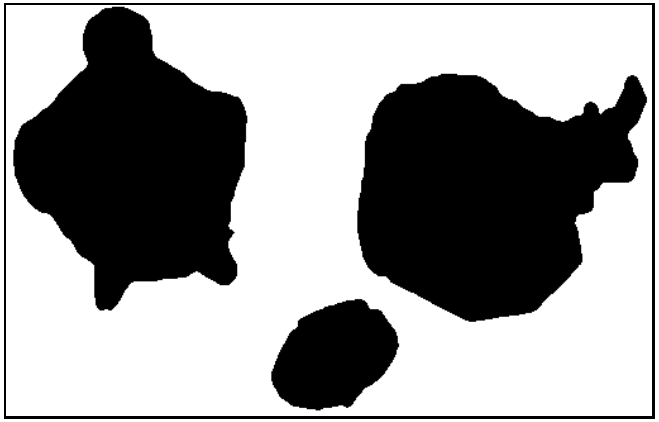


(c)

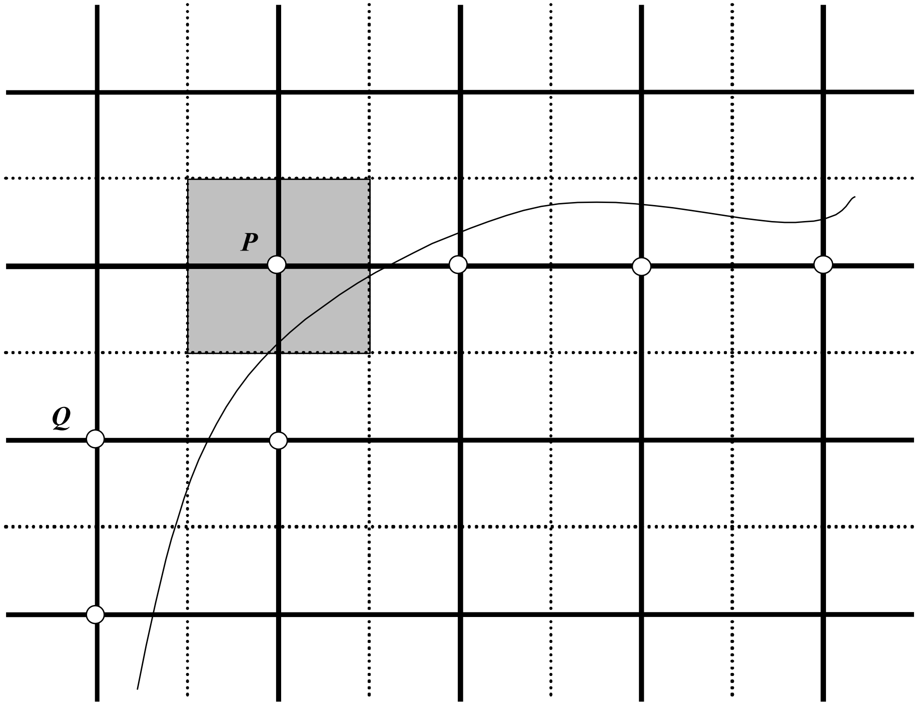
Image segmentation

[1] : *Pixel value conventions.*

Gray level images	Binary images
0 (dark)	0 = white = background
255 (bright)	1 = black = foreground

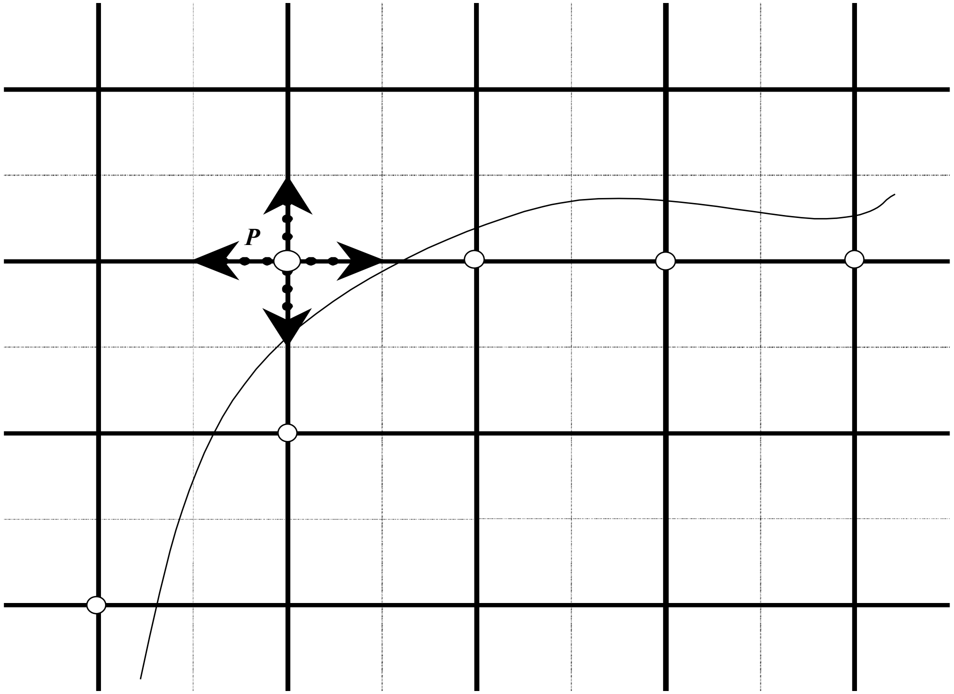


Shape sampling



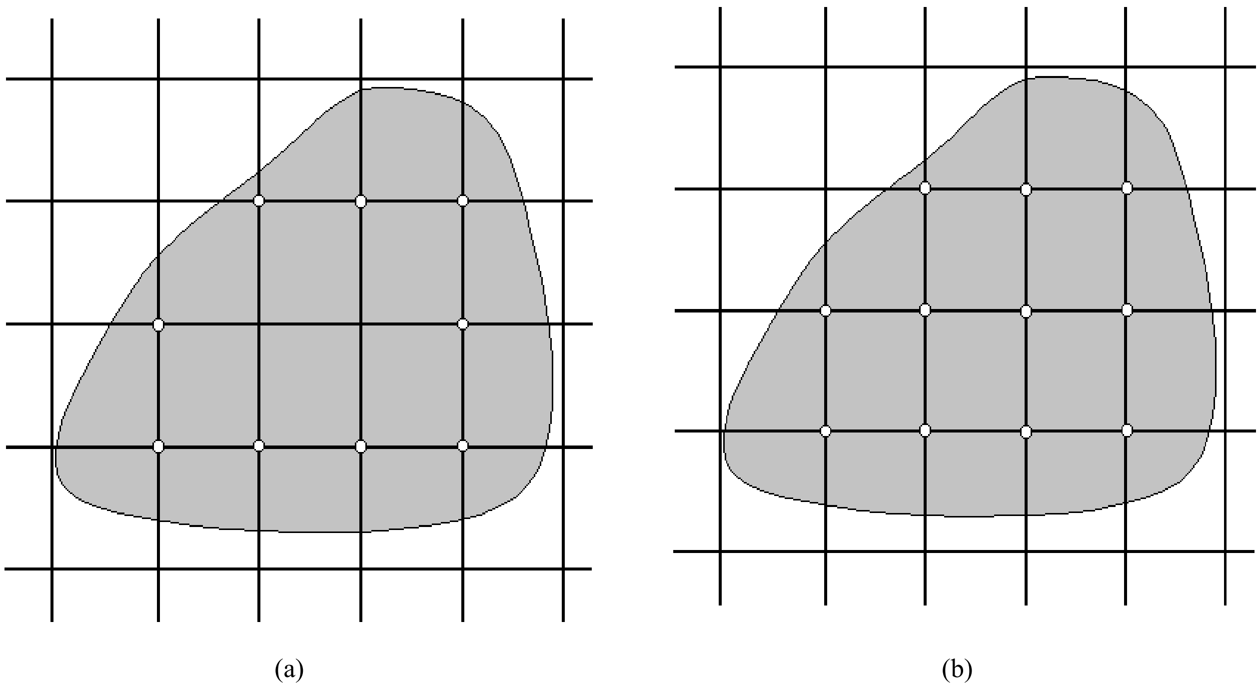
1 : *The square-box quantization scheme (SBQ).*

Shape sampling



1 : *The grid intersect quantization scheme (GIQ).*

Shape sampling



The object boundary quantization scheme (OBQ) of a thick shape (a) and the whole quantized shape after region-filling (b).

Discrete geometry concepts

- Pixel neighborhood
- 4-neighborhood and 8-neighborhood
- Connected path
- Connected component
- Labeling algorithm

Discrete geometry concepts

$$
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0\\
0&0&0&1&0&0&1&1&1&0\\
0&0&0&0&0&0&1&1&1&0\\
0&0&1&1&1&1&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&1&1&1&1&1&0\\
0&0&0&0&0&0&0&1&1&0
\end{bmatrix}
$$

(a)

$$
\begin{array}{c}
\\
\mathbf{A}\\
\\
\mathbf{B}\\
\\
\\
\\
\\
\\
\\
\end{array}
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0\\
0&0&0&1&0&0&1&1&1&0\\
0&0&0&0&0&0&1&1&1&0\\
0&0&1&1&1&1&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&1&1&1&1&1&0\\
0&0&0&0&0&0&0&1&1&0
\end{bmatrix}
$$

(b)

$$
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0\\
0&0&0&1&0&0&1&1&1&0\\
0&0&0&0&0&0&1&1&1&0\\
0&0&1&1&1&1&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&1&1&1&1&1&0\\
0&0&0&0&0&0&0&1&1&0
\end{bmatrix}
$$

(c)

$$
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0\\
0&0&0&1&0&0&2&2&2&0\\
0&0&0&0&0&0&2&2&2&0\\
0&0&3&3&3&3&0&0&0&0\\
0&0&3&3&3&0&0&0&0&0\\
0&0&0&0&0&4&4&0&0&0\\
0&0&0&0&4&4&4&4&4&0\\
0&0&0&0&0&0&0&4&4&0
\end{bmatrix}
$$

(d)

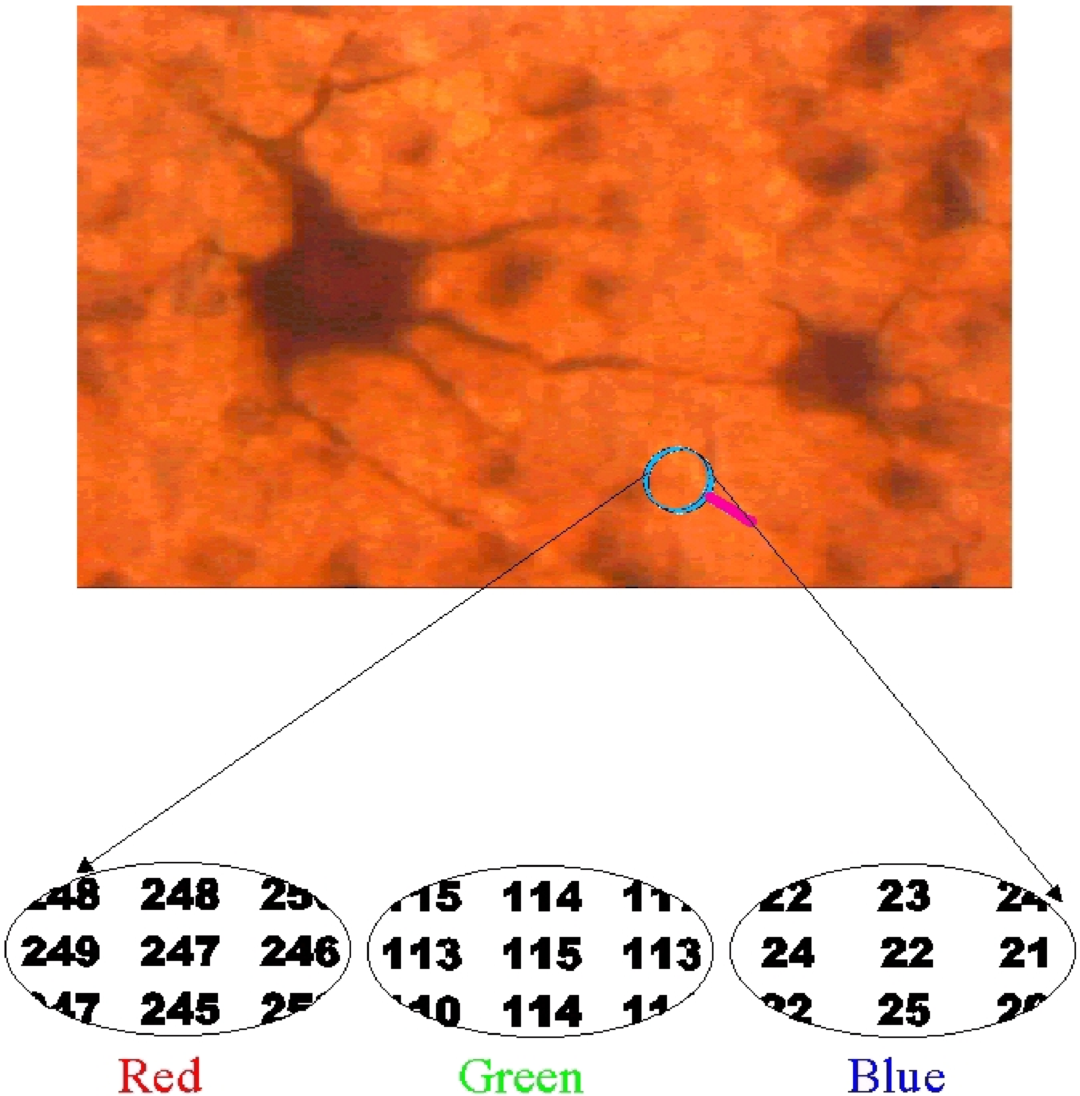
$$
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0\\
0&0&0&1&0&0&1&1&1&0\\
0&0&0&0&0&0&1&1&1&0\\
0&0&1&1&1&1&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&1&1&1&1&1&0\\
0&0&0&0&0&0&0&1&1&0
\end{bmatrix}
$$

(e)

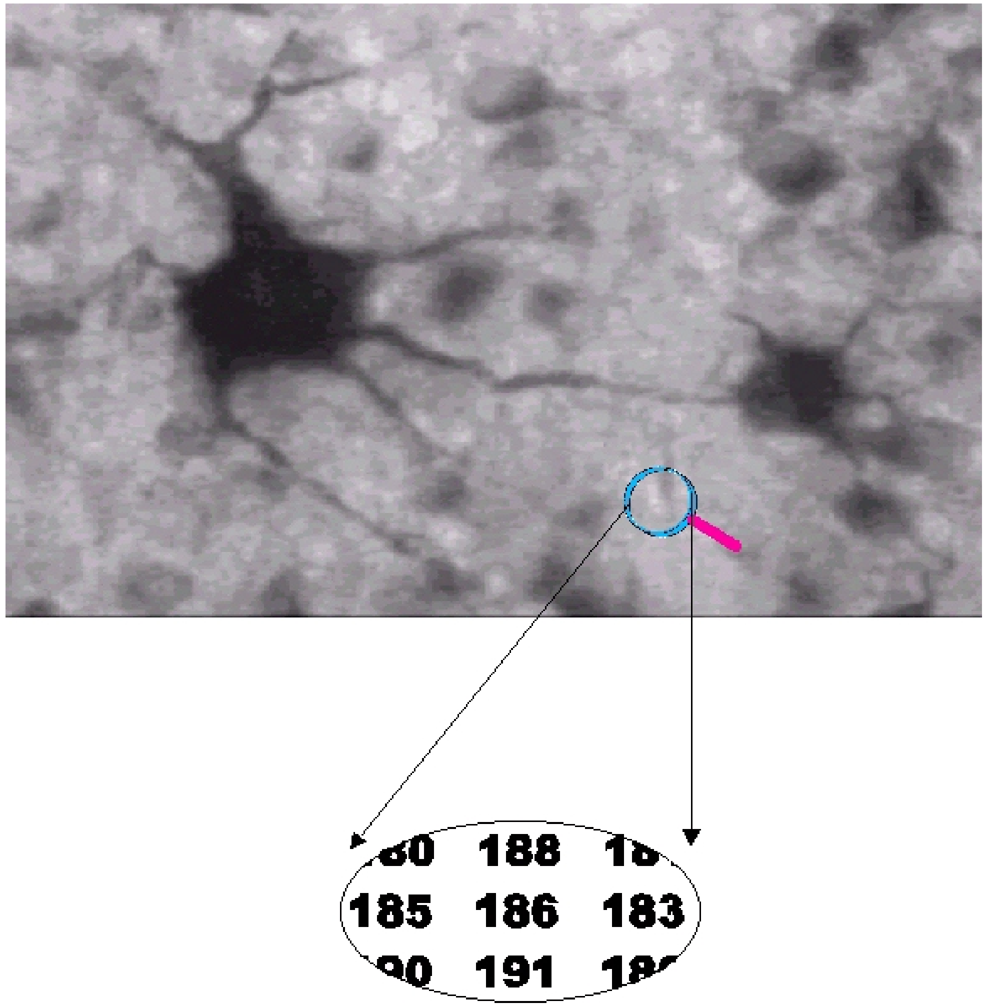
$$
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0\\
0&0&0&1&0&0&2&2&2&0\\
0&0&0&0&0&0&2&2&2&0\\
0&0&2&2&2&2&0&0&0&0\\
0&0&2&2&2&0&0&0&0&0\\
0&0&0&0&0&2&2&0&0&0\\
0&0&0&0&2&2&2&2&2&0\\
0&0&0&0&0&0&0&2&2&0
\end{bmatrix}
$$

(f)

Color images



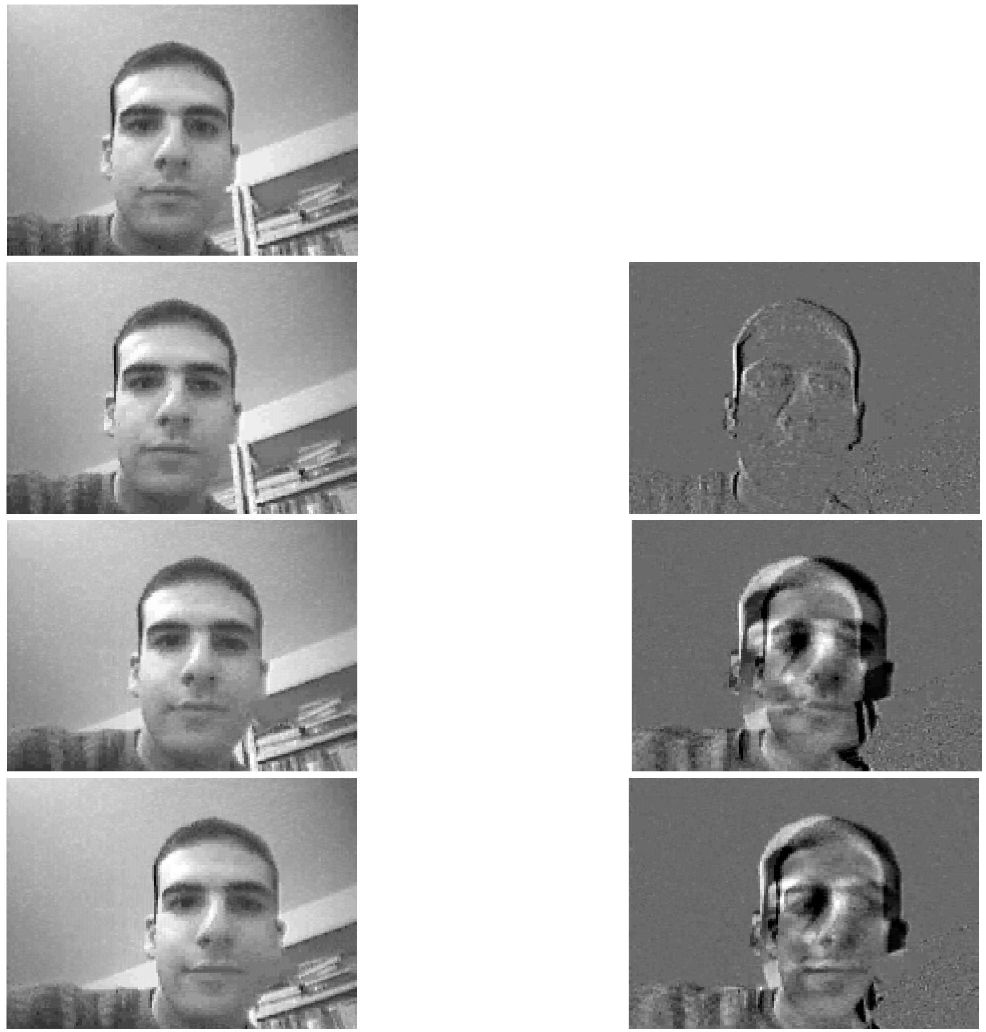
Gray scale images



Gray-Level

Video sequences

Video sequences



Many other image types...

- Multispectral images
- Voxels
- Range
- 3D: geometry and texture
- 3D sequences
- Log-polar (foveal)
- ...

Image processing

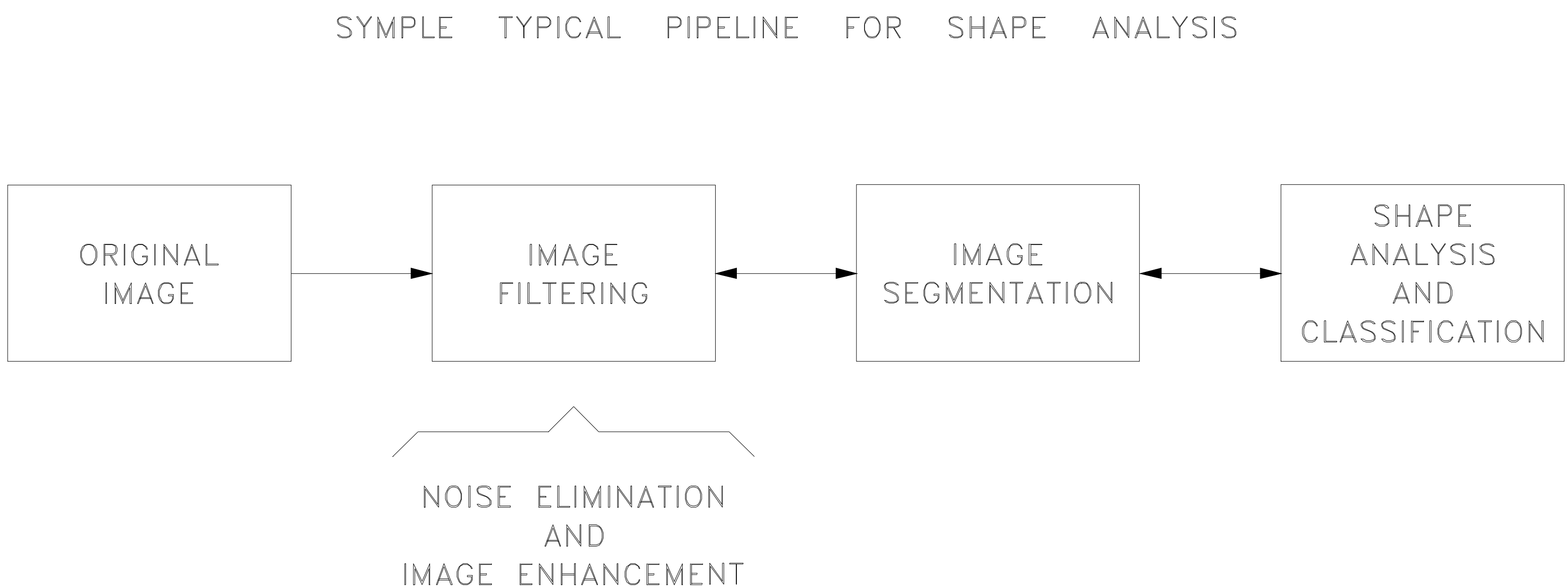
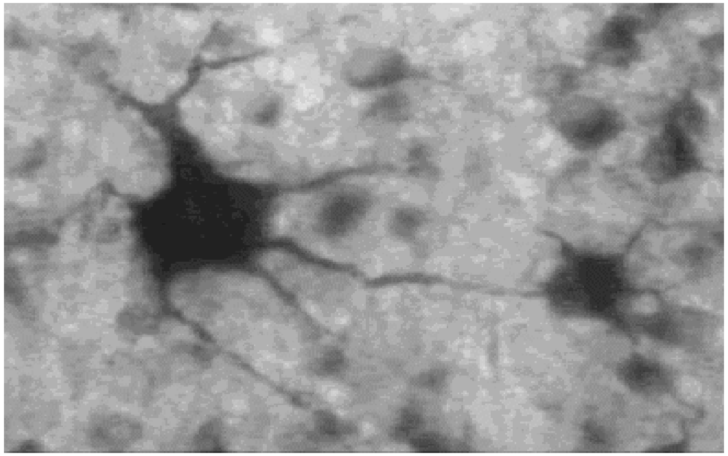
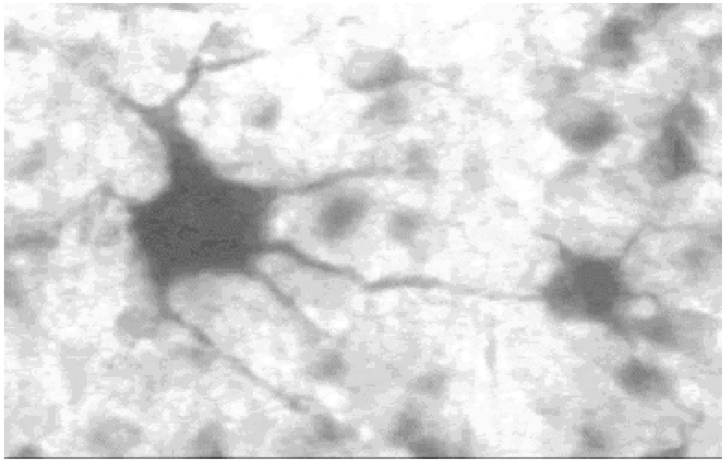


Image processing





Simple image processing operation: adding a constant

Image histograms

Image histograms

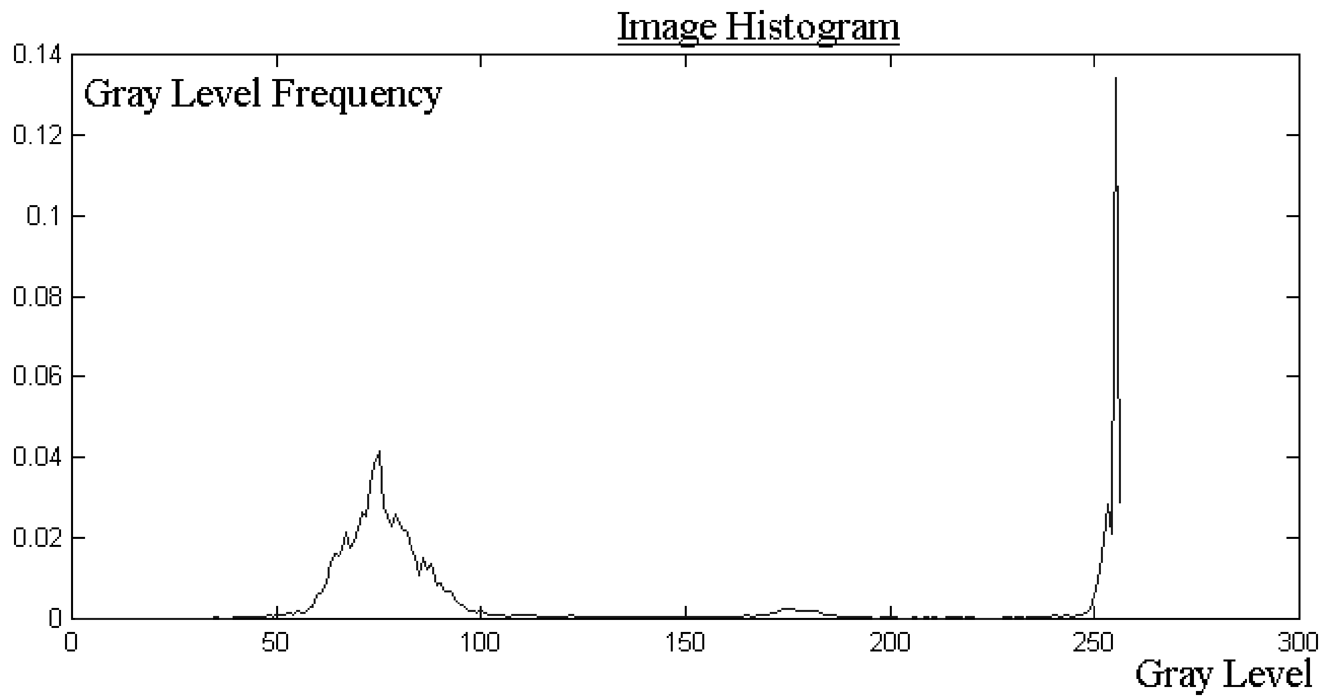


Image histograms

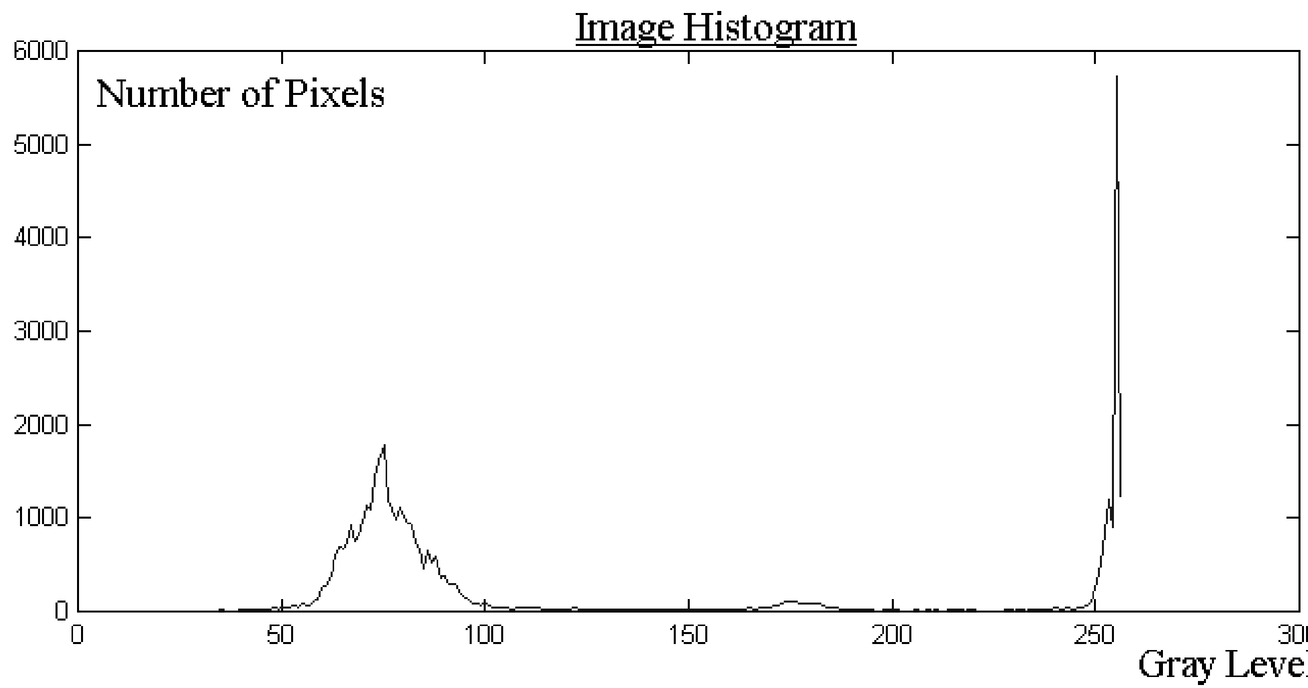


Image histograms

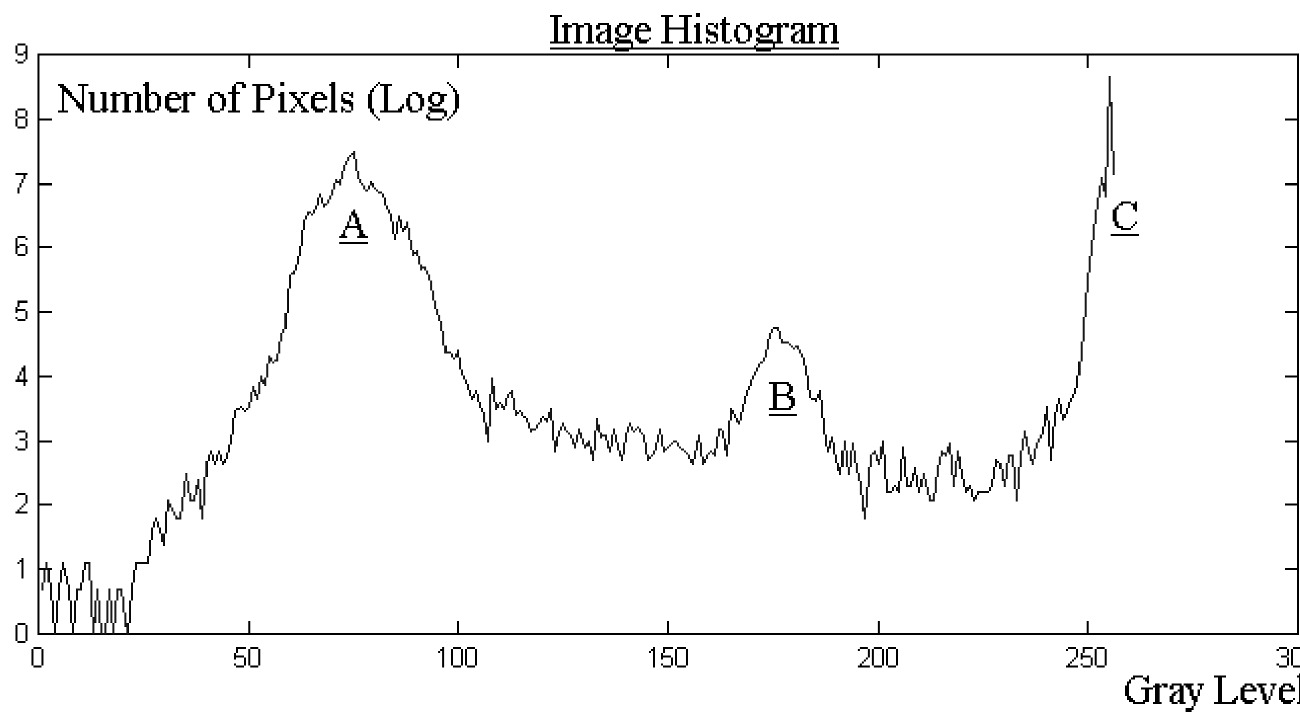


Image histograms

Image histograms

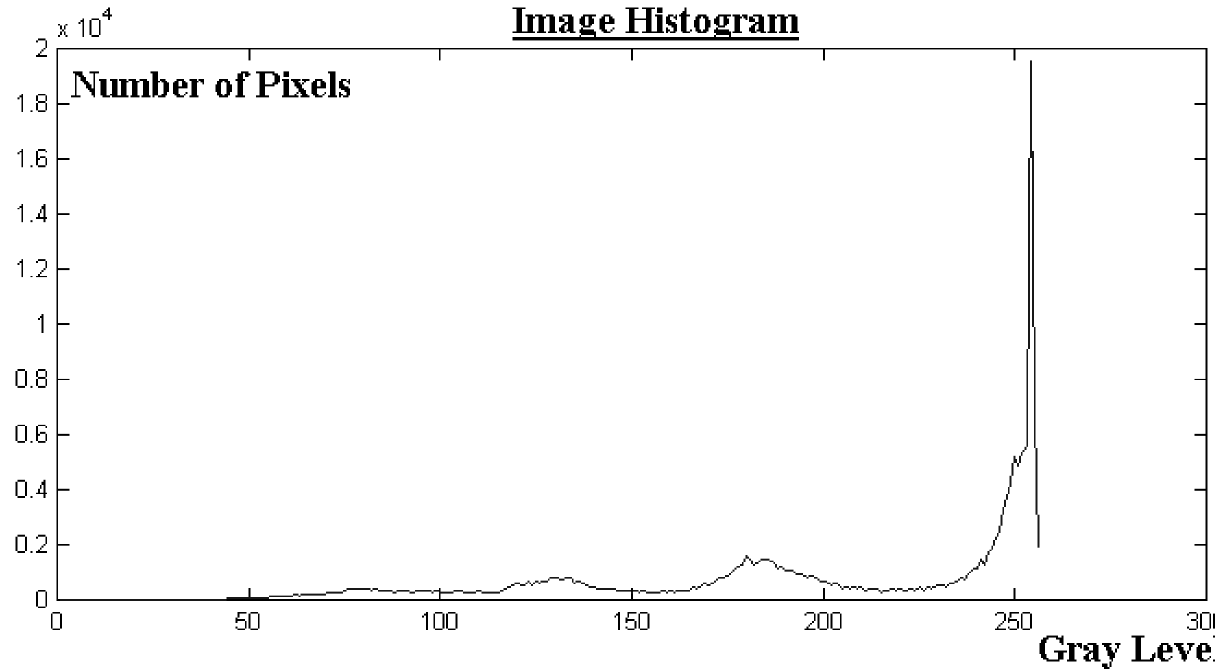
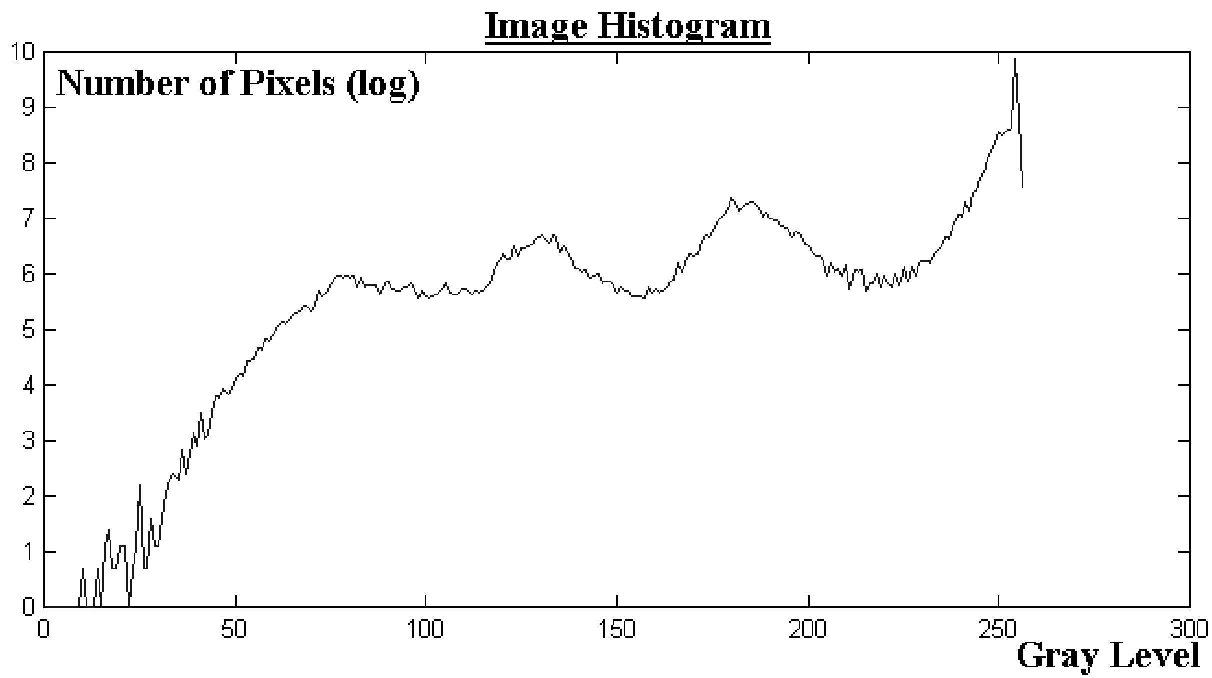


Image histograms



Histogram equalization

$$f(n) = (N-1)\sum_{k=0}^{n} h(k)$$

Algorithm: ***Histogram Equalization***

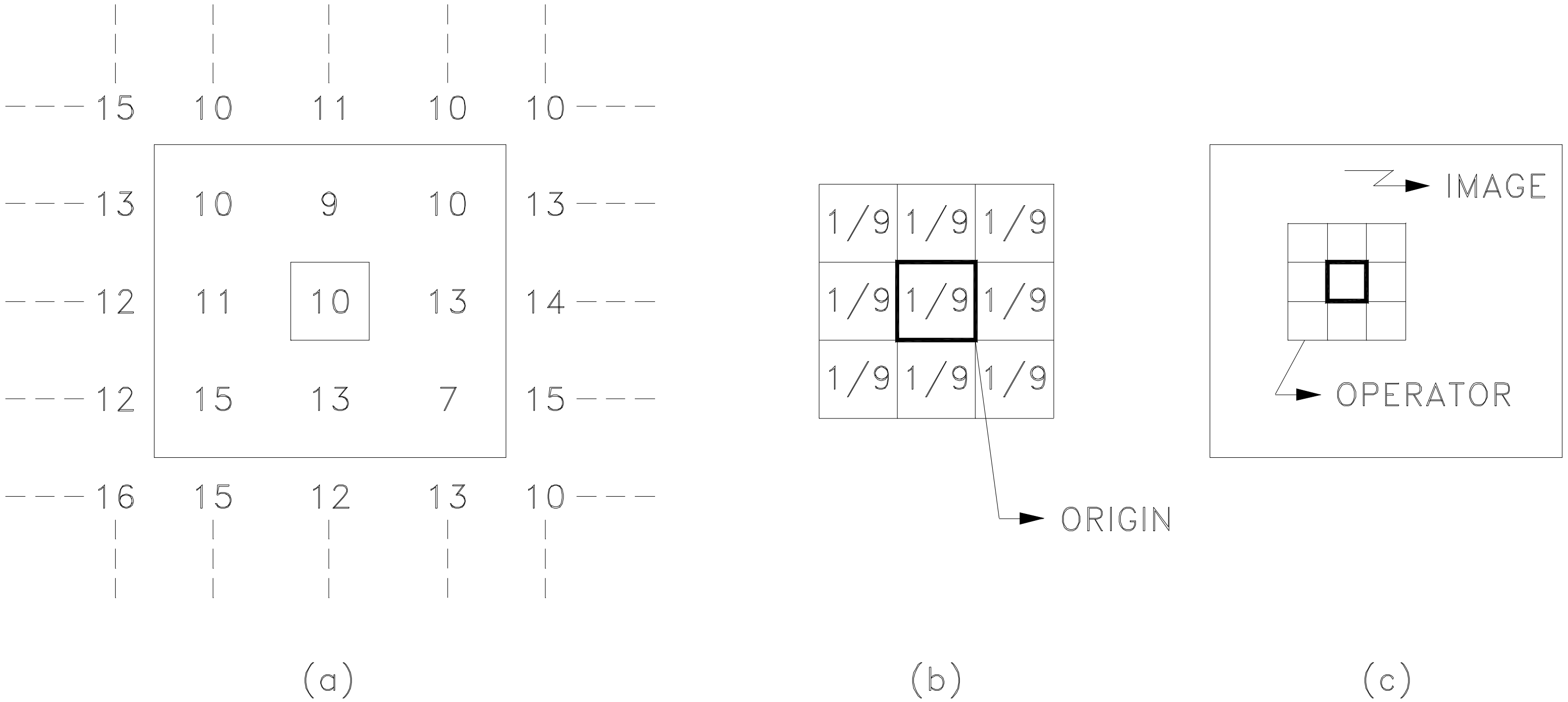
```
h = histogram2(g, N);
For n = 0 to N-1 do
   f(n) = round(sum(h, n) * (N-1)) ;
For each pixel g(p, q) do
   i(p, q) = f(g(p, q));
```

Histogram equalization

Local processing

- Pixel neighborhood
- Local properties
- Scale
- Image processing:
  - Filtering
  - Feature detection
  - Edge detection

Local processing



Local processing

Local processing

Local processing

Local processing

Local processing

- Smoothing
- 3 X 3 window
- Weights
- Window origin
- Window size: scale
- Problems with border pixels

Local processing

- Particular case: *linear filtering*
- Convolution:

$$f(p,q)=\frac{1}{MN}\sum_{m}\sum_{n}h(m,n)g(p-m,q-n)$$

- Many different filters may be implemented as convolutions

Convolution

- Let $g(t)$ and $h(t)$ be two real or complex functions. The *convolution* is defined as:

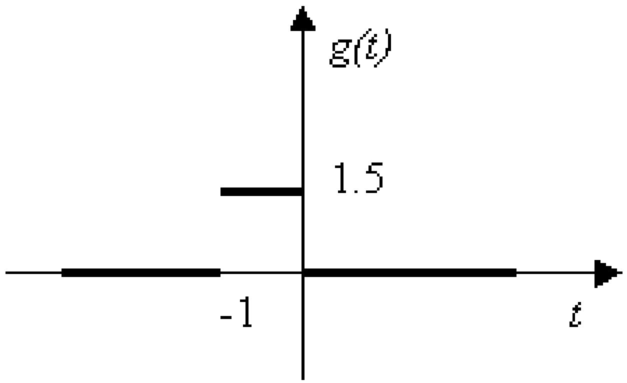
$$q(\tau) = g(\tau) * h(\tau) = (g * h)(\tau) = \int_{-\infty}^{\infty} g(t)\, h(\tau - t)\, dt$$

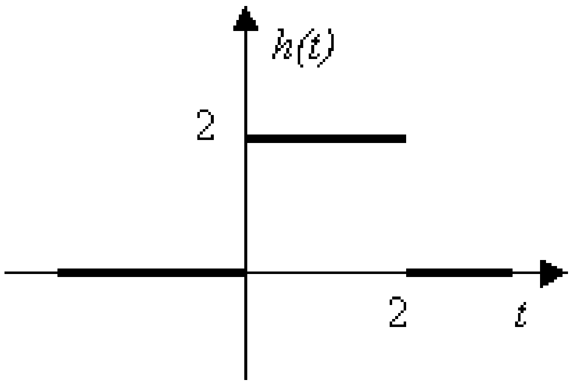
Convolution

- Example:

$$g(t) = \begin{cases} 1.5 & \text{if } -1 < t \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

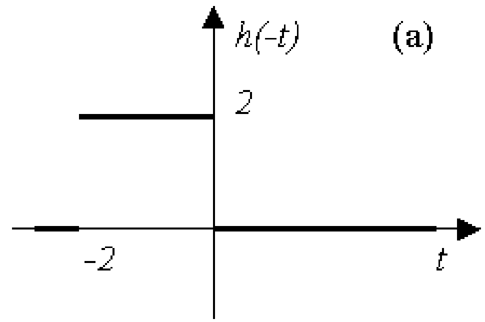
$$h(t) = \begin{cases} 2 & \text{if } 0 < t \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

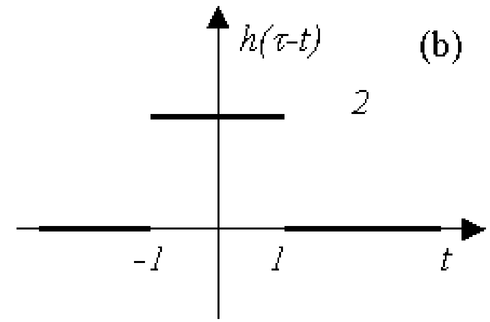


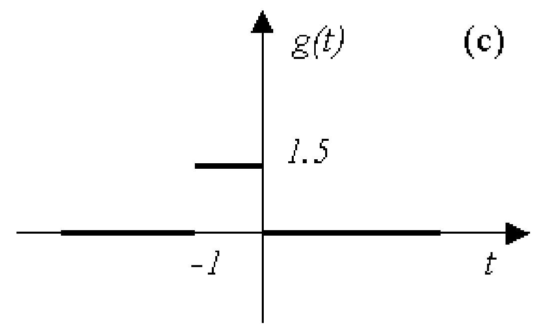


Convolution

- Example:

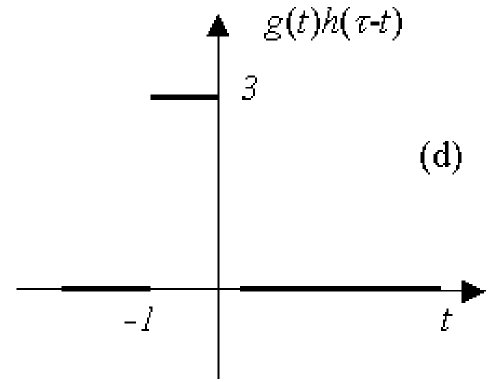






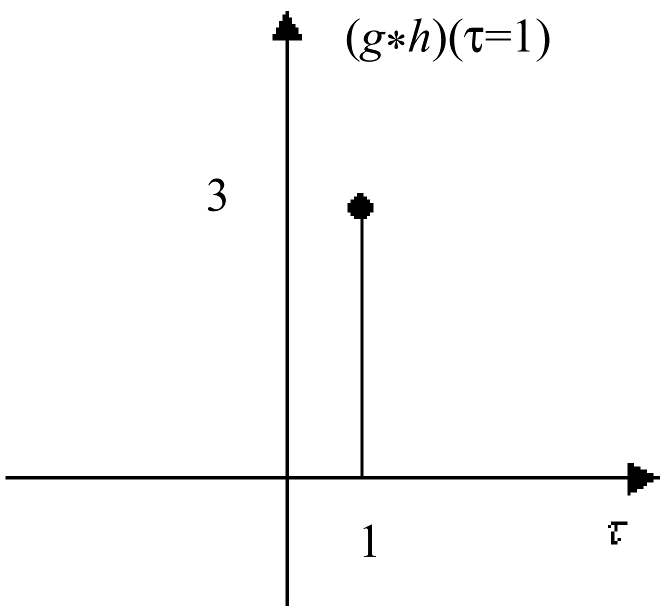
$$(g*h)(\tau)=\int_{-\infty}^{\infty} g(t)\,h(\tau-t)\,dt=3$$

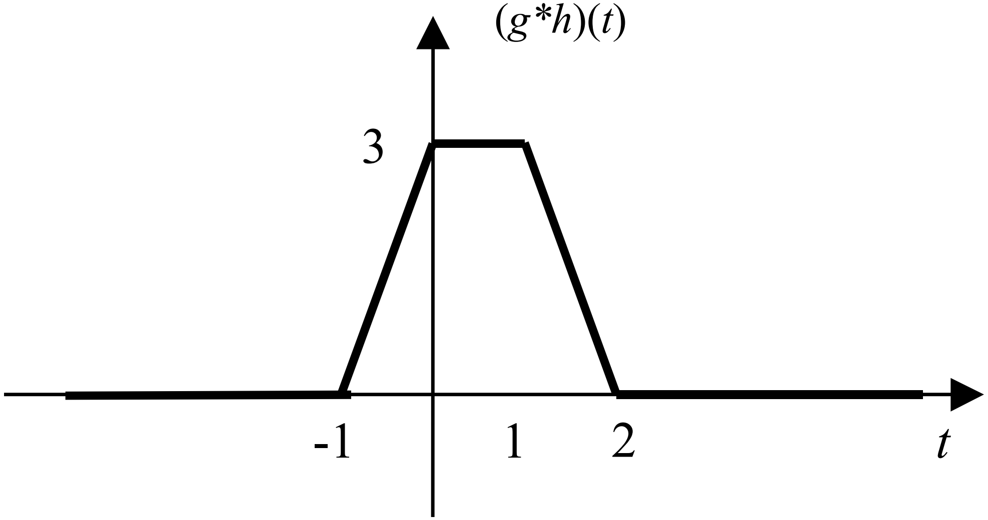
$$\tau \;=1$$



Convolution

- Example:



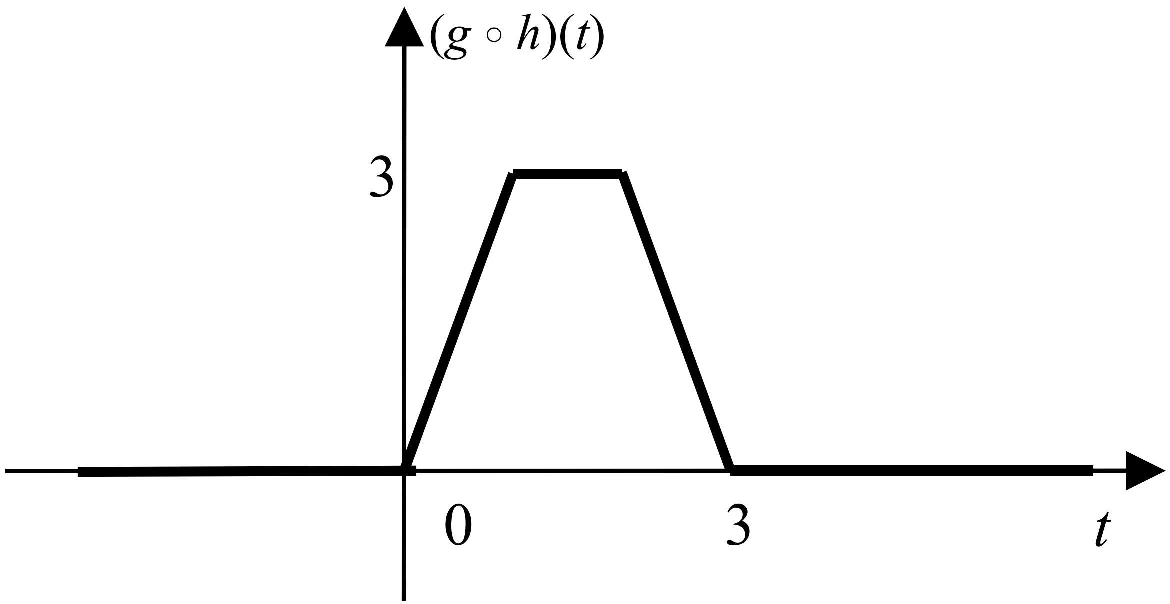


Correlation

- Let $g(t)$ and $h(t)$ be two real or complex functions. The *correlation* is defined as:

$$q(\tau) = g(\tau) \circ h(\tau) = (g \circ h)(\tau) = \int_{-\infty}^{\infty} g^{¿}(t)\, h(\tau + t)\, dt$$

Correlation



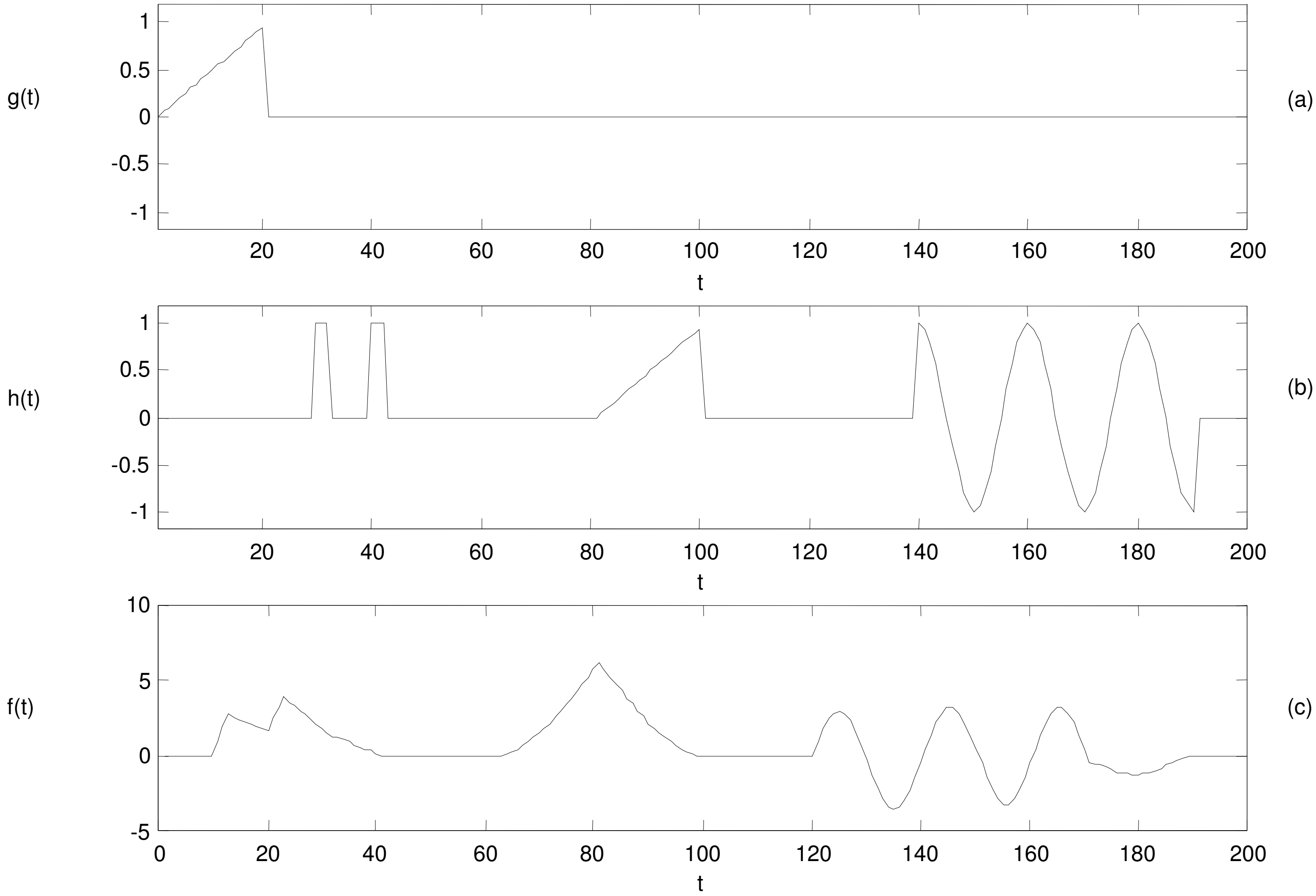
Correlation

- Let $g(t)$ and $h(t)$ be two real or complex functions defined on *[a,b]*. The *inner product* is defined as:

$$\langle g, h \rangle = \int_{a}^{b} g^{*}(t)\, h(t)\, dt$$

Similarity between vectors: pattern matching

Correlation



2D cases

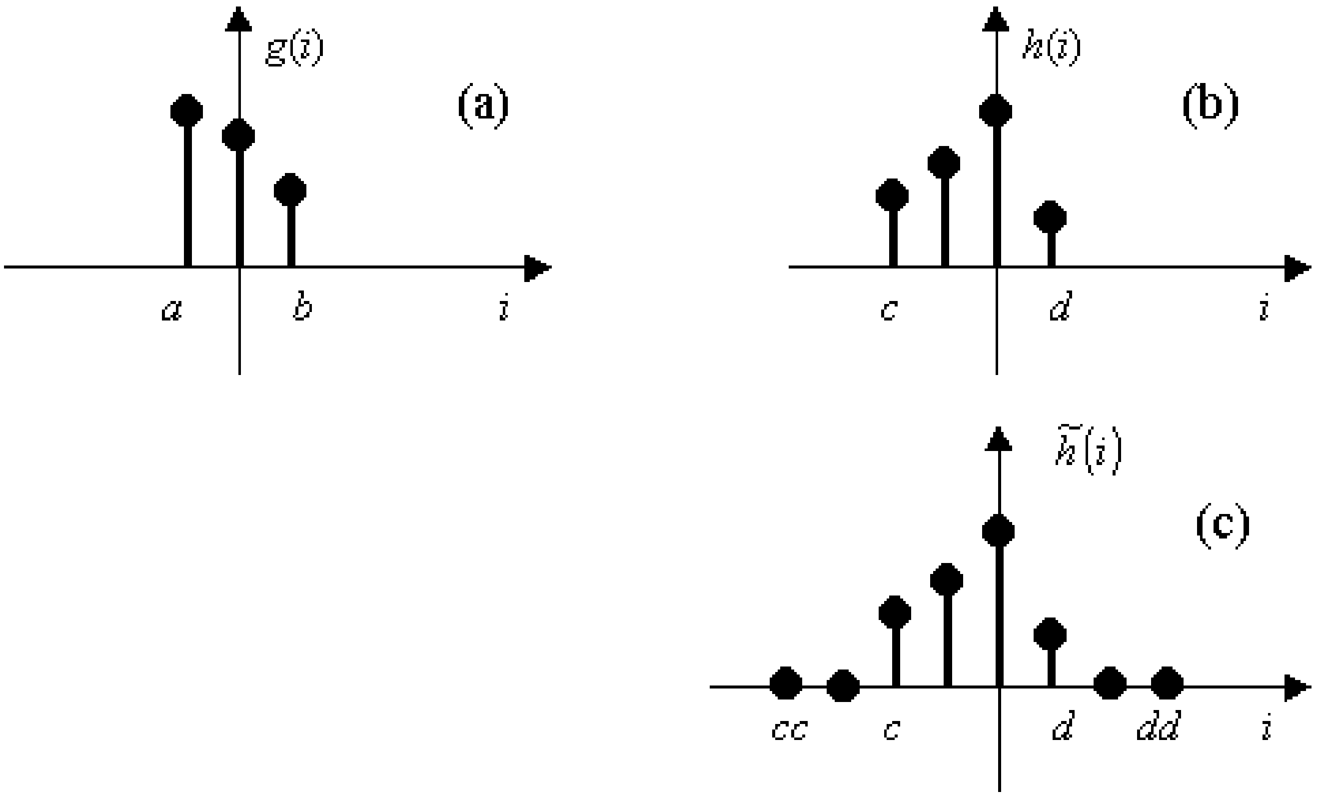
- Convolution

$$(g*h)(\alpha,\beta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\,h(\alpha-x,\beta-y)\,dxdy$$

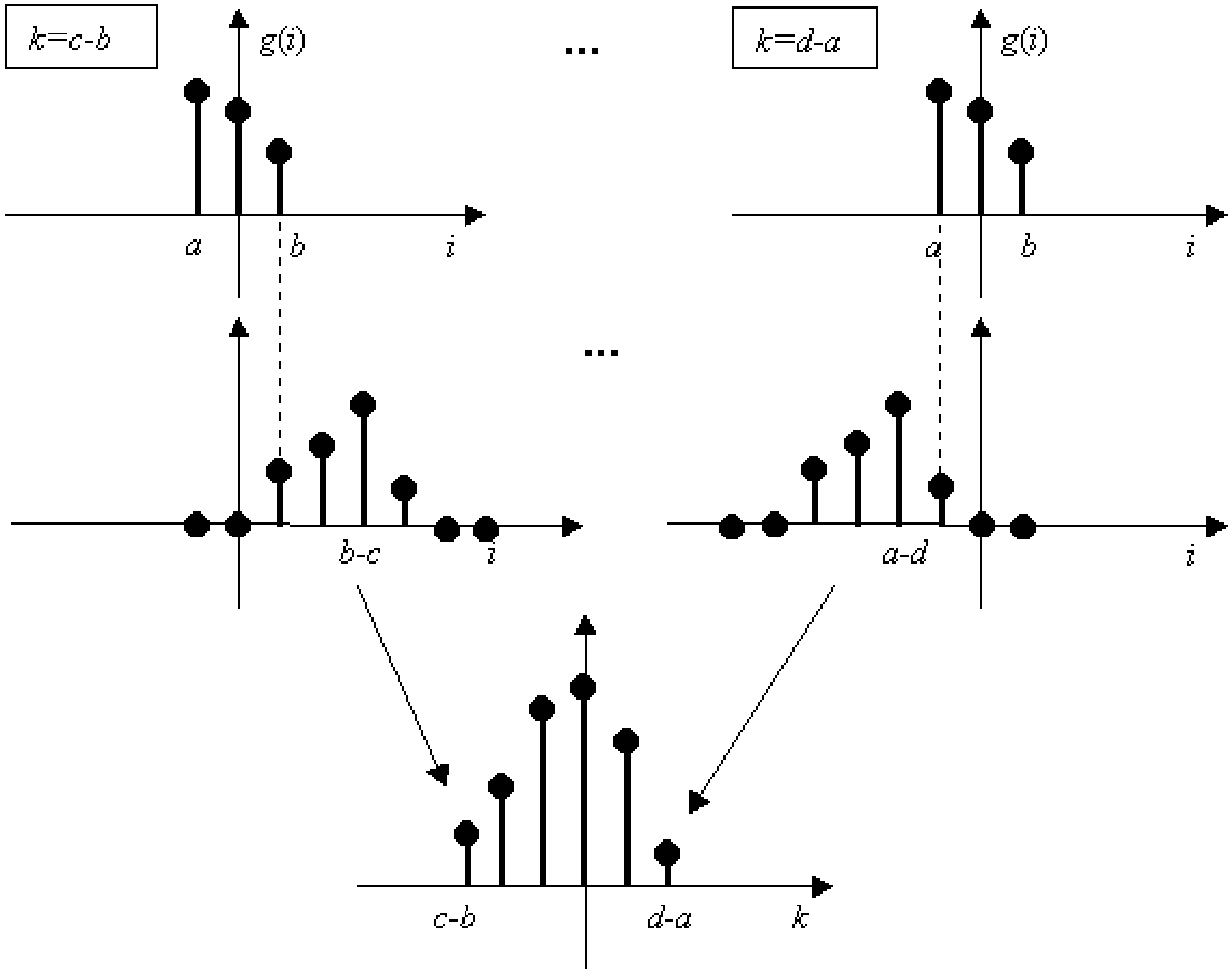
- Correlation

$$(g\circ h)(\alpha,\beta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g^{\text{¿}}(x,y)\,h(x+\alpha,y+\beta)\,dxdy$$

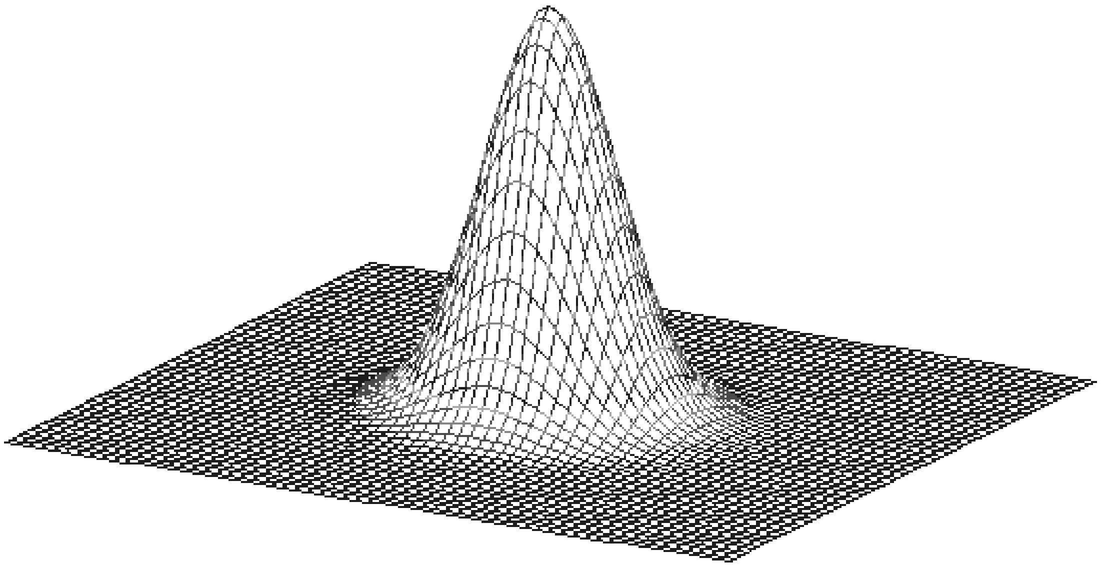
Discrete cases: example



Discrete cases: example



Gaussian smoothing



$$g(p,q) = \exp\left(-\frac{\left(p^2+q^2\right)}{2a^2}\right)$$

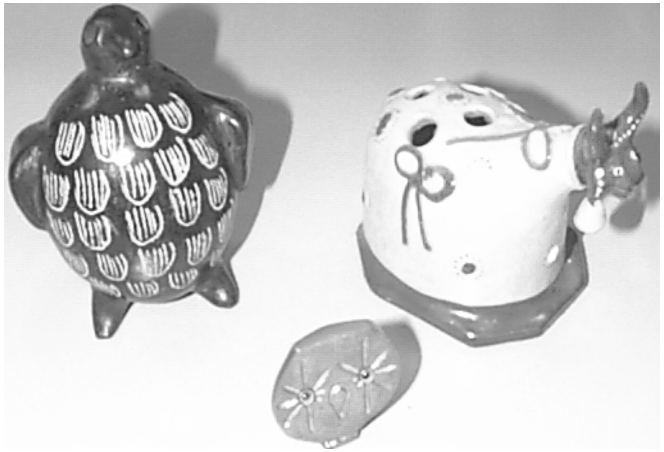
Gaussian smoothing

Gaussian smoothing

Gaussian smoothing

Gaussian smoothing

Gaussian smoothing



(a)

(b)

(c)

(d)

Fourier analysis

- The Fourier series of a periodic function $g(t)$, with period $2L$:

$$g(t) = a_0 + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right) \right]$$

$$a_0 = \frac{1}{2\mathrm{L}} \int_{-L}^{L} g(t)\, dt$$

$$a_n = \frac{1}{L} \int_{-L}^{L} g(t) \cos\left(\frac{n\pi t}{L}\right) dt$$

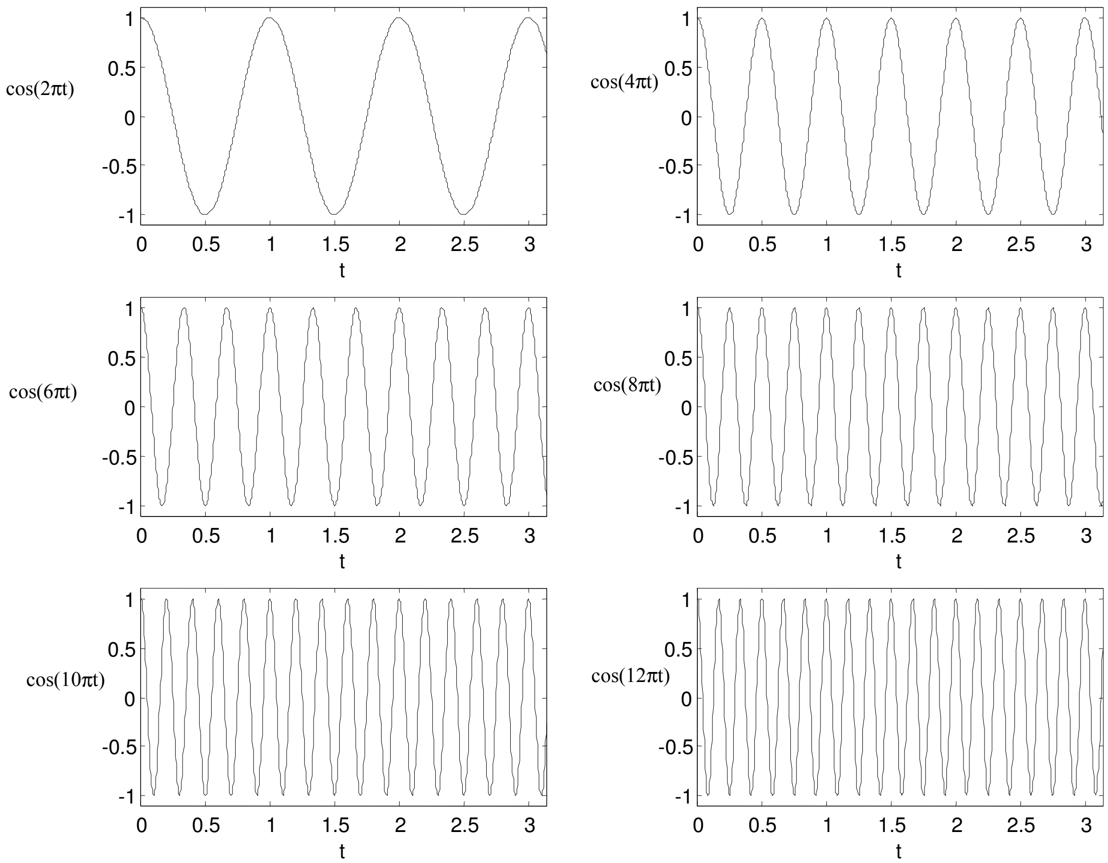
$$b_n = \frac{1}{L} \int_{-L}^{L} g(t) \sin\left(\frac{n\pi t}{L}\right) dt$$

Fourier analysis

- $g(t)$ is represented as a weighted sum of sines and cosines (ie linear combination), with frequencies defined as:

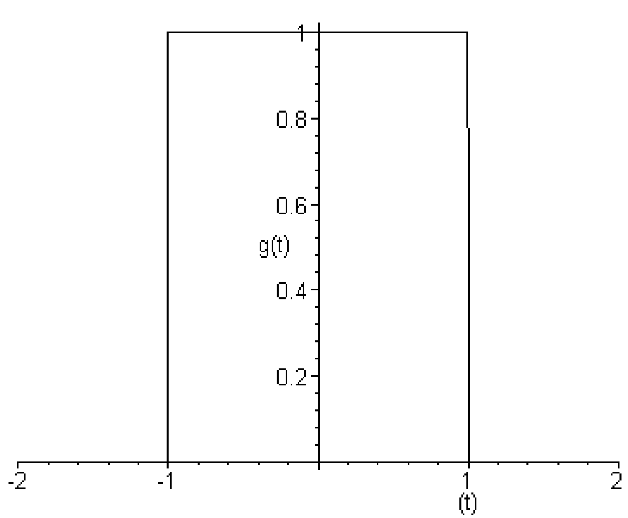
$$\frac{n\pi t}{L} = 2\pi\, ft \Leftrightarrow f = \frac{n}{2\mathrm{L}}$$

Fourier analysis

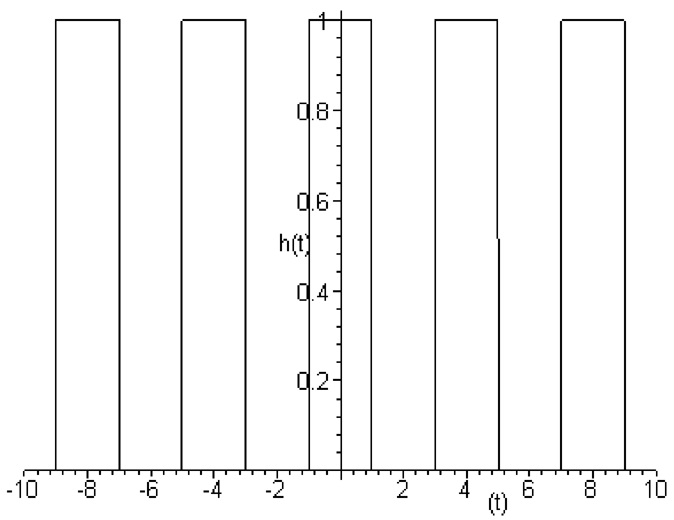


Fourier analysis

Example: $$g(t)=\begin{cases}1 & if\ -a\le t<a\\ 0 & otherwise\end{cases}$$



Original (non-periodic)



Periodic version

Fourier analysis

$$a_0 = \frac{1}{2L} \int_{-L}^{L} h(t)\,dt = \frac{1}{4a} \int_{-2a}^{2a} h(t)\,dt = \frac{1}{4a} \int_{-a}^{a} 1\,dt = \frac{1}{2}$$

Fourier analysis

$$a_n = \frac{1}{L}\int_{-L}^{L} h(t)\cos\left(\frac{n\pi t}{L}\right)dt = \frac{1}{2\mathrm{a}}\int_{-2\mathrm{a}}^{2\mathrm{a}} h(t)\cos\left(\frac{n\pi t}{2\mathrm{a}}\right)dt = \frac{1}{2\mathrm{a}}\int_{-a}^{a}\cos\left(\frac{n\pi t}{2\mathrm{a}}\right)dt =$$

$$\text{¿}\,\frac{1}{2\mathrm{a}}\left[\frac{2\mathrm{a}}{n\pi}\sin\left(\frac{n\pi t}{2\mathrm{a}}\right)\right]_{-a}^{a} \overset{a=1}{=} \frac{1}{n\pi}\left[\sin\left(\frac{n\pi}{2}\right) - \sin\left(-\frac{n\pi}{2}\right)\right] =$$

$$\text{¿}\,\frac{1}{n\pi}\left[\sin\left(\frac{n\pi}{2}\right) + \sin\left(\frac{n\pi}{2}\right)\right] = \frac{2}{n\pi}\sin\left(\frac{n\pi}{2}\right) = \mathbf{sinc}\left(\frac{n}{2}\right)$$

Fourier analysis

$$b_n = \frac{1}{L}\int_{-L}^{L} h(t)\sin\left(\frac{n\pi t}{L}\right)dt = \frac{1}{2\mathrm{a}}\int_{-2\mathrm{a}}^{2\mathrm{a}} h(t)\sin\left(\frac{n\pi t}{2\mathrm{a}}\right)dt = \frac{1}{2\mathrm{a}}\int_{-a}^{a} \sin\left(\frac{n\pi t}{2\mathrm{a}}\right)dt =$$

$$¿ -\frac{1}{2\mathrm{a}}\left[\frac{2\mathrm{a}}{n\pi}\cos\left(\frac{n\pi t}{2\mathrm{a}}\right)\right]_{-a}^{a} \overset{a=1}{=} -\frac{1}{n\pi}\left[\cos\left(\frac{n\pi}{2}\right) - \cos\left(-\frac{n\pi}{2}\right)\right] =$$

$$¿ -\frac{1}{n\pi}\left[\cos\left(\frac{n\pi}{2}\right) - \cos\left(\frac{n\pi}{2}\right)\right] = 0$$

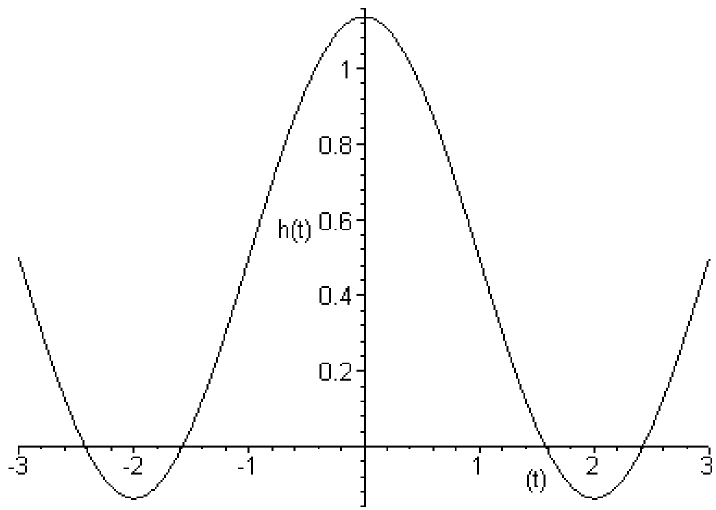
Fourier analysis

$$g(t)=\begin{cases}1 & if \ \ -a \leq t < a \\ 0 & otherwise\end{cases}$$

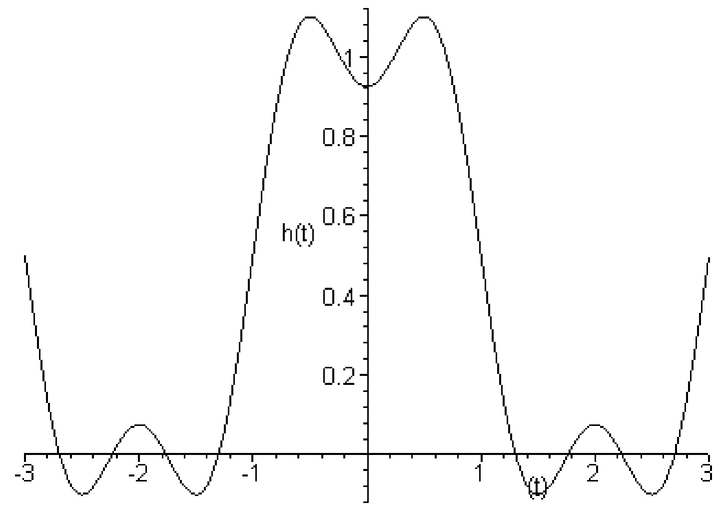
$$h(t)=\frac{1}{2}+\sum_{n=1}^{\infty}\left[\mathbf{sinc}\left(\frac{n}{2}\right)\cos\left(\frac{n\pi t}{2\mathrm{a}}\right)\right]$$

Fourier analysis

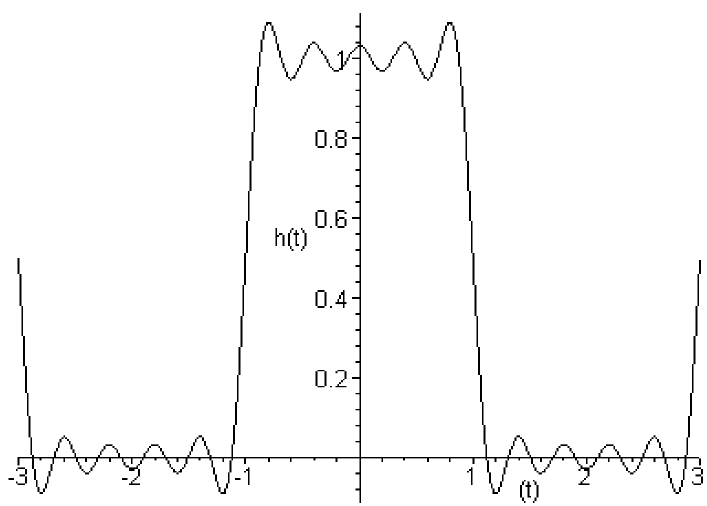
Considering a_0, b_1



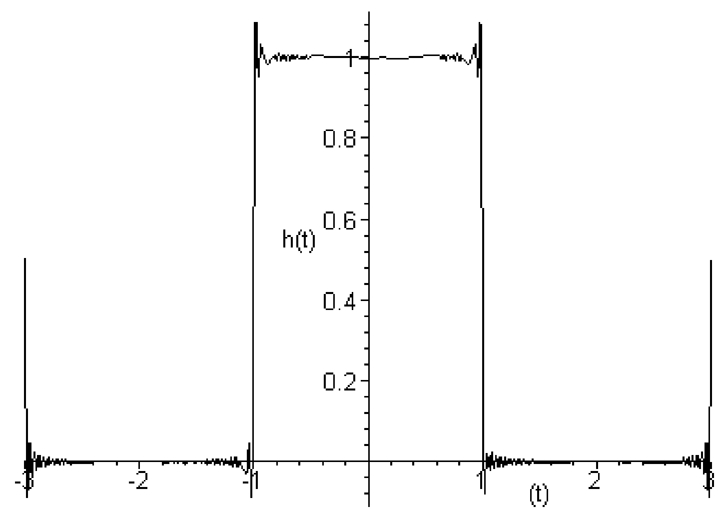
Considering a_0, b_1, b_2, b_3



Considering $a_0, b_1, b_2, \ldots, b_5$



Considering $a_0, b_1, b_2, \ldots, b_{100}$



Fourier analysis

Euler formula: $\exp\{j\theta\} = \cos(\theta) + j\sin(\theta)$

$$g(t) = \sum_{n=-\infty}^{\infty} \left[c_n \exp\left\{ \frac{jn\pi t}{L} \right\} \right]$$

$$c_n = \frac{1}{2\mathrm{L}} \int_{-L}^{L} g(t) \exp\left\{ -\frac{jn\pi t}{L} \right\} dt$$

Fourier analysis

Continuous Fourier transform:

$$G(f) = \Im\{g(t)\} = \int_{-\infty}^{\infty} g(t) \exp\{-j2\pi ft\}\, dt$$

Inverse Fourier transform:

$$g(t) = \Im^{-1}\{G(f)\} = \int_{-\infty}^{\infty} G(f) \exp\{j2\pi ft\}\, df$$

Fourier analysis

2D Continuous Fourier transform:

$$G(u,v) = \Im\{g(x,y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\exp\{-j2\pi(ux+vy)\}\,dxdy$$

2D Inverse Fourier transform:

$$g(x,y) = \Im^{-1}\{G(u,v)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(u,v)\exp\{j2\pi(ux+vy)\}\,dudv$$

Fourier analysis

Fourier pair

$$g(x,y) \leftrightarrow G(u,v)$$

Fourier analysis

2D Fourier transform properties

Property	Description
Separability (DFT)	The discrete Fourier transform can be computed in terms of 1D Fourier transforms of the image rows followed by 1D transforms of the columns (or vice-versa).
Spatial Translation (Shifting)	$g(x-x_0, y-y_0) \leftrightarrow \exp[-j2\pi(ux_0+vy_0)]G(u,v)$
Frequency Translation (Shifting)	$\exp[j2\pi(xu_0+yv_0)]g(x,y) \leftrightarrow G(u-u_0, v-v_0)$
Conjugate Symmetry	If $g(x, y)$ is real, then $G(u,v) = G^*(-u,-v)$
Rotation by θ	$g(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta) \leftrightarrow$ $\leftrightarrow G(u\cos\theta + v\sin\theta, -u\sin\theta + v\cos\theta)$
Linearity – Sum	$g_1(x,y) + g_2(x,y) \leftrightarrow G_1(u,v) + G_2(u,v)$

Fourier analysis

Linearity – Multiplication by Scalars	$ag(x, y) \leftrightarrow aG(u, v)$
Scaling	$g(ax, by) \leftrightarrow \frac{1}{\lvert ab \rvert} G\left(\frac{u}{a}, \frac{v}{b}\right)$
Average Value	The image average value is directly proportional to $G(0,0)$ (the so called DC component).
Convolution Theorem	$g(x, y) * h(x, y) \leftrightarrow G(u, v)H(u, v)$ and $g(x, y)h(x, y) \leftrightarrow G(u, v) * H(u, v)$
Correlation Theorem	$g(x, y) \circ h(x, y) \leftrightarrow G^*(u, v)H(u, v)$ and $g^*(x, y)h(x, y) \leftrightarrow G(u, v) \circ H(u, v)$
Differentiation	$\left(\frac{\partial}{\partial x}\right)^m \left(\frac{\partial}{\partial y}\right)^n g(x, y) \leftrightarrow (j2\pi u)^m (j2\pi v)^n G(u, v)$

Fourier analysis

2D Discrete Fourier transform:

$$G_{r,s} = \Im\left\{ g_{p,q} \right\} = \frac{1}{MN} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} g_{p,q} \exp\left\{ -j2\pi \left(\frac{pr}{M} + \frac{qs}{N} \right) \right\}$$

2D Discrete Inverse Fourier transform:

$$g_{p,q} = \Im^{-1}\left\{ G_{r,s} \right\} = \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} G_{r,s} \exp\left\{ j2\pi \left(\frac{pr}{M} + \frac{qs}{N} \right) \right\}$$

Fourier analysis

2D Discrete Fourier transform:

Algorithm: ***Frequency Filtering***

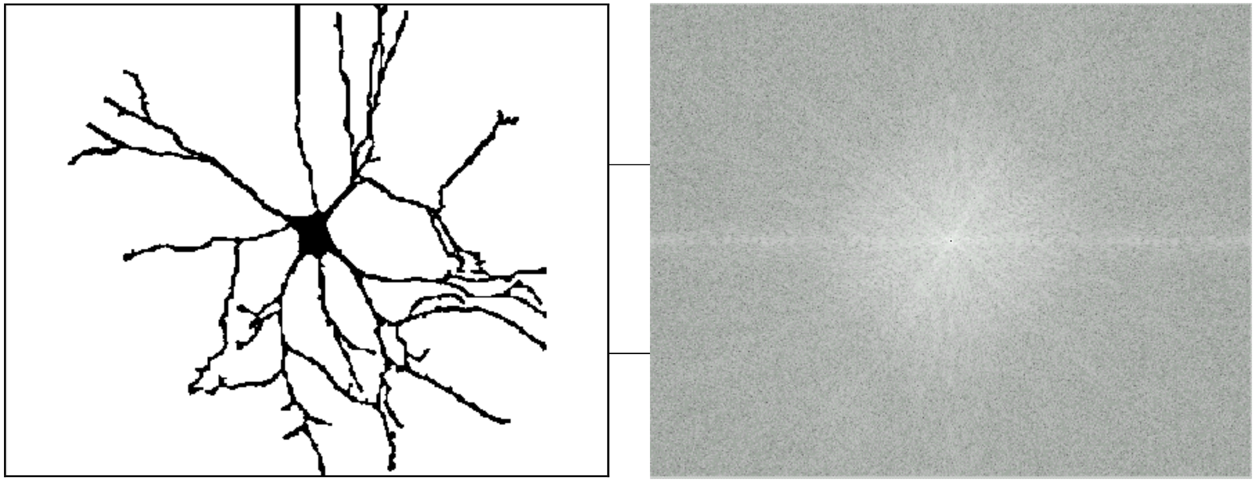
```
1.Choose G(r,s);
2.Calculate the Fourier transform F(r,s);
3.H(r,s) = F(r,s) G(r,s);
4.Calculate the inverse Fourier Transform h(p,q);
```

Fourier analysis

Example: Box filter

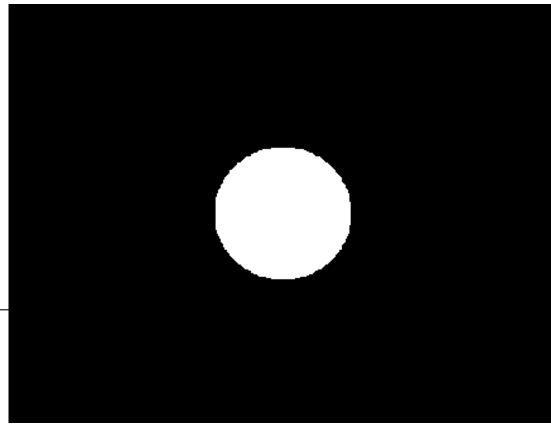
$$G_{r,s} = \begin{cases} 1, & \text{if } \left(r^2 + s^2\right) \leq T \\ 0, & \text{if } \left(r^2 + s^2\right) > T \end{cases}$$

Fourier analysis

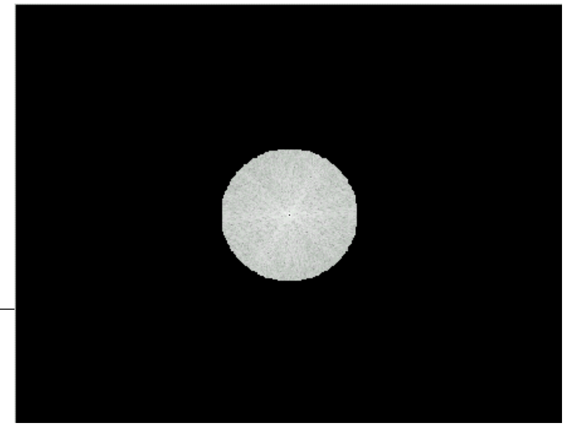


(a) (b)

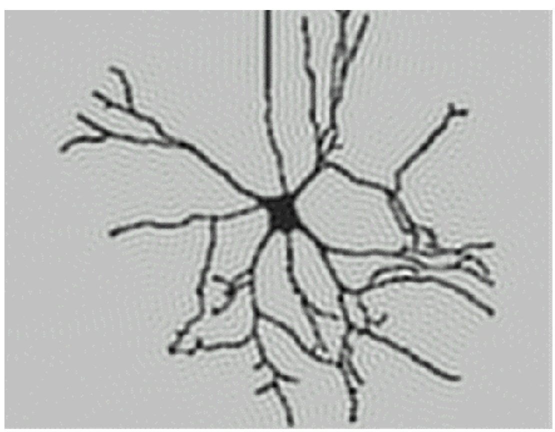
Fourier analysis



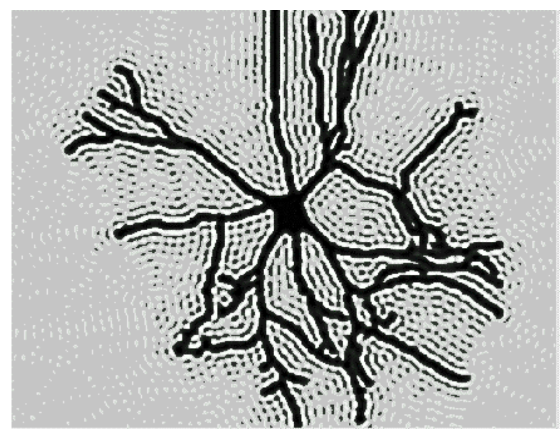
(a)



(b)



(c)



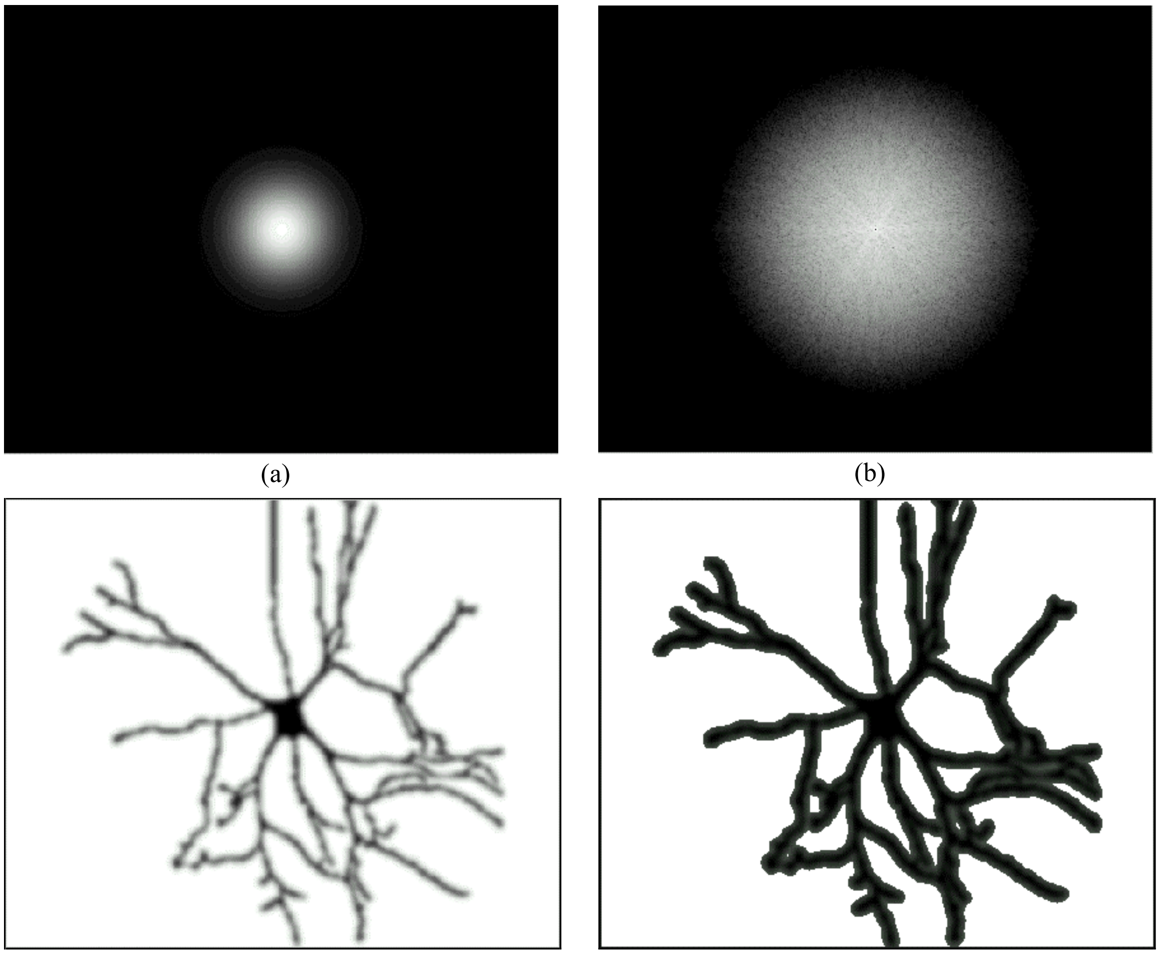
(d)

Fourier analysis

Example: Gaussian filter

$$G_{r,s} = \exp\left(-\frac{\left(r^2 + s^2\right)}{2\sigma^2}\right)$$

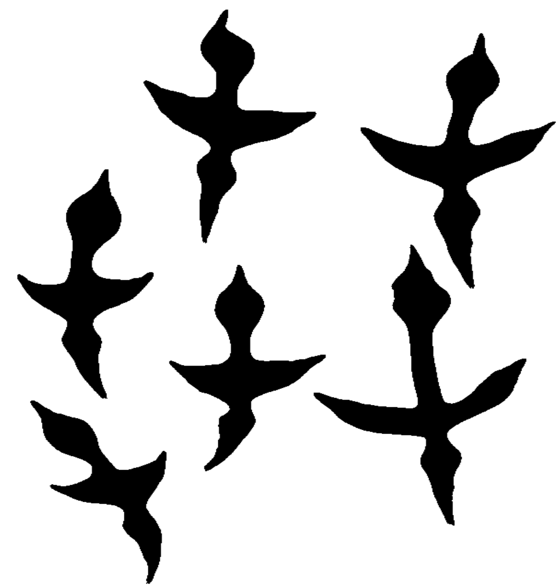
Fourier analysis



(a) (b) (c) (d)

Image Segmentation

(a)

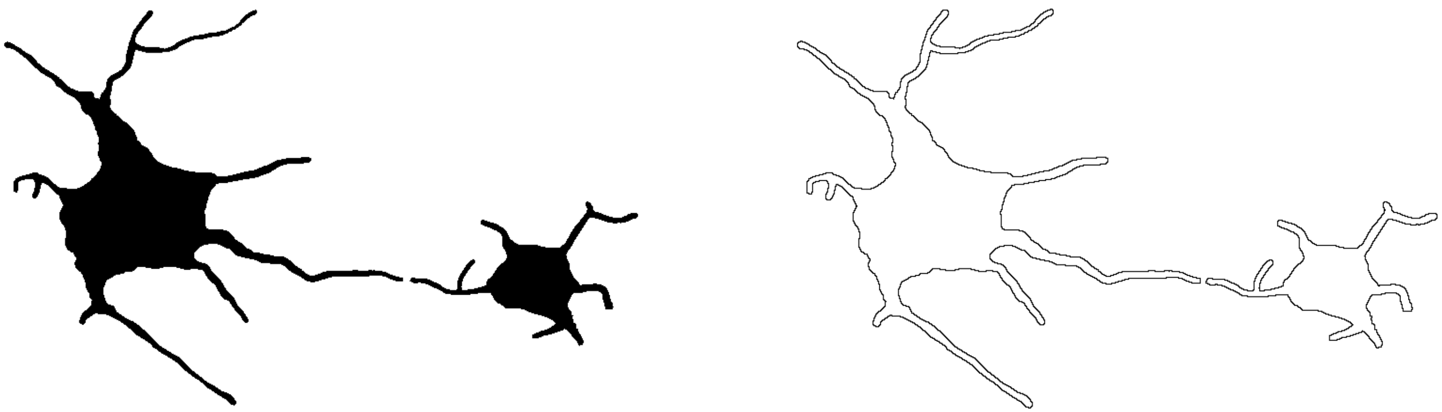


(b)

Image Segmentation

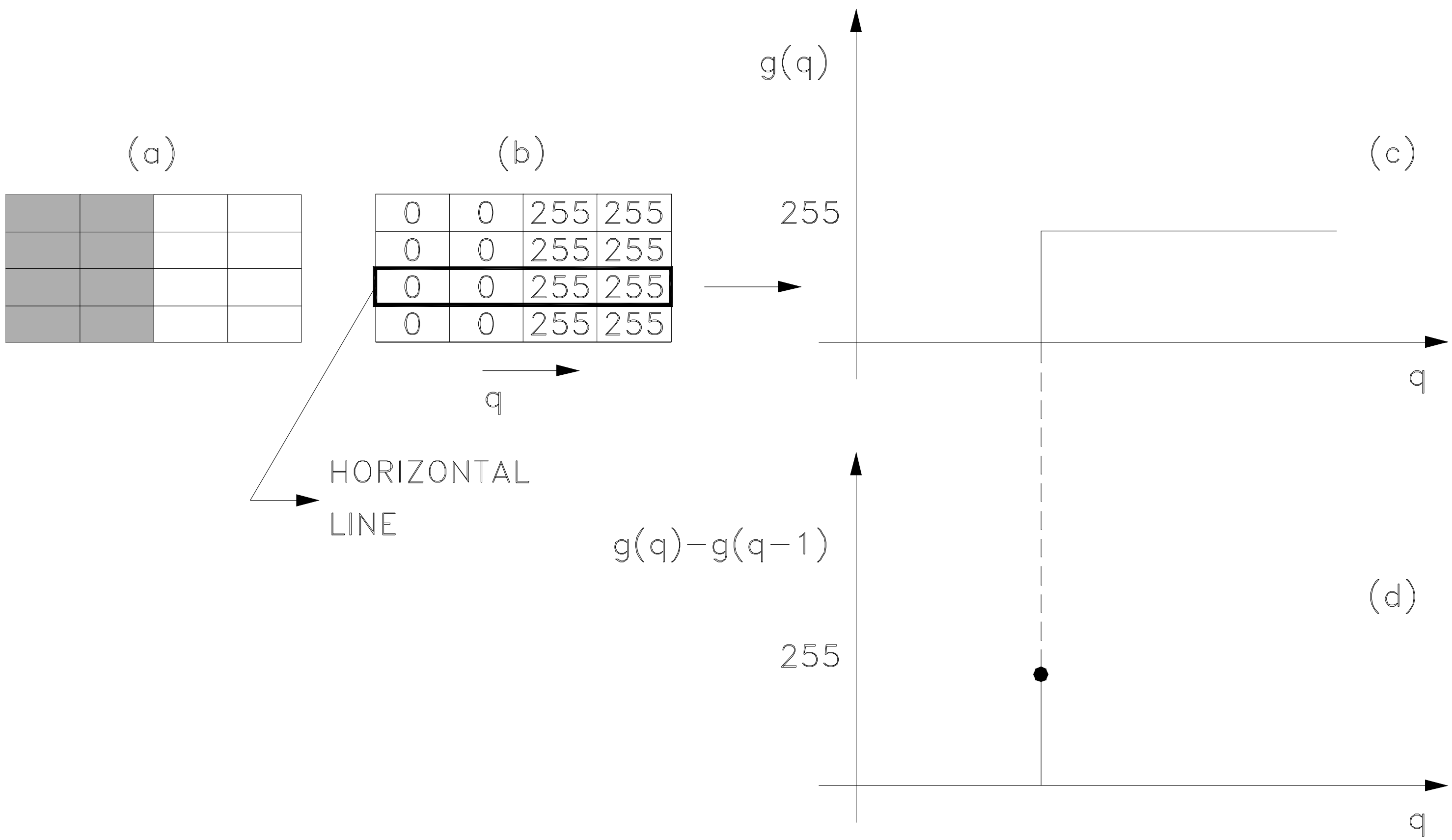
- Image thresholding
- Region Growing
- Optimization methods
- Snakes and active models
- …

Edge Detection



(a) (b)

Edge Detection



Edge Detection

Solution: numerical differentiation!

$$\vec{\nabla} g(x_0, y_0) = \left(\frac{\partial g}{\partial x}(x_0, y_0), \frac{\partial g}{\partial y}(x_0, y_0) \right)$$

$$|\vec{\nabla} g| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$$

Edge Detection

Example: finite differences

Algorithm: ***Simple Edge Detection***

```
hx = [-1 1]
hy = [-1 1]T
```

$$sk = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix};$$

```
sg = linfilt(g, sk);

gx = linfilt(g, hx);

gy = linfilt(g, hy);
```

There are many other different convolution-based edge detectors.

Edge Detection

The Sobel masks are defined as

$$\Delta x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\Delta y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Edge Detection

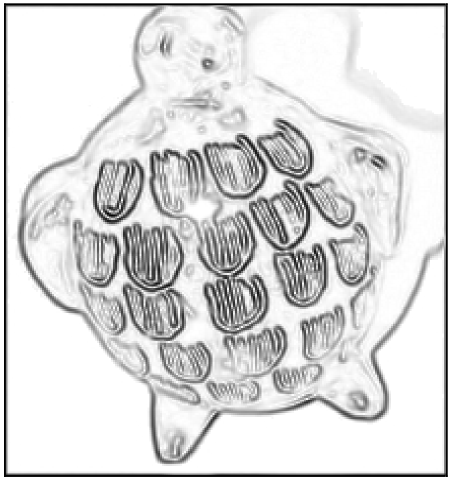
Fourier-Based Edge Detection

$$\left(\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial y}\right) g(x, y) \leftrightarrow (j2\pi u)(j2\pi v) G(u, v)$$

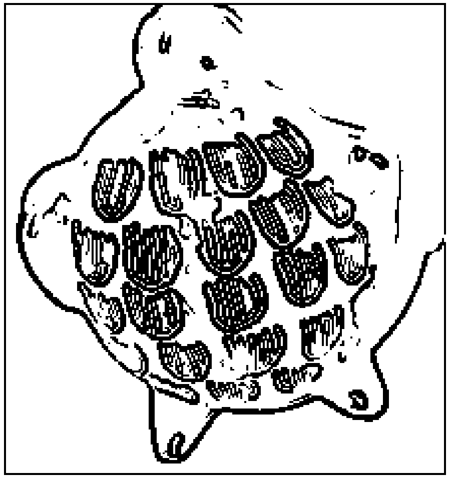
Edge Detection

Edge Detection

Fourier-based

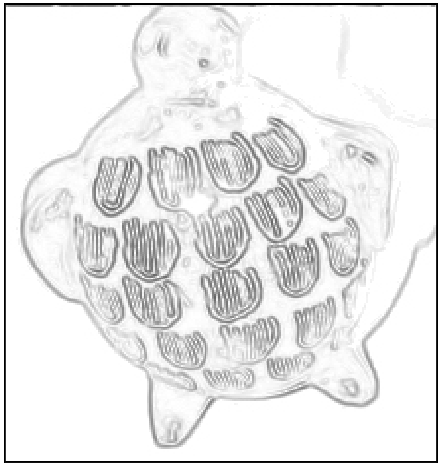


(b)

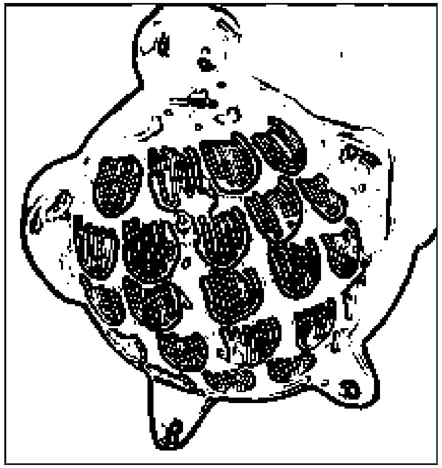


(c)

Sobel



(d)



(e)

Edge Detection

Second-order derivative: Laplacian

$$\nabla^2 g = \vec{\nabla} \cdot \vec{\nabla} g = \frac{\partial^2 g}{\partial x^2}(x, y) + \frac{\partial^2 g}{\partial y^2}(x, y)$$

Edge Detection

Marr-Hildreth transform: Laplacian of Gaussian - *LoG*

$$\frac{d}{dt}(g*u)=\left(\frac{d}{dt}g\right)*u=g*\frac{d}{dt}u$$

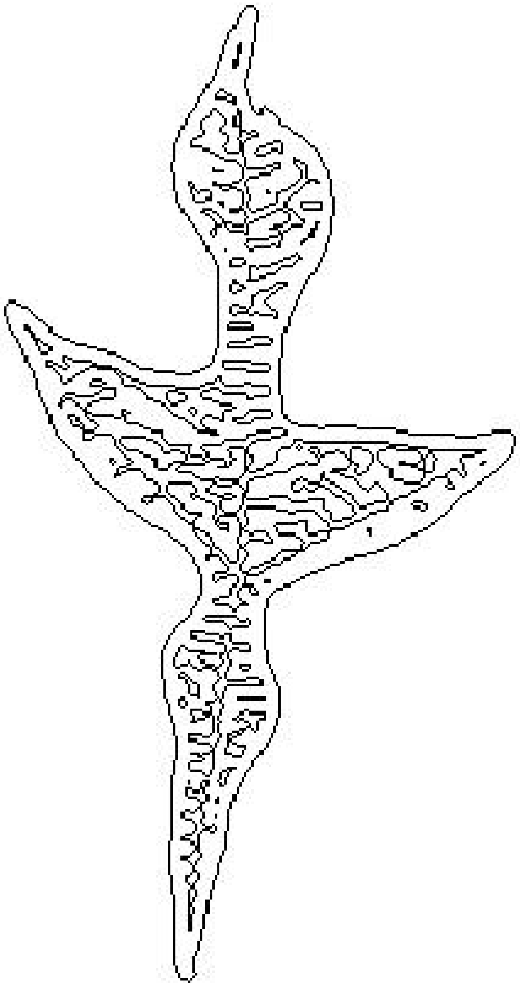
Edge Detection

Marr-Hildreth transform: Laplacian of Gaussian - *LoG*

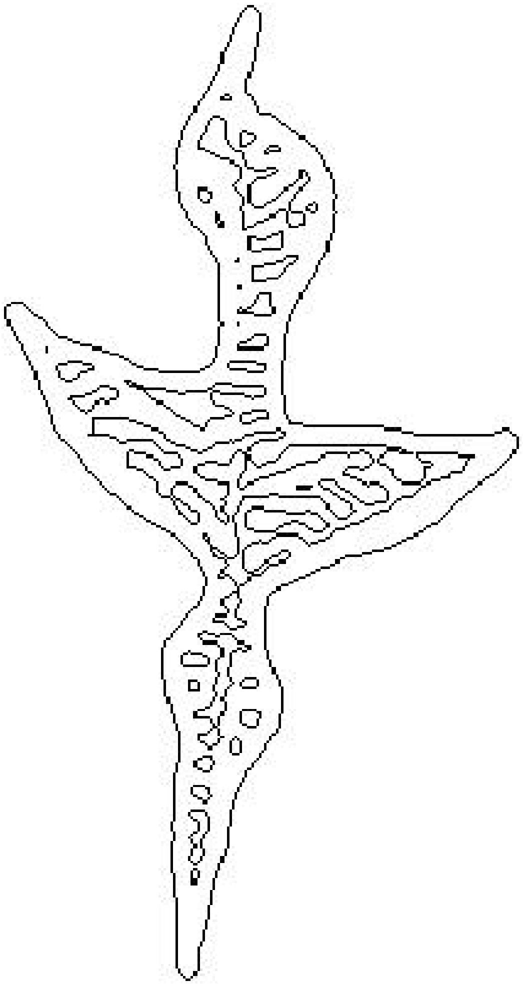
$$\nabla^2 g = \left(1 - \frac{x^2 + y^2}{\sigma^2}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Edge Detection

Edge Detection



Edge Detection



Edge Detection