

Theory and Practice of Digital Image Processing

Matthias Jung

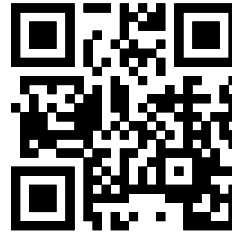
18. November 2010

1 Introduction

Fields of application for DIP:

- *Quality control:* defect detect., surface control
- *Assembly automation, robotics*
- *Production:* sorting, surveillance
- *Medical:* radiology, ultrasono. , tomography
- *Biology, Metallurgy:* eval. microscopic images
- *Criminology:* finger prints, identikit
- *Meteorology:* evaluation of satellite images
- *Security:* iris-, finger-, bodyscanner...
- *Automotive:* night vision, object recognition...
- *Daily Life:* bar code, deposit machines...

Examples:



URL jung.ms as QR-Code

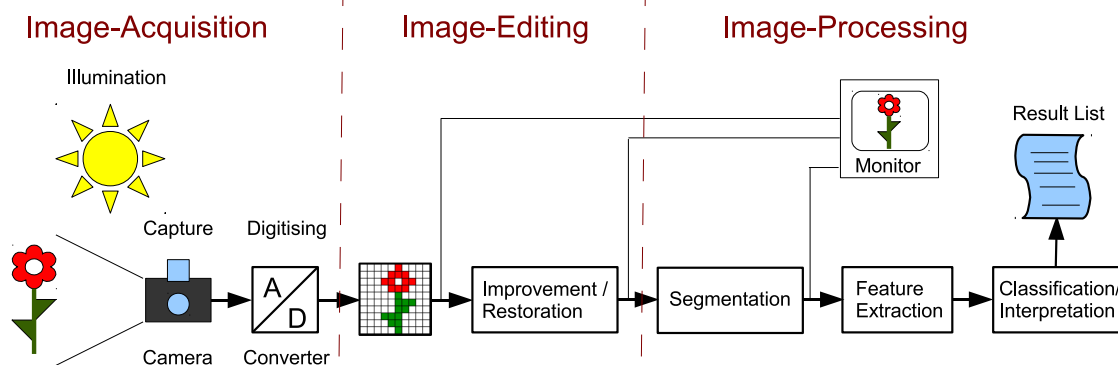


Ultrasonography



Barcode

Image Processing Pipeline:



Pipeline Stages:

- **Image Acquisition:**
placement of objects, illumination, capturing
- **Image Editing:**
De-noising, Shading correction, filtering
- **Image Processing:**
Segmentation: separation from objects and rest
labeling, feature extraction, classification

Tasks of Digital Image Processing:

- object detection
- object classification (recognition)
- measuring object positions or orientation
- counting of objects
- acquisition of object features:
 - geometrical dimensions
 - surface texture
 - surface characteristics

2 Illumination

The most important and critical part at the very beginning of the image processing task is the correct selection of the illumination.

Characteristics:

light intensity (Candela [cd]):	I
light density [cd/m^2]:	$L = \frac{I}{A}$
luminous flux (Lumen [lm]):	$\Phi = \int I(\Omega) d\Omega$
	$\Omega = \frac{A}{r^2}$
specific light emittance [lm/m^2]:	$M = \frac{d\Phi}{dA_s}$
quantity of light [$lm \cdot sec$]:	$Q = \int dt$
luminance (lux[lx]):	$E = \frac{d\Phi}{dA}$
	$E = I \cdot \frac{\cos(\alpha)}{r^2}$
light exposure [$Lx \cdot sec$]:	$H = \int E dt$

Light Sources:

- point light
- line light
- spot light (point spot, area spot) *most freq.*
- Area / Backlights
- Ambient (try to keep out ambient light!)

