Shape extraction

Edge detection

Segmentation

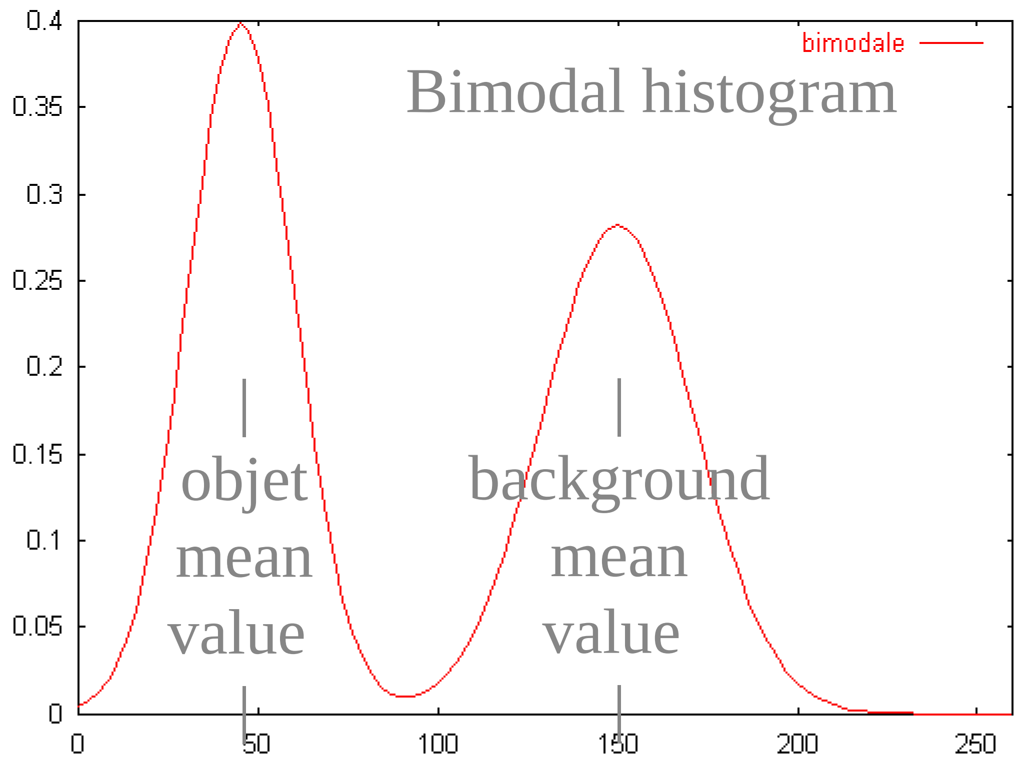
- Image segmentation consists into the decomposition of the image in segments (i.e. components)
- This process is based on a given criteria of homogeneity (chromatic, morphologic, motion, depth, etc.)
- From the operational viewpoint, three approach have been proposed:
  - Clustering image data and growing regions
  - Border following
  - Search of borders

Binary Images

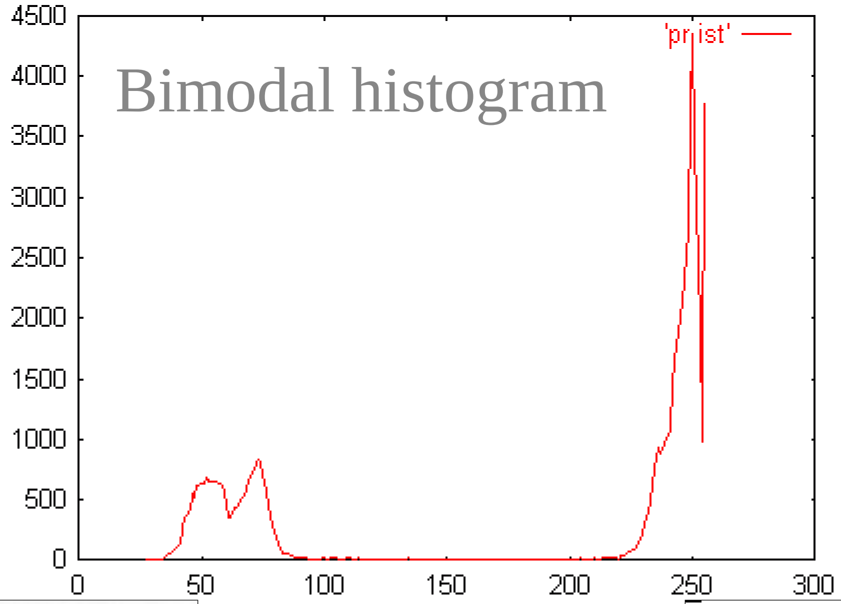
- The segmentation process leads to detect an individual object (foreground) in contrast to the background so it is a binarization process
- Some applications are by nature binary: black and white printing, writing, mechanical parts, bio-imagery like cells or chromosomes, etc. ....
- Often the originals contains various grey levels due to:
  - Electric noise of the camera
  - Non-uniform scene illuminations
  - Shadowing
  - …

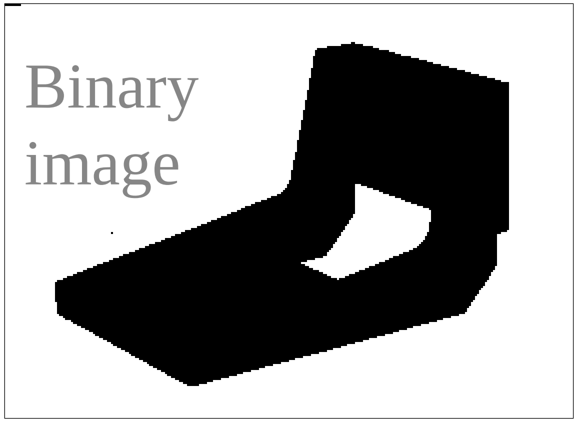
Bimodal Distribution

- The easest solution is a threshold applied to the grey levels:
  - O(i, j) = 0 se I(i, j) < S
  - O(i, j) = 255 otherwise
- It is required the evaluation of the optimal threshold S.
- Operating on the histogram, there are two possibilities:
  - Finding the minimum
  - Applying statistic criteria

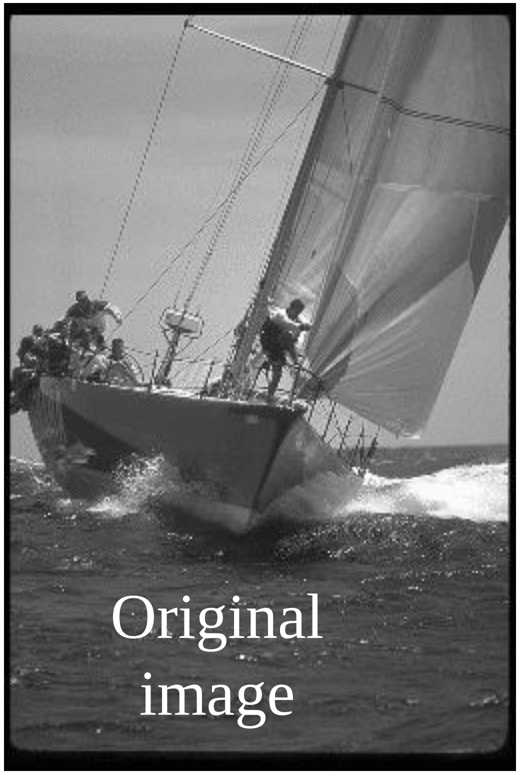


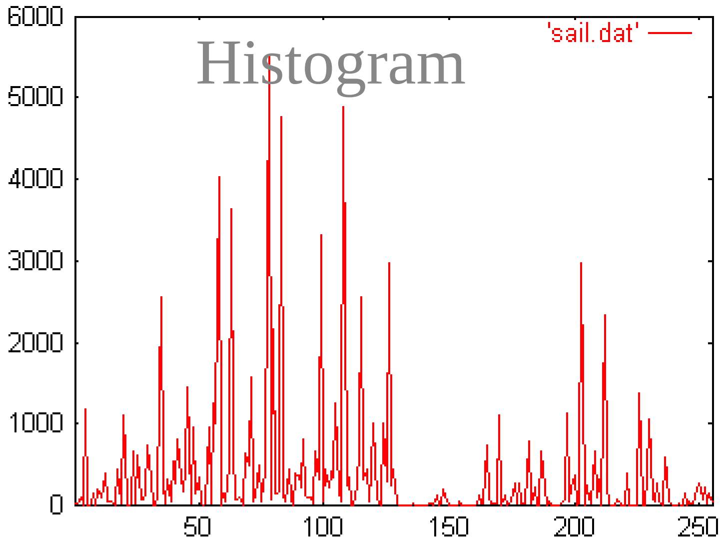
Example: mechanical part



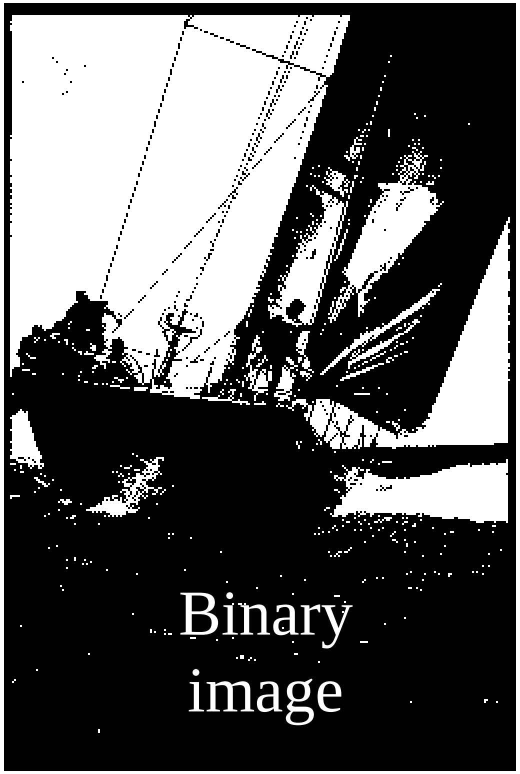


Example: sailing



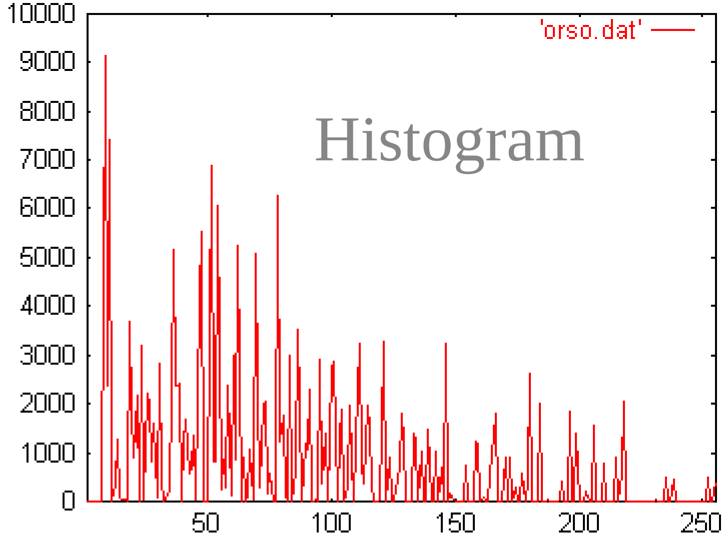


Threshold = 140

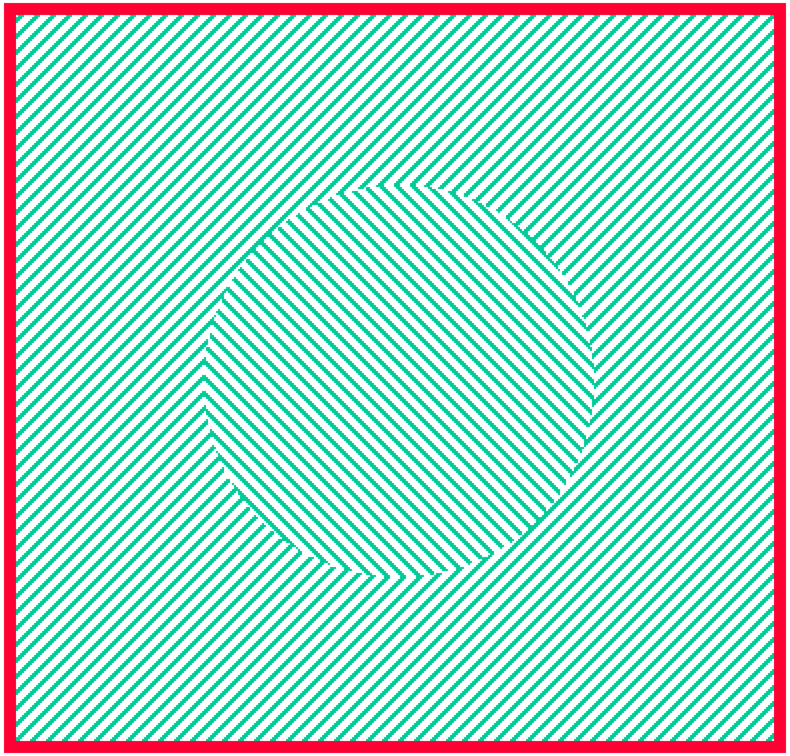


Example: bear

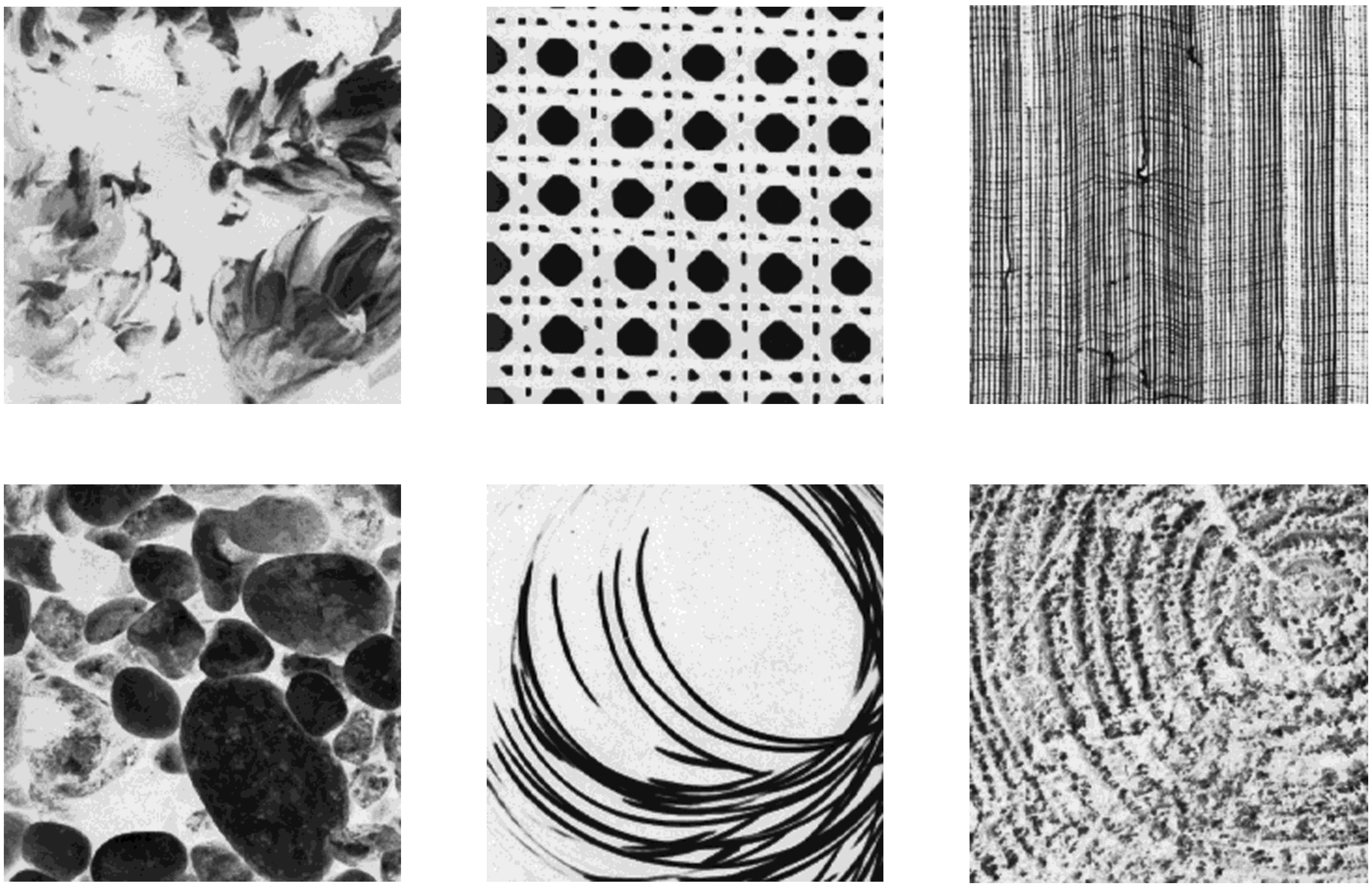
Original image



Example: circle

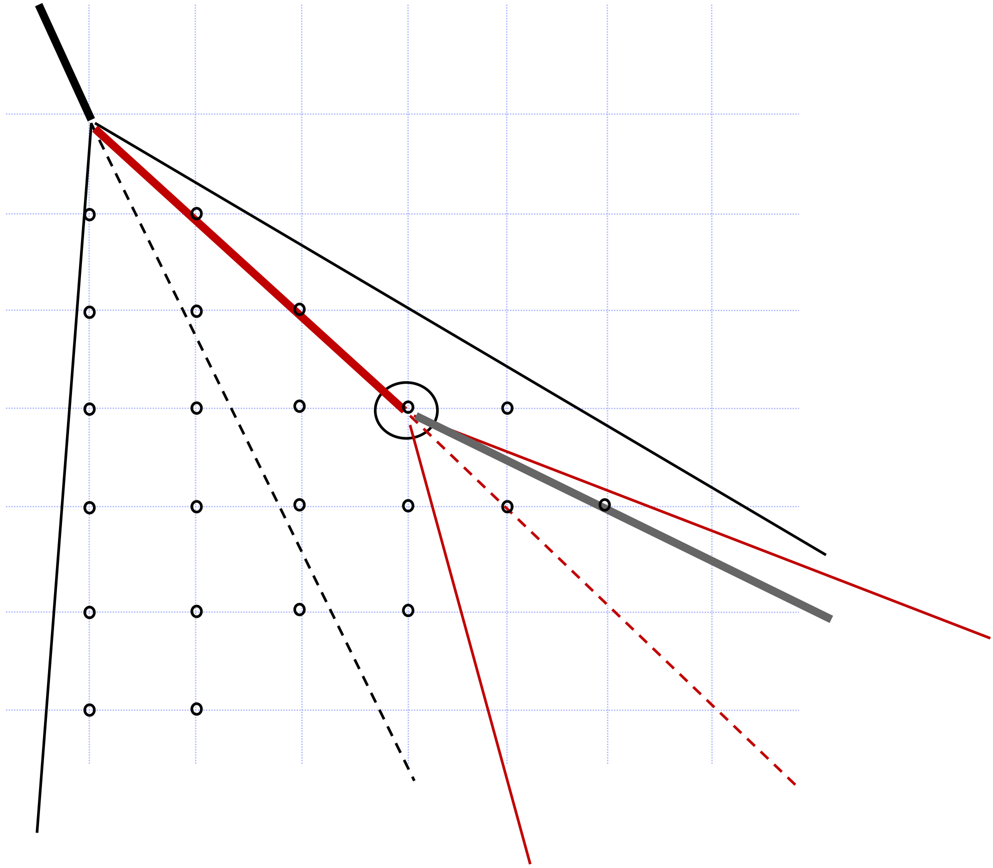


Texture: Brodatz album



Border following

- An example of a recursive walk over the image, following the contour to be exhibited. The horizon of an edge point is the triangle of depth 5 and basis 6, in the direction of the last found edge segment.



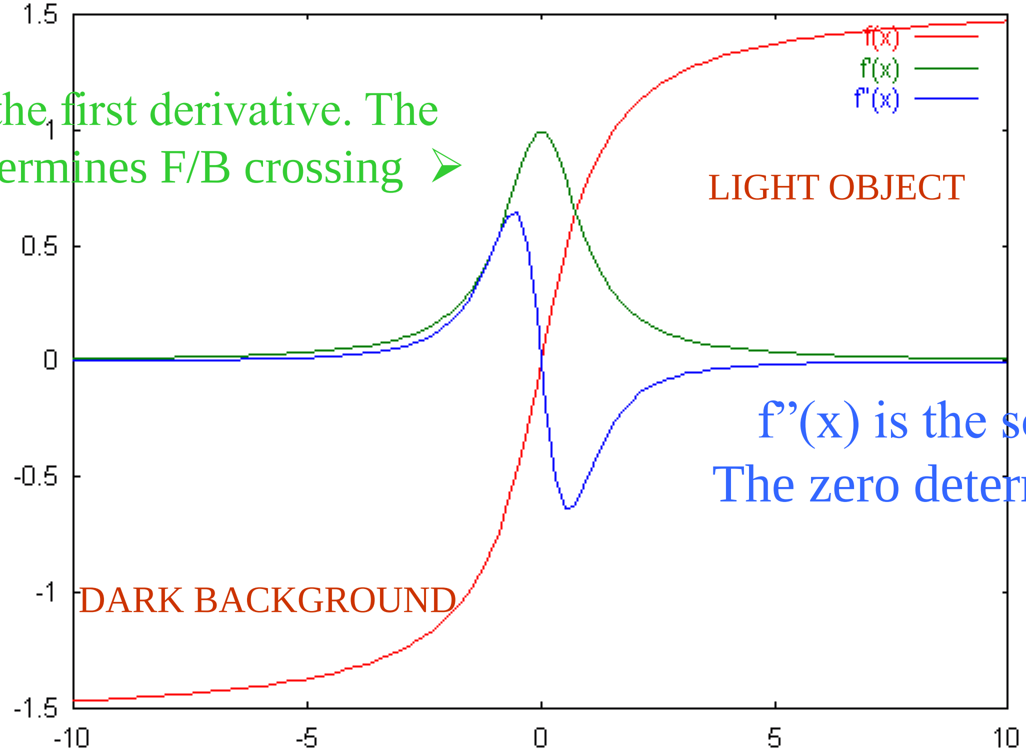
Search of borders

Analytic derivative model

- The border search can be based on the discontinuity of an image feature like the grey level, a texture or a motion parameter, the depth in the scene, etc.
- For operators stemming from first order partial derivatives a maximum response is looked for, either local maximum or over a threshold whether given or adapted
- Note that the second derivative is used too, and among second order operators the Laplacian is peculiarly popular as being scalar then isotropic. There, of course, the zero crossing – inflection points - are looked for

Analytic derivative model

f’(x) is the first derivative. The max determines F/B crossing ➢



f”(x) is the second derivative
The zero determines F/B crossing

f(x) is the grey level, here representing the image in one dimension

Analytic derivative model

- The first derivative is given by:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

- The second derivative is given by:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

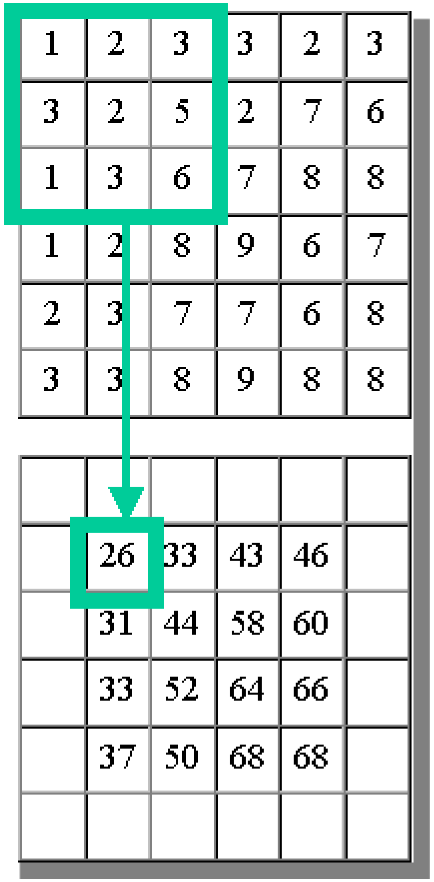
- In 2D the derivate is substituted by the vector gradient

Convolution

- The convolution is a linear operator, that is applied when the image I(x, y) is continue. To the digital image I(i, j) a filter is applied represented by the mask:

$$O(x_0,y_0) = \iint f(x_0-x, y_0-y)\, I(x,y)\, dx\, dy$$

$$O(x,y) = \sum\sum f(x-i, y-j)\, I(i,j)$$



Convolution (decomposition)

- In general the convolution is a computer demanding operator, e.g. the 5x5 template:

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

is implemented by 25 multiplications for each pixel; note that often complex template may be decomposed in simple 1D operators (e. g. the isotropic, monotonic decreasing template)

- The previous convolution can be decomposed in the following two 1D operators:

1 4 6 4 1 et

1
4
6
4
1

in this implementation only10 (5+5) multiplications per pixel are required

Gradient approximations

- The gradient is a 2D vector
- The digital differential operators are implemented by template in which the sum of the kernel parameters is null: in a uniform area the result must be zero (no variation)
- The basic and historical convolution kernels have an extension limited to 2x2 and 3x3, for each of the two components

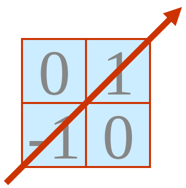
Roberts Operator

- It is the simplest solution
  - Two templates are applied M_1 and M_2, obtaining the two orthogonal gradient components:
  - $G_1=M_1*I$, $G_2=M_2*I$
  - It is very sensible to noise
- The gradient module and phase are:

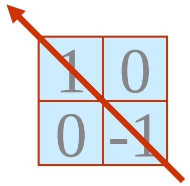
$$G_m = \sqrt{G_1^2 + G_2^2}$$

$$G_\phi = \mathrm{arctg}(G_2/G_1)+\pi/4$$

G_1



G_2



Isotropic operator

- Two templates are applied M_1 and M_2, obtaining the two orthogonal gradient components:

  $G_x = M_x * I,\ G_y = M_y * I$

- The gradient module and phase are:

  $G_m = \sqrt{G_1^2 + G_2^2}$

  $G_\phi = \text{arctg}(G_y / G_x)$

- In C:
  - phi = atan2(gy, gx)

G_x

-1	0	1
-√2	0	√2
-1	0	1

G_y

1	√2	1
0	0	0
-1	-√2	-1

Prewitt and Sobel operators

- To simplify the computation often the isotropic filter is implemented by these two simplified solutions:
  - Prewitt

G_x

-1	0	1
-1	0	1
-1	0	1

G_y

1	1	1
0	0	0
-1	-1	-1

  - Sobel

G_x

-1	0	1
-2	0	2
-1	0	1

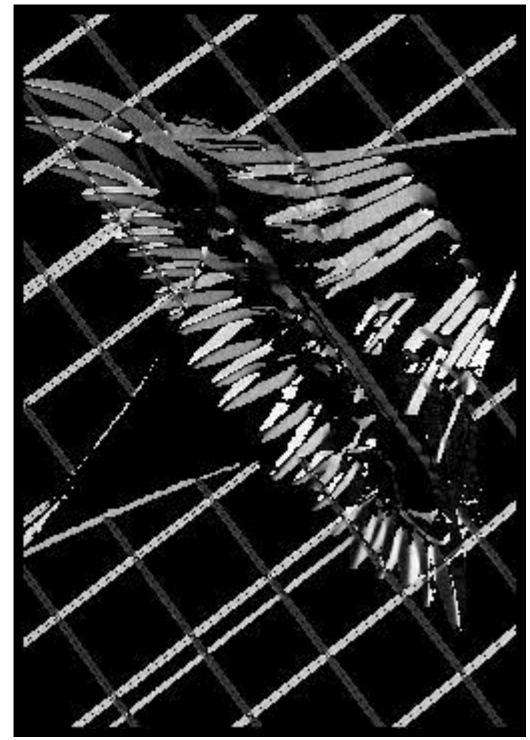
G_y

1	2	1
0	0	0
-1	-2	-1

Example Sobel

Original image

Module



Phase

Example Sobel

Original image

Module

Phase

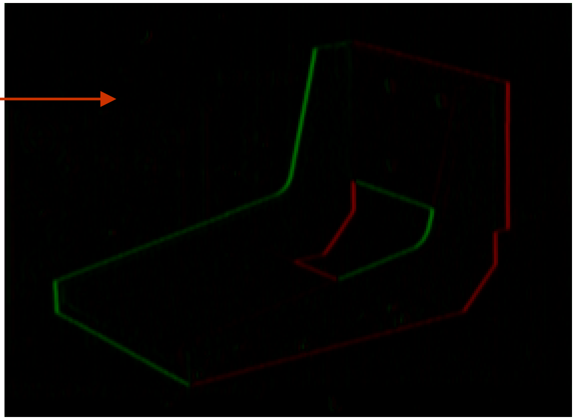
Sobel operator

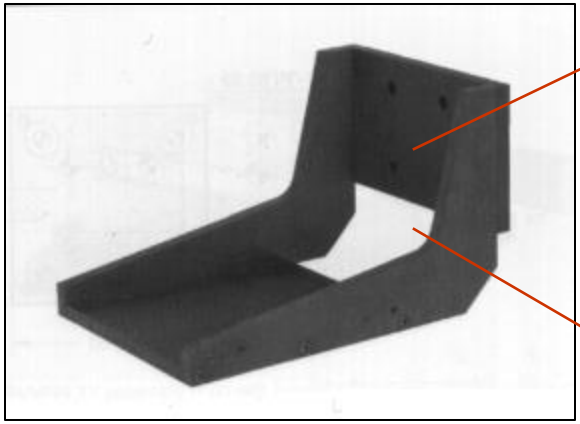
horizontal gradient

G_1

-1	0	1
-2	0	2
-1	0	1

vertical contour



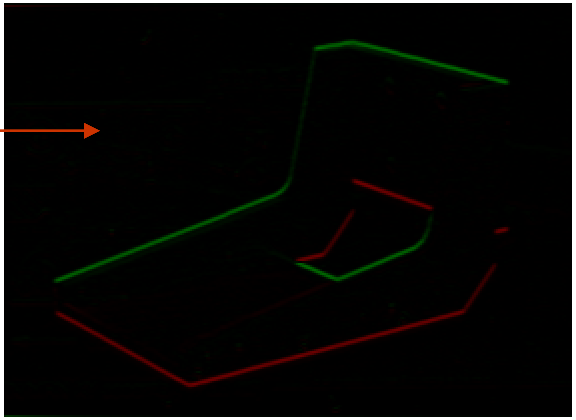


Original image

G_2

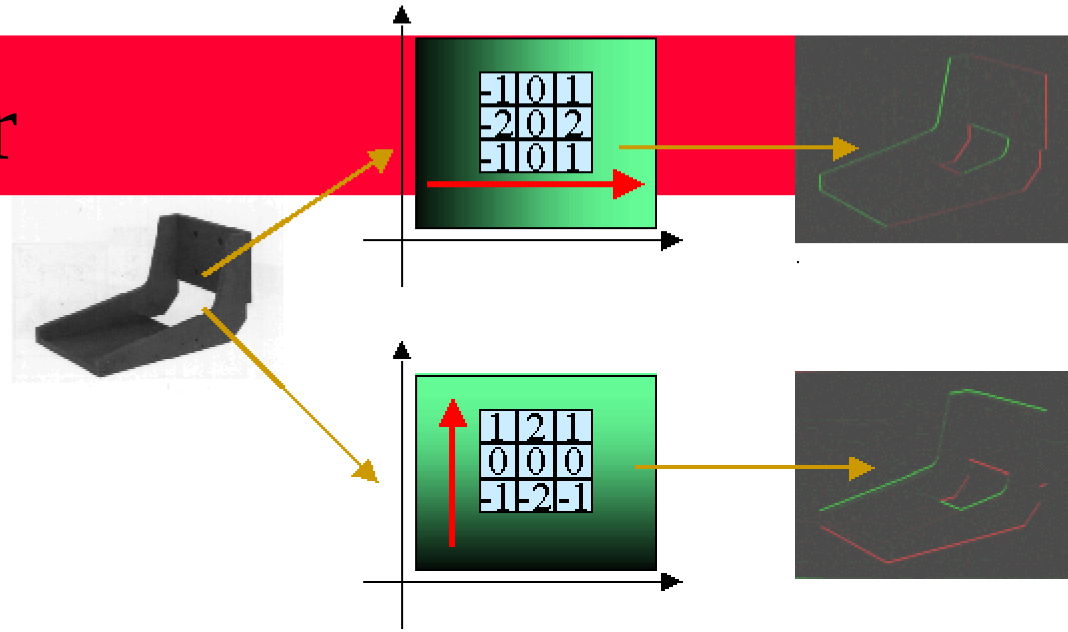
1	2	1
0	0	0
-1	-2	-1

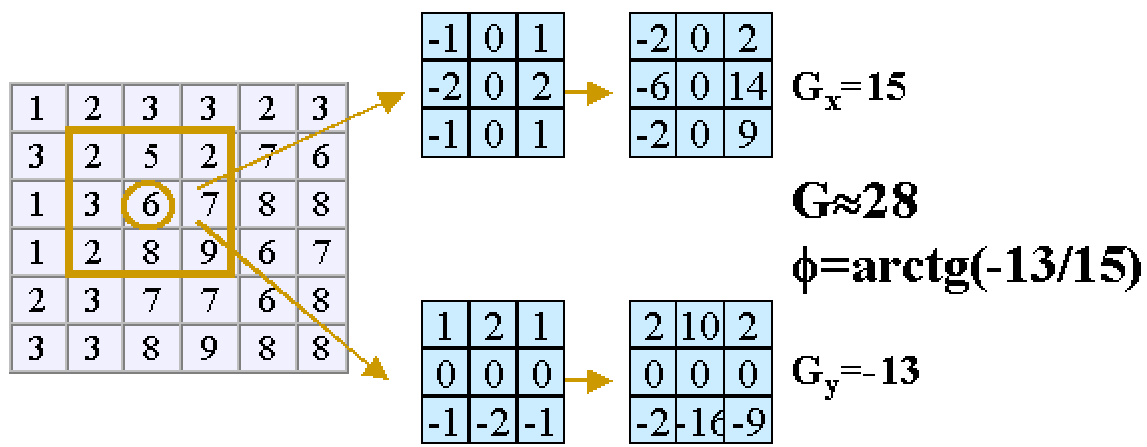
vertical gradient



horizontal contour

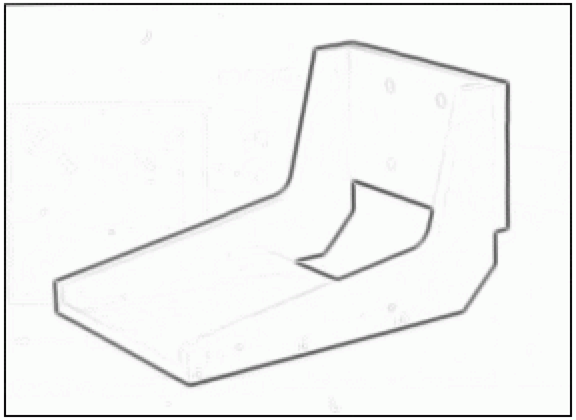
Sobel operator



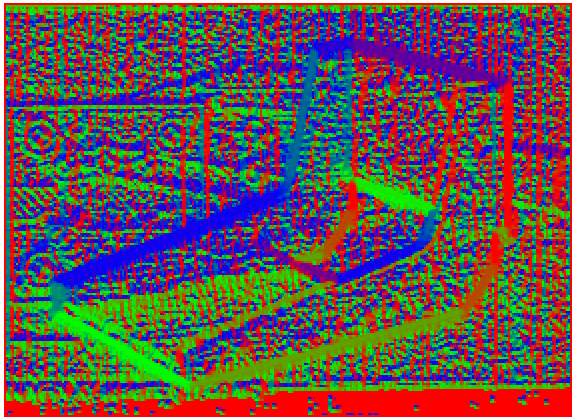


Sobel operator

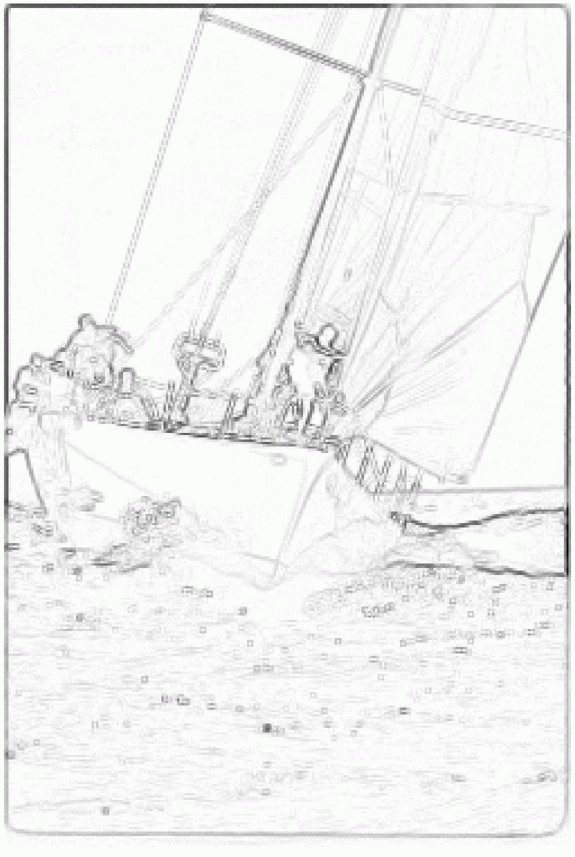
Module



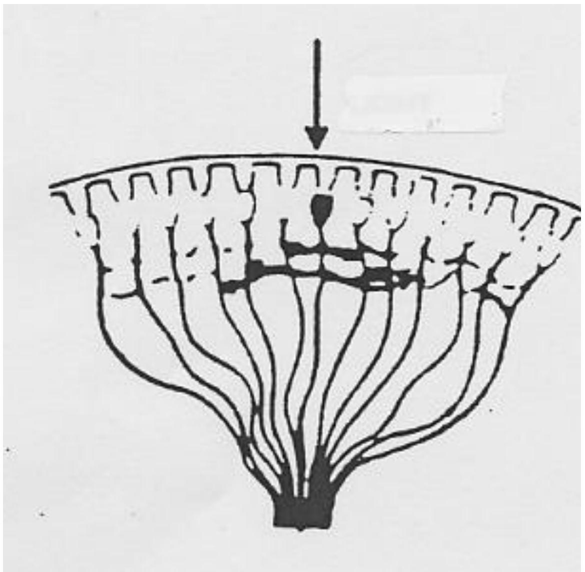
Phase

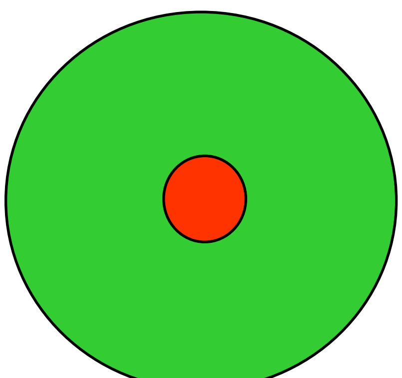


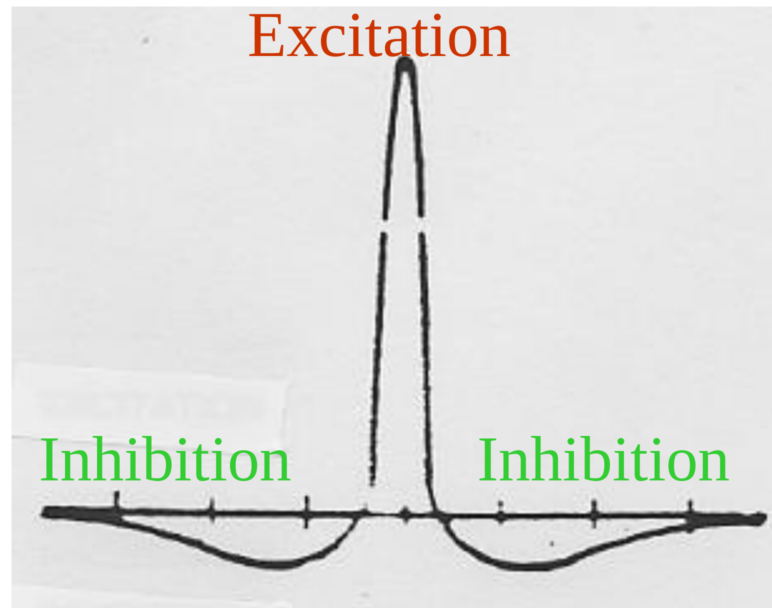
Example (module)



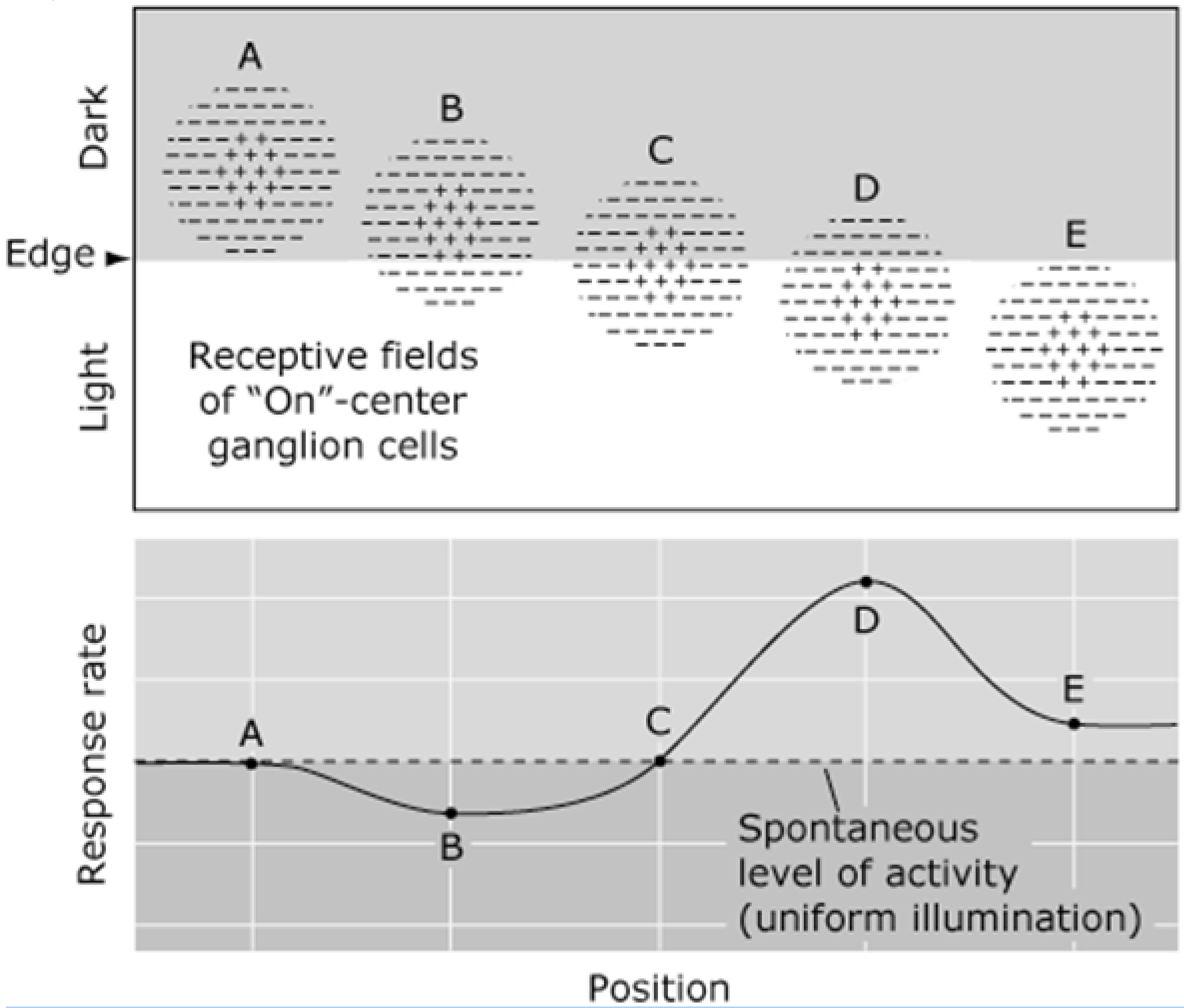
Lateral inhibition







Lateral inhibition



Lateral inhibition

- The retina receptor apply a lateral inhibition mechanism.
- The implementation of this mechanism can be done by a filter obtained by the difference of two Gaussian of equal area, having different σ (and amplitude):

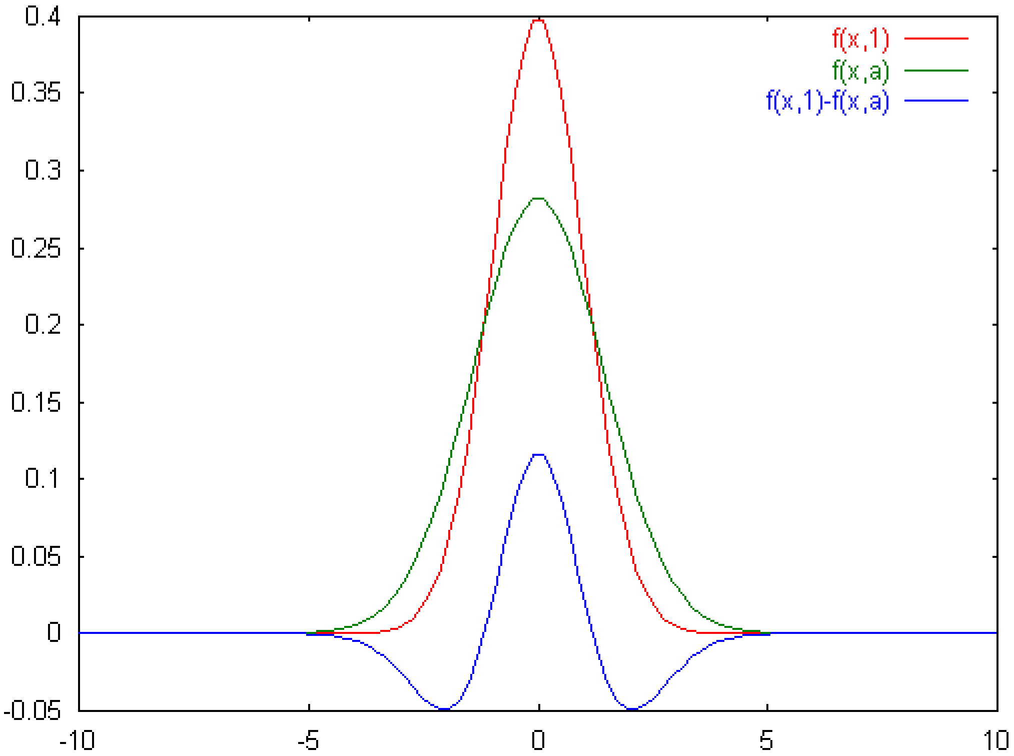
$$\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- The 'zero-crossing' correspond to the border points. An advantage of this technique is that the produced contour are closed.

The DoG operator

- This operator is called usually Difference of Gaussians (DoG)
- The best results are obtained maintaining the external Gaussian as large as possible but avoiding to include more than one border
- The internal Gaussian is optimized if it covers just the transition area
- Complex scene are better analyzed if a set of different DoG filters with various σ are applied.

The DoG operator



DoG Example

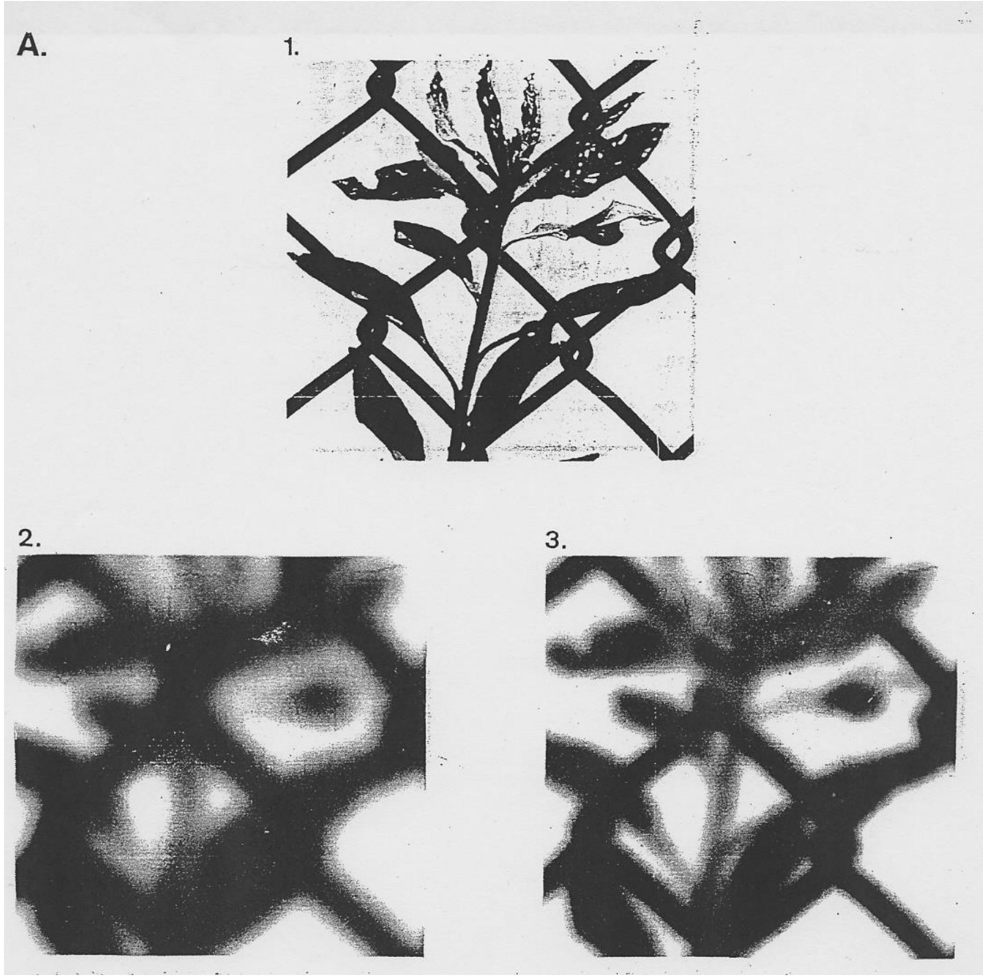
DoG Example

Gaussian Filter

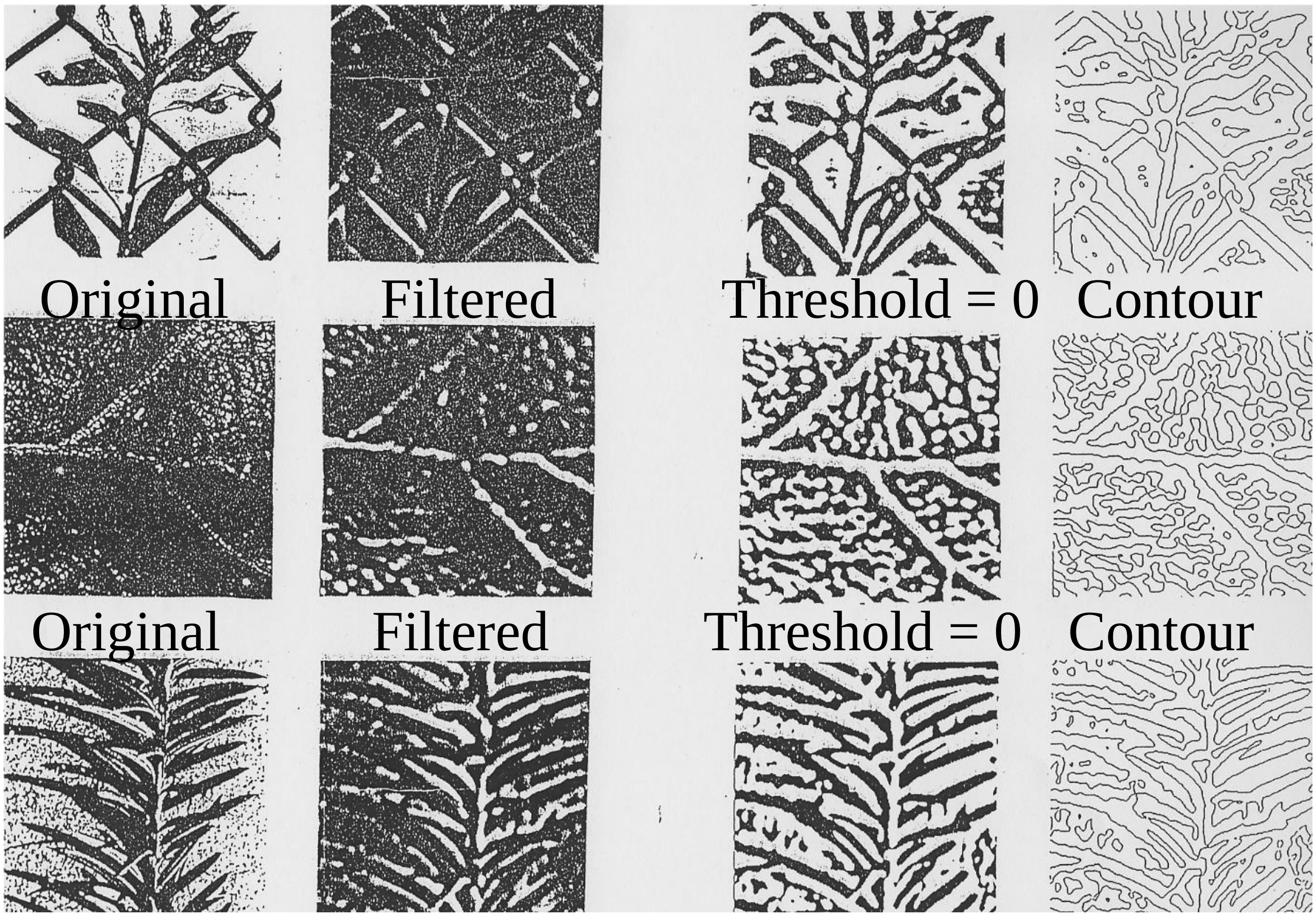
1 Original image

2 Filtered image σ=8

3 Filtered image σ=4



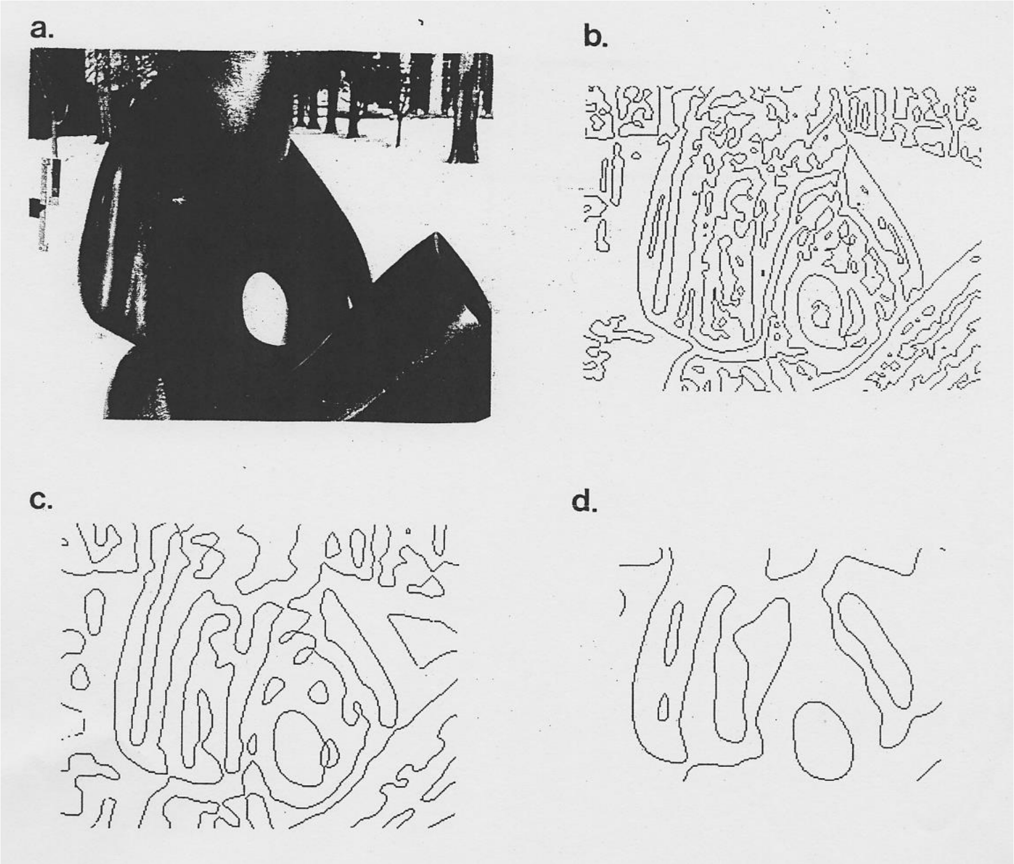
DoG Filter


Original
Filtered
Threshold = 0
Contour
Original
Filtered
Threshold = 0
Contour

DoG: σ dependence

Original

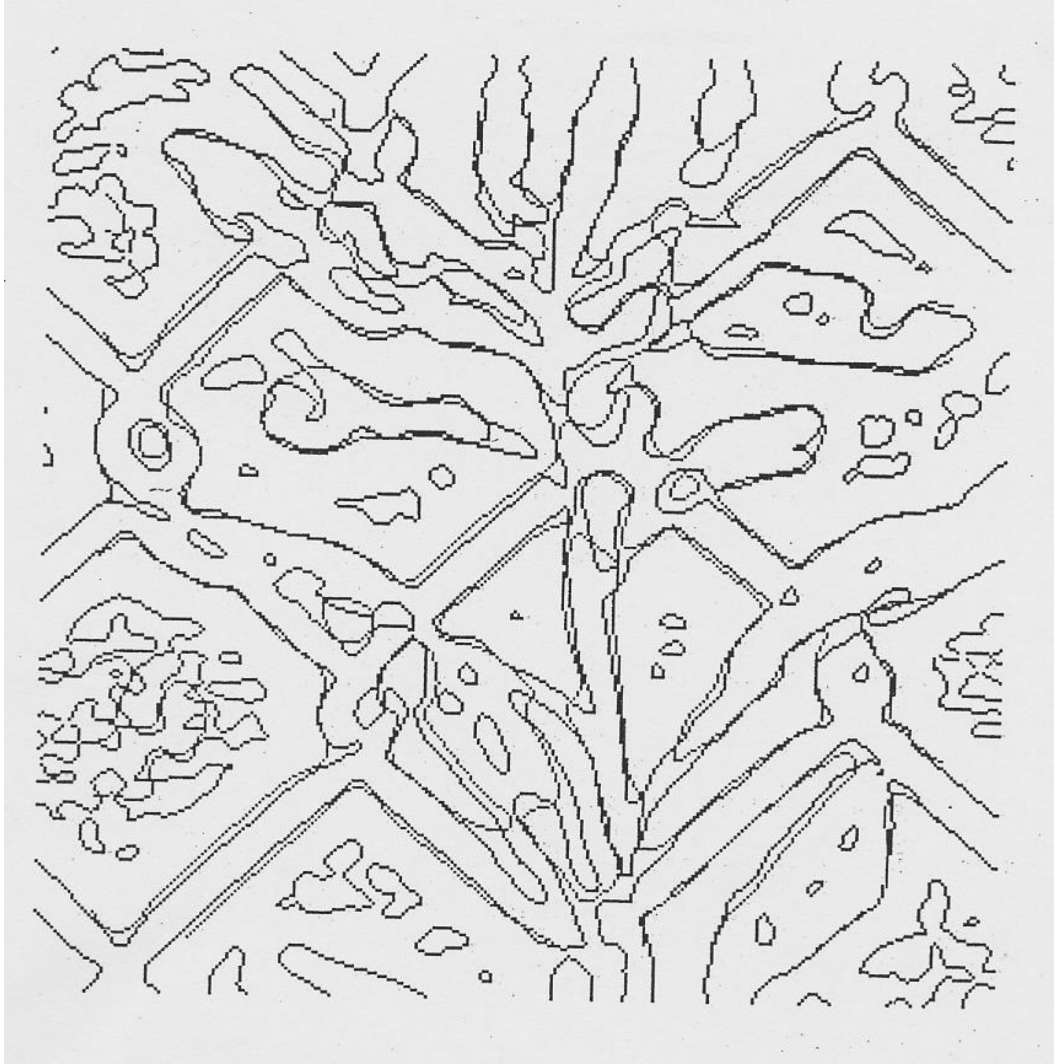
σ = 6



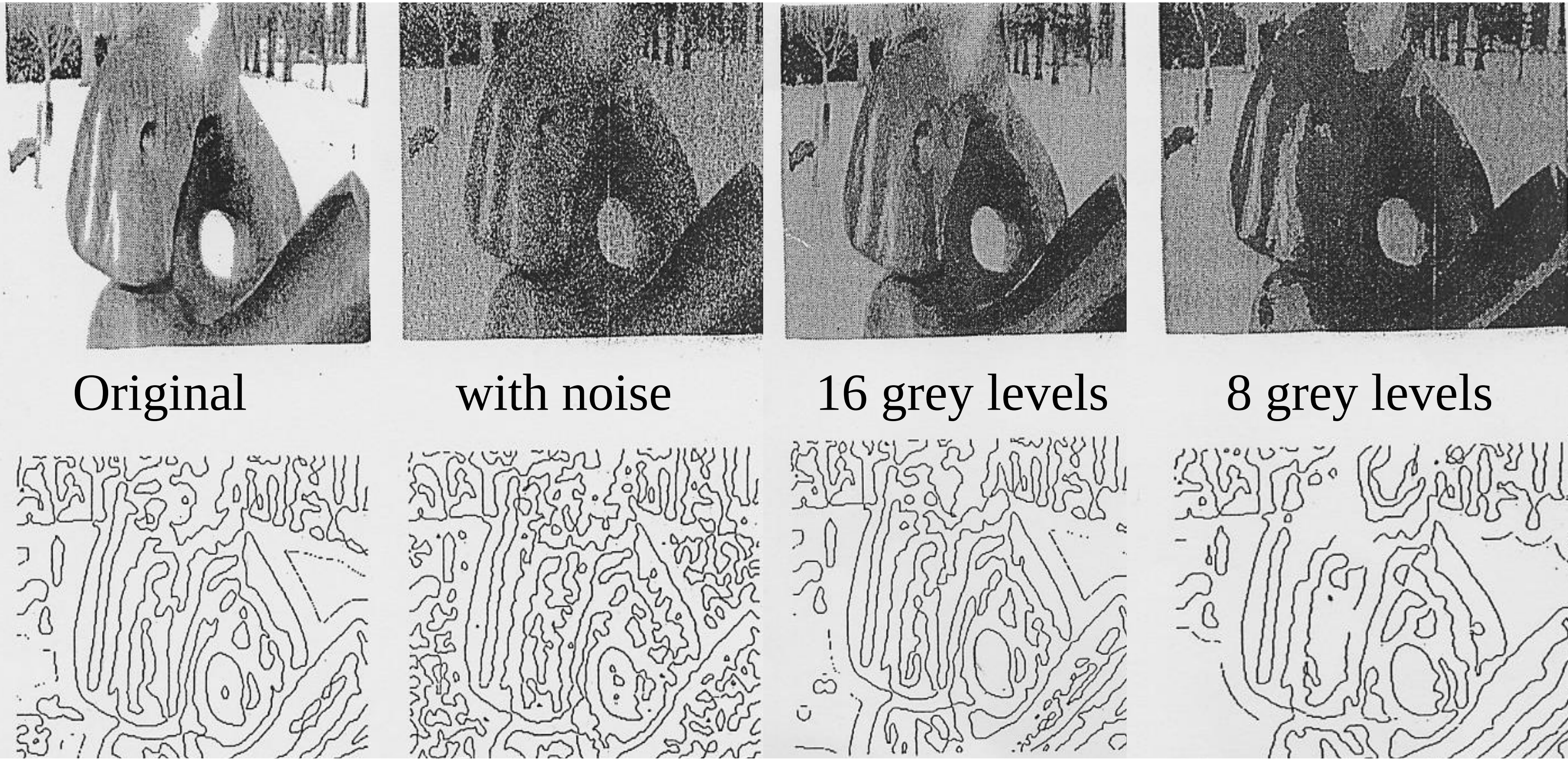
σ = 12

σ = 24

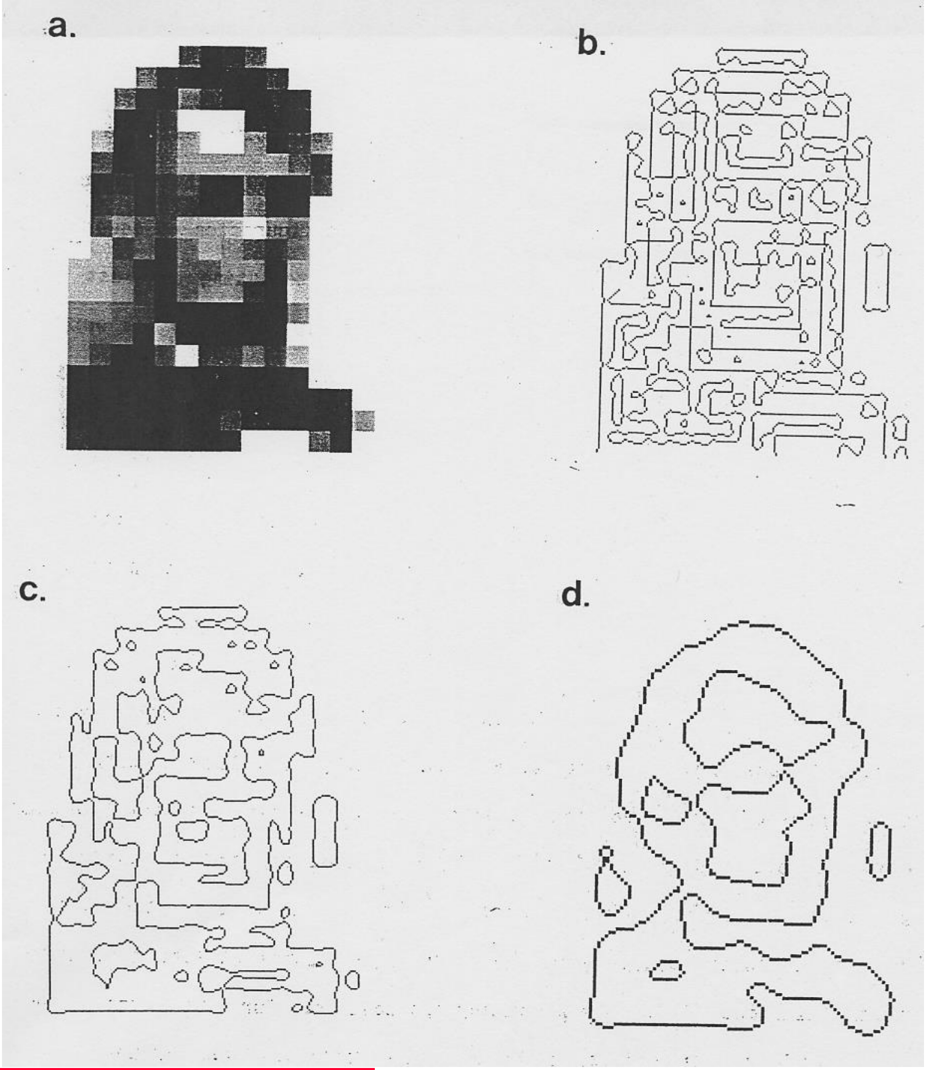
DoG: contour robustness



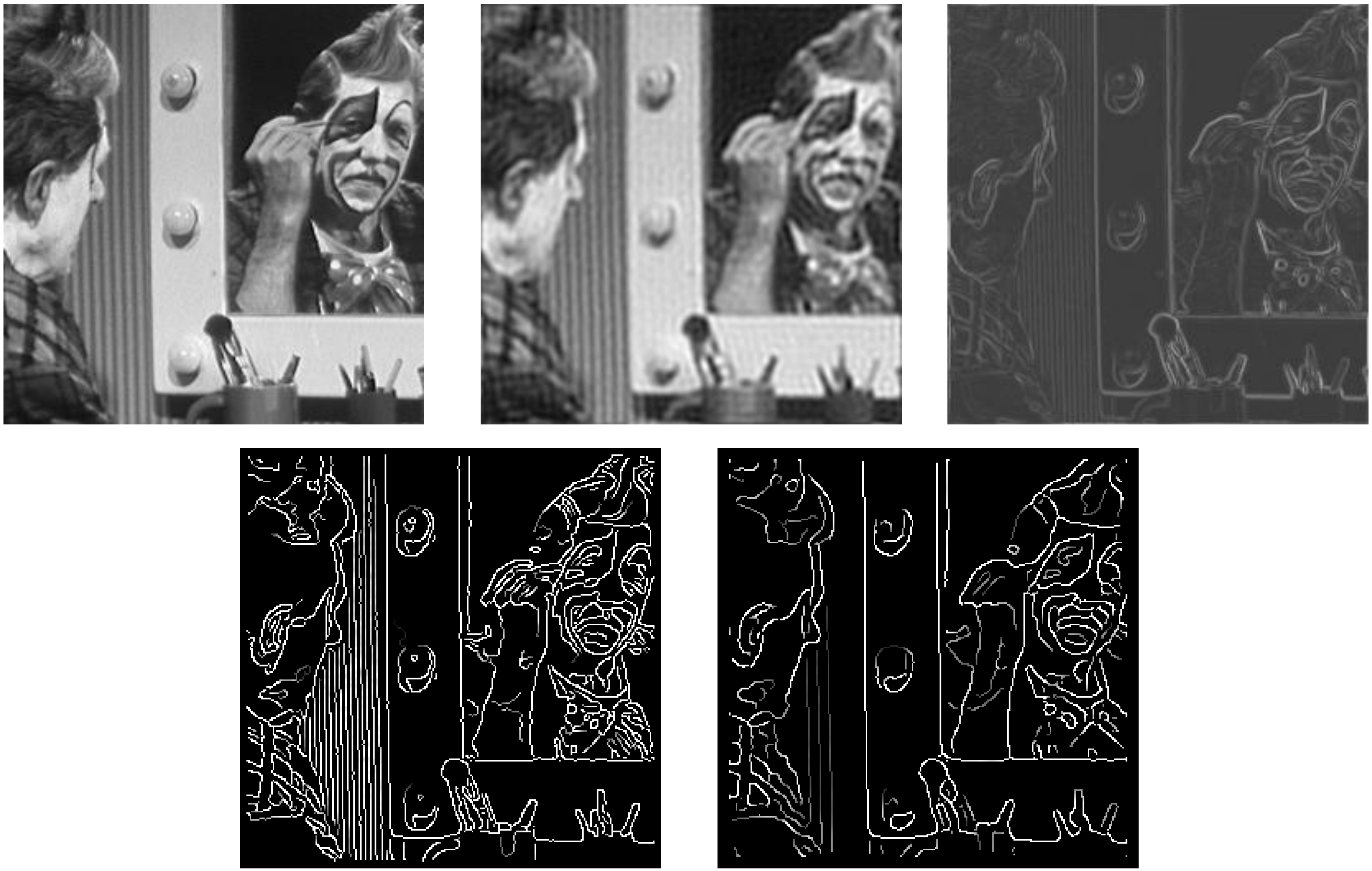
DoG: discretization of grey level and noise



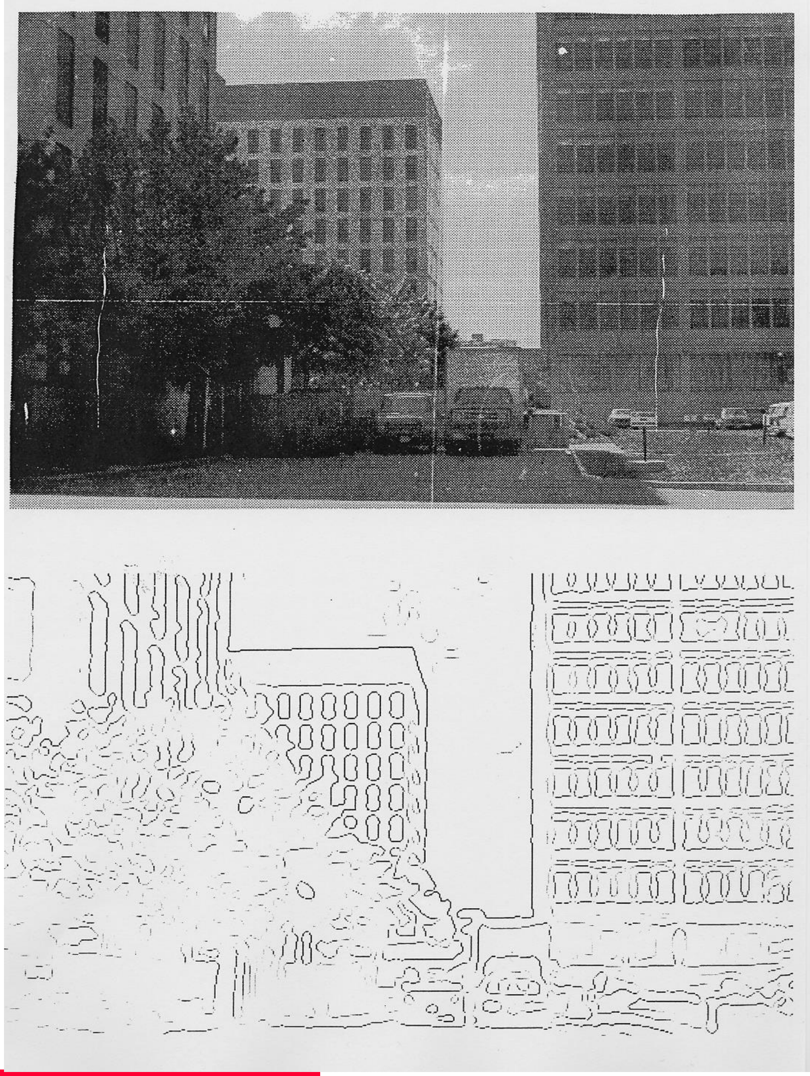
DoG: spatial discretization



John Canny, Rachid Deriche, etc operators



DoG + Sobel



DoG(2, 9)+Sobel

DoG(1, 9)+Sobel

Template Matching

- An alternative method for edge detection computes the closest (over all four/eight directions) approximations of *g(i,j)* in every 3x3 neighborhood, to keep the one with maximum convolution value, provided it is large enough
- Even if the sum of the kernel parameter is null note that starting with grey level images in the range 0:255 the final range is -3825:+3825 and -1275:1275 for Kirsh and compass operators respectively (the equivalent are -255:255, -765:765, -871:871, -1020:1020 for Roberts, Prewitt, isotropic and Sobel respectively)
- Obviously the greater is the number of values different from zero of the kernel parameters the higher is the robustness to noise.

Template Matching

- Kirsh’s operator

-3	-3	5	-3	5	5	5	5	5	5	5	-3
-3	0	5	-3	0	5	-3	0	-3	5	0	-3
-3	-3	5	-3	-3	-3	-3	-3	-3	-3	-3	-3
5	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
5	0	-3	5	0	-3	-3	0	-3	-3	0	5
5	-3	-3	5	5	-3	5	5	5	-3	5	5

- Compass operator

-1	1	1	1	1	1	1	1	1	1	1	1
-1	-2	1	-1	-2	1	1	-2	1	1	-2	-1
-1	1	1	-1	-1	1	-1	-1	-1	1	-1	-1
1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	1
1	-2	-1	1	-2	-1	1	-2	1	-1	-2	1
1	1	-1	1	1	1	1	1	1	1	1	1

3/9 operator

1	1	1
0	0	0
0	0	0

1	1	0
1	0	0
0	0	0

1	0	0
1	0	0
1	0	0

0	0	0
1	0	0
1	1	0

0	0	0
0	0	0
1	1	1

0	0	0
0	0	1
0	1	1

0	0	1
0	0	1
0	0	1

0	1	1
0	0	1
0	0	0

$$P_i = \sum_{j=0}^{8} I_{i,j}$$

3	2	1
4	8	0
5	6	7

$$P_{i,j} = I_{i,j} + I_{i,j-1} + I_{i,j+1}$$

Contour extraction

$$P = 1.5\left[\frac{P_{i,k}}{P_i} - 0.333\right]$$

- $P_{i,\,k}$ is the maximum among the 8 parameters $P_{i,j}$
- The coefficients 3/2 et 1/3 are introduced to normalize the result
- The final threshold can be applied depending on the minimum average contrast τ admitted in the neighborhood

Practical aspects of the 3/9 filter

- The filter implements a relative gray level intensity analisys. Also the human eye apply a similir approach.
- It must be payed attention when looking contours in the dark!
- Note that if P_i is low this edge estimation suffers very much for the effect of the noise (if it is 0 the edge value determined by the ratio 0/0!).
- Selecting the threshold for P_i note that it is 9 time di average intensity of the area (if this intensity is 10 - over 255 – so very low P_i is 90: edges are looked for in the very dark)

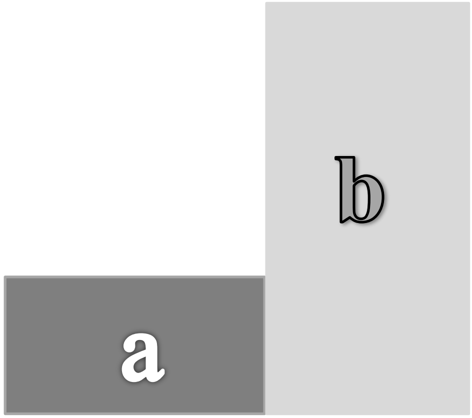
Contrast and threshold

- Let us call 'contrast' the ratio $\tau = \frac{a}{b}$, the threshold Th is given by:

$$P_i = \frac{3}{2}\left[\frac{3b}{6a + 3b} - \frac{1}{3}\right]$$

$$P_i = \frac{3}{2}\left[\frac{1}{2\tau + 1} - \frac{1}{3}\right]$$

$$Th = \frac{1 - \tau}{2\tau + 1}$$



Example: Op. 3 / 9

Example: Op. 3 / 9

Degraded image: uniform noise

- The standard model of this noise is additive, independent at each pixel and independent of the signal intensity with continuous uniform distribution in a given interval. The noise caused by quantizing the pixels to discrete levels has an approximately uniform distribution.

This noise can be simulated adding in each pixel $N(i,j) = 2K(rnd - 0{,}5)$ being K the noise intensity and *rnd* a random number with $0 \leq rnd \leq 1$

Degraded image: ‘salt and pepper’

- This is an impulsive or spike noise for which the image has dark pixels and bright pixels randomly distributed.

This noise can be simulated for each pixel in this way:

if $rnd \geq Th_1$ I(i,j) = 255

if $rnd \leq Th_2$ I(i,j) = 0

else N(i,j) = 2K(*rnd-0,5[1-th1+th2])* and if N(i,j)>255: N(i,j)=255, if N(i,j)<0:N(i,j)=0

being K the uniform component noise intensity, $0 \leq rnd \leq 1$, and Th_1 and Th_2 two suitable thresholds (1-Th_1 and Th_2 are the percentage of extra white and black pixels respectively)

Average value filter

- Each pixel takes the average value over the neighbors (3x3 in the example)

- Example - given the neighborhood:

3	6	8
3	4	2
5	8	3

the central pixel will take the new value:

(3+6+8+3+4+2+5+8+3)/9 = 4.67

Average value filter: uniform noise

Noisy image

Filtered image

Second iteration

Average value filter: uniform noise

Noisy image

Filtered image

Second iteration

Average value filter: salt and pepper

Noisy image

Filtered image

Second iteration

Average value filter: salt and pepper

Noisy image

Filtered image

Second iteration

Median and rank filters

- The median filter assigns to pixel the median value of neighborhood
- It is a particular case of the *rank filters* family, in which to the pixel is assigned the average value over a predefined range of the neighbors histogram.
- The average excluding the extremes is suited for impulse or spike noise such as the salt and pepper case.

- Example - given the neighborhood:

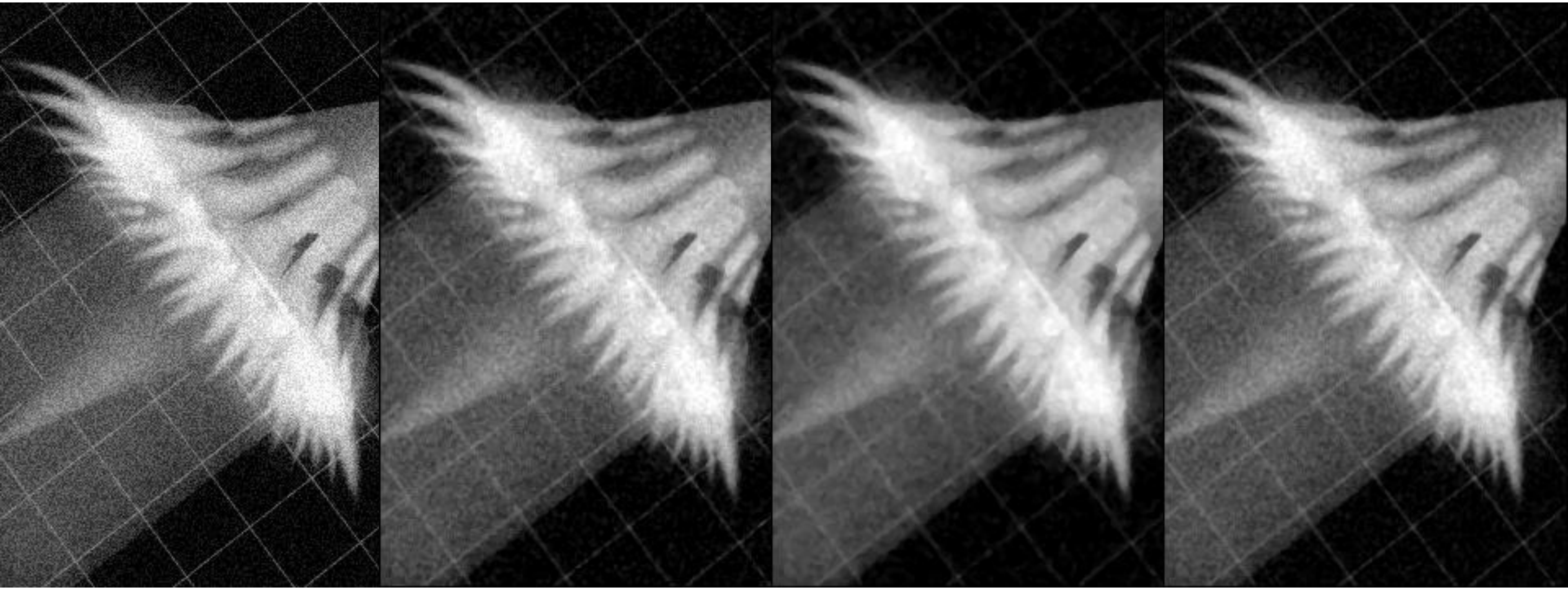
3	6	8
3	4	2
5	8	3

the correspondent values are:

2	3	3	3	4	5	6	8	8

median value: 4;
over three values: 4;
over five values: 4,2;
over seven values: 4,57
over nine values: 4,66

Median filter: uniform noise



Noisy image | Filtered image | Second iteration | Rank 3

Median filter: uniform noise

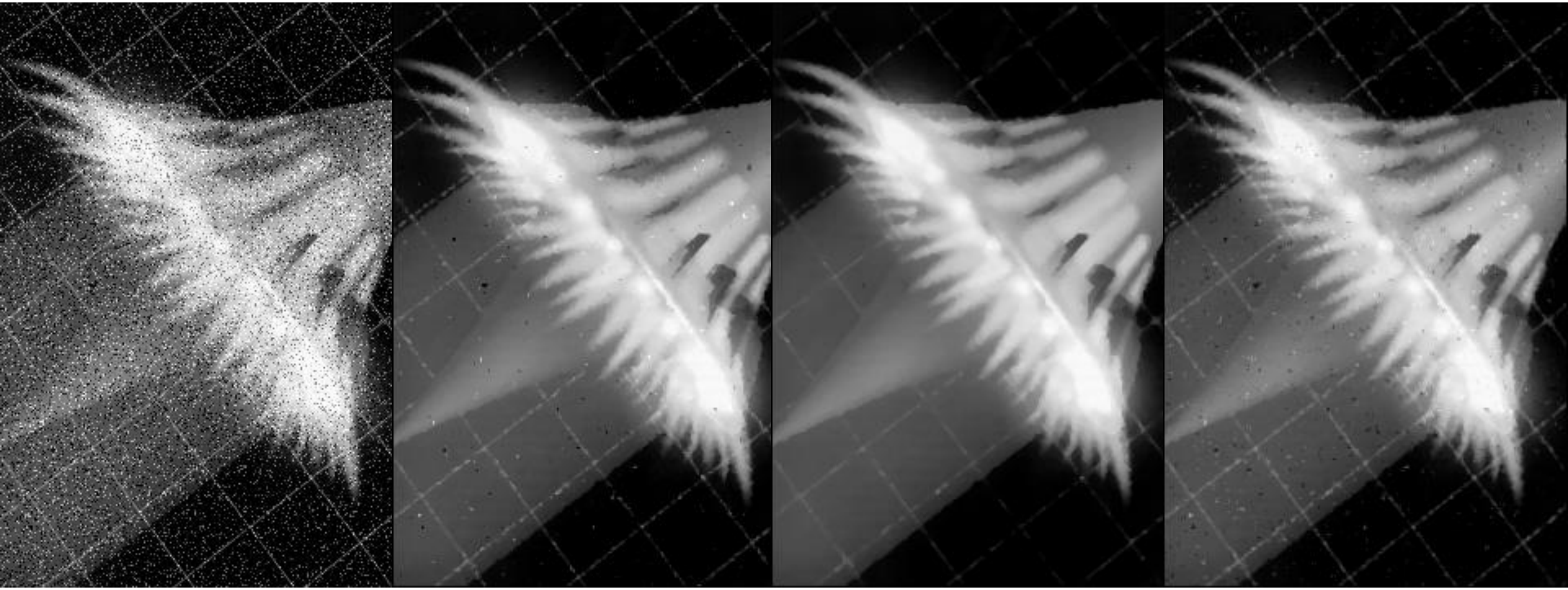
Noisy image

Filtered image

Second iteration

Rank 3

Median filter: salt and pepper



Noisy image Filtered image Second iteration Rank 3

Median filter: salt and pepper

Noisy image

Filtered image

Second iteration

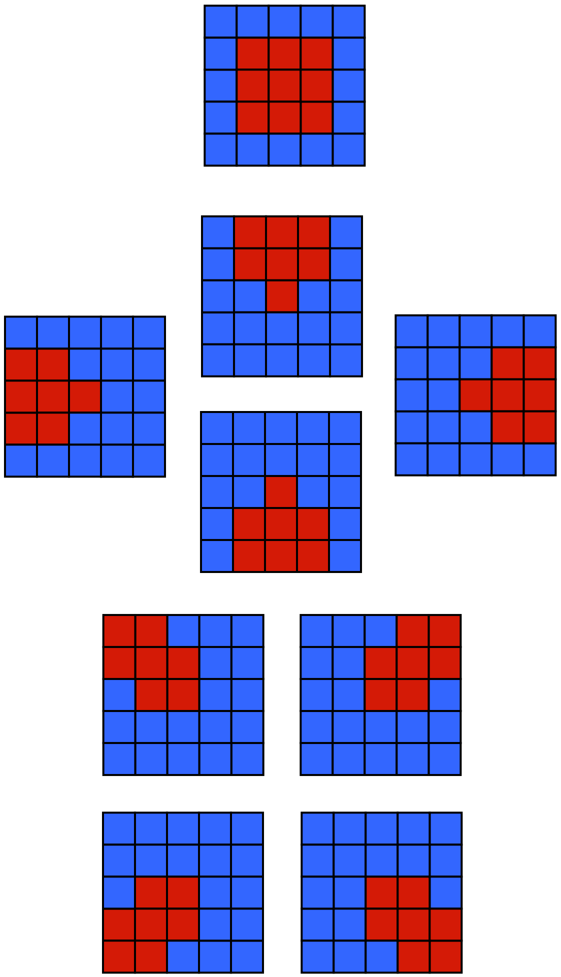
Rank 3

The Nagao-Matsuyama Filter

This filter selects for the centre pixel the average for the orientation with the least variation. Hence, the steps are as follows:

1. Calculate the variance for each of the nine sub-groups shown to the right (including the centre pixel).
2. Determine the sub-group with the lowest variance.
3. Assign the mean of this sub-group to the centre pixel.

Nagao-Matsuyama improves the borders, and is effective at reducing the edges smoothing. Clearly there is a cost in terms of computation due to the calculation of nine variances for each pixel.



Nagao filter: uniform noise

Noisy image

Filtered image

Nagao filter: uniform noise

Noisy image

Filtered image

Nagao filter: salt and pepper

Noisy image

Filtered image

Nagao filter: salt and pepper

Noisy image

Filtered image

Examples

Examples