Characterisation of Light Sources:

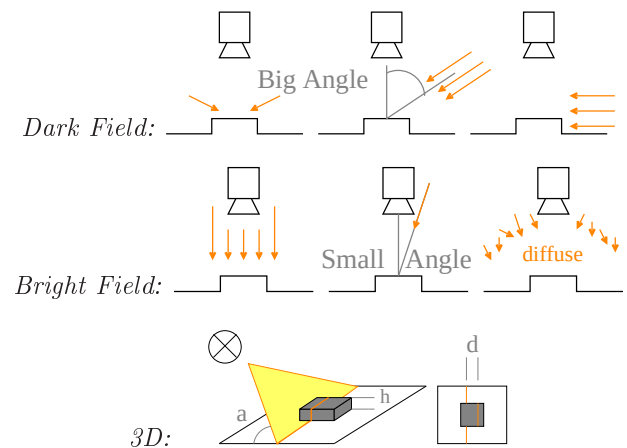
- duration of emission:
 - continuous
 - flash
- Spectral composition:
 - monochromatic (Laser, LED)
 - multi spectral (halogen lamp...)

Kinds of Illumination:

- incident light
 - *dark field*:
The desired structures are dark
 - *bright field*:
The desired structures are bright
- backlight (e.g. microscopy)
- 3D Measurement (e.g. Triangulation)
 $h = \tan(a) \cdot d$

Modelation e.g. with Phong model.

Examples:



3 Image Aquisition

Camera:

- *matrix* or *line* (e.g. scanner)
- *colour* or *monochrome*
- *rolling* or *global shutter*
(rolling = scanning line by line)
- sensor: *type* (CMOS, CCD) and *size*
- pixel *resolution*
- *frame rate*
- *interface* (FireWire or USB...)

CCD (Charge Coupled Devices):

- 2 phases:
 - ◇ capture (parallel)
 - ◇ output (serial)
- + quality

CMOS Sensor:

- flexible acquisition (random)
- + cheaper
- + less power consumption
- rolling shutter effect

Frame Grabber:

- analog-to-digital converter
- digitizing the video input signal online
- amplitude quantization of the image signal
- digitized image is transferred to RAM
- often supports low level image processing

Trend: Cameras have a dedicated digital output and USB interface

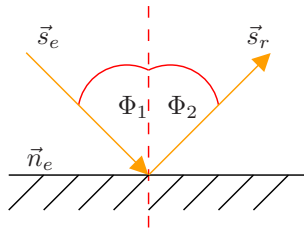
Technical Optics

- only geometrical optics
→ no wave characteristic
- emission as light rays (vectors)

Fermat Principle:

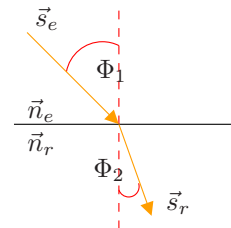
The optical path ($x \cdot n$) will be an extremum.
(Light takes the fastest way)

Reflection Law:



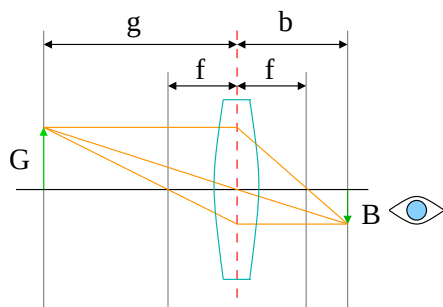
- e.g Mirror
- $\Phi_e < \text{critical angle}$
→ total reflection
- $\Phi_e = \Phi_r$

Reflection Law:

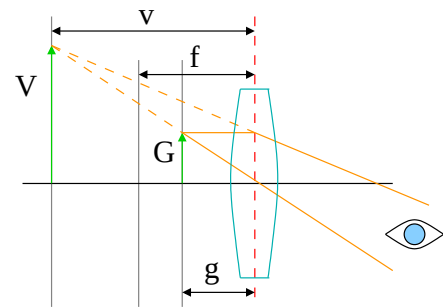


$$n_e \cdot \sin(\Phi_e) = n_r \cdot \sin(\Phi_r)$$

Real Imaging:

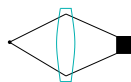


Virtual Imaging



Kinds of Imaging:

- Stigmatic: point in object space → point in image space
- Collinear: line in object space → line in image space
- Astigmatic (Reality): point in object space → area in image space



Thin Lenses:

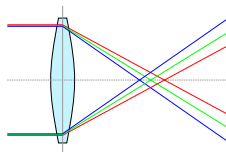
Refraction:

$$\frac{1}{b} + \frac{1}{g} = \frac{1}{f} = D \quad [\text{dpt}]$$

Imaging scale:

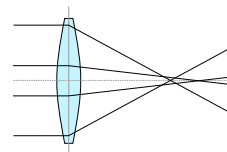
$$\beta = \frac{B}{G} = \frac{b}{g}$$

Chromatic Aberration (Imaging Errors)



- Refraction index depends on the wave length
- *longitudinal*: focal points of red and blue have different positions
- *lateral*: images for red and blue have different sizes

Spherical Aberration (Imaging Errors)



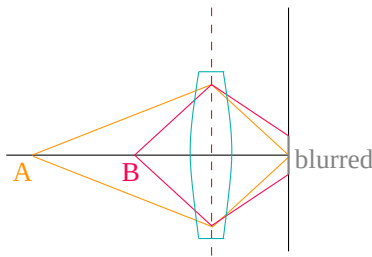
Abaxial rays are refracted more than the close ones

Coma:

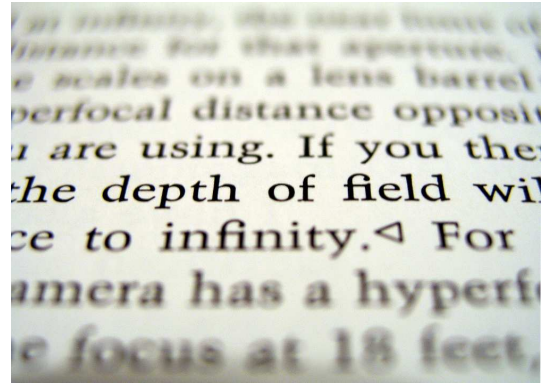
Point → oval figure with tail

Depth of Field:

- 3D-objects on 2D image → focus plane? → sharpness problem
- Reducing the problem by using an aperture
 - less light on the imager → more exposure time!



Example Depth of Field:



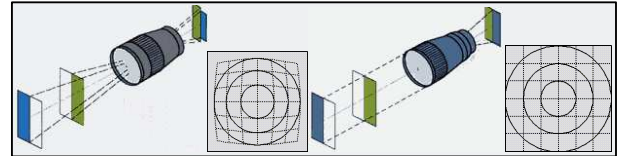
No aperture is used

Classes of Objectives:

super wide angle	100 °
wide angle	50 °
standard wide angle	35 °
standard	20 °
slight tele	12 °
tele	6 °

Telecentric Objective:

Aperture on image side (focal point) which can only be passed by axially parallel rays!



Sampling / Digitizing the Image:

- **discrete spatial sampling of the scene:**
partitioning the continuous scene information into discrete pixel information (Sensor, **Sampling Theorem**... overlap... dirac comb... $f_a > 2 \cdot f_g$...)
- **discrete magnitude sampling of the scene:**
partitioning the continuous colour information into quantized pixel information

How it is done?

The spatial sampling is done by the camera. In analog systems the magnitude sampling is done by a frame grabber and in digital systems from the camera itself.

Storing:

When storing image data for further processing, you have to make sure that you use no lossy compressing formats (e.g. JPEG). Better suited are RAW, BMP, ASCII or other not or lossless compressing formats.

4 2D-Image Processing:

4.1 Fundamentals

Image Enhancement: 1st Order

Operations restricted to *single individual pixels*

- histogram
- colour corrections
- (shading correction)

Image Restoration: 2nd Order

Operations on clusters of pixels (neighbourhood)

- linear filters
- morphology
- textures

Shading Correction:

Due to illumination, lense errors and sensors there can be unwanted shading. $X = X_w + X_s$ $X^* = X - X_s$
Shading correction by:

1. reference image (real/synth)
2. filtering (low pass filtering)
3. regression calc. equaling plane)

Basic Terms:

- grey value set: $G = \{0, 1, \dots, 255\}$ 8 Bit
- image function: $s(n, m)$
- image matrix: $S = [s(n, m)]$
 $n = \{0, \dots, N\}, m = \{0, \dots, M\}$
- spatial coordinates n, m (line, column)
- grey value of a pixel: $g = s(n, m) \in G$

4.2 Histogram

Absolute Histogram:

$$H_a(g) = \sum_{m=1}^{N_1} \sum_{n=1}^{N_2} \delta(g - s(n,m))$$

$$\delta(x) = \begin{cases} 1 & x = 0 \\ 0 & \text{else} \end{cases}$$

Explanations:

- A histogram gives the relative frequency of the grey values in an image
- A histogram considers only single pixels (1st order statistics)
- Two images with the same histograms need not to be identical

Brightness: mean grey value

$$\eta = \frac{1}{N_1 \cdot N_2} \sum_{g=0}^{255} g \cdot H_a(g)$$

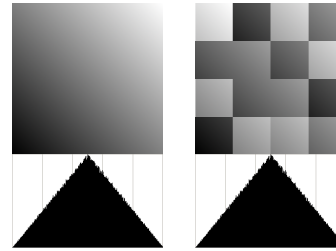
Contrast: variance

$$s^2 = \frac{1}{(N_1 \cdot N_2) - 1} \sum_{g=0}^{255} (g - \eta)^2 \cdot H_a(g)$$

Relative Histogram:

$$H_r(g) = \frac{H_a(g)}{N_1 \cdot N_2}$$

Example for Histogram:



Different Pictures

Same Histograms

Examples for Brightness and Contrast:



Normal

Bright

Contrast

Histogram Transformation: Transform grey values with a function f :

$$s^*(n, m) = f(s(n, m))$$

For instance: Thresholding, linear histogram transformation, partial contrast enhancement.

Thresholding:

$$f_t(s(n, m)) = \begin{cases} 1 & g(n, m) \geq t \\ 0 & g(n, m) < t \end{cases} \quad t \in G$$

- result: binary image
- information loss (8bit $\rightarrow$ 1bit)
- goal often pursued:
reduction of image content to relevant and not relevant (kind of segmentation)

Example Thresholding:



Linear Histogram Transformation:

$$f(g) = a \cdot g + b$$

Goal: Visual improvement of an image by changing the **brightness** or **contrast**. But there is no information gain!

The parameters a and b can be calculated as follows:

$$a = \frac{s_{new}}{s_{old}} \quad b = \eta_{new} - \frac{s_{new}}{s_{old}} \cdot \eta_{old}$$

Grey values are natural numbers (0...255) $\rightarrow$ truncation errors and information loss.

Histogram Spreading:

$$f(g) = (g - 0) \cdot \frac{M - m}{255 - 0} + m$$

$$\begin{matrix} [100 \dots 190] \\ m \quad M \end{matrix} \rightarrow [0 \dots 255]$$

Special linear histogram transformation

Histogram Smoothing:

$$H_r(g^*) = C$$

All grey values have the same count and the Area of the old histogram is the same as the area of the new histogram.

Partial Contrast Enhancement:

Linear Histogram Transformation only for a part of grey values

4.3 Linear Filters**Special Functions:**

Unit Impulse:

$$\delta(n, m) = \begin{cases} 1 & (n, m) = (0, 0) \\ 0 & \text{else} \end{cases} = \delta(n) \cdot \delta(m)$$

Unit Step:

$$\sigma(n, m) = \begin{cases} 1 & n > 0 \text{ and } m > 0 \\ 0 & \text{else} \end{cases} = \sigma(n) \cdot \sigma(m)$$

Properties of Image Signal Sequences:

- Separability: $s(n, m) = s(n) \cdot s(m)$
- Periodicity: $s(n, m) = s(n + k \cdot N, m + l \cdot M)$
- Causality: $s(n, m) = 0$ for $n < 0, m < 0$
- Linearity: $T\{a \cdot s_1 + s_2\} = a \cdot T\{s_1\} + T\{s_2\}$
- Shift Invariant:
 $x(n, m) \rightarrow y(n, m),$
 $x(n - 1, m - 1) \rightarrow y(n - 1, m - 1)$

FIR Filter:

$$y(n, m) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(n - k_1, m - k_2) \cdot h(k_1, k_2)$$

$$y(n, m) = x(n, m) ** h(n, m)$$

Implemented as a non-recursive algorithm. The result is independent of the predecessor.

IIR Filter:

$$y(n, m) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(n - k_1, m - k_2) \cdot h(k_1, k_2) - y(n - k_1, m - k_2) \cdot r(k_1, k_2)$$

Implemented as a recursive algorithm. The result depends on the predecessor.

Convolution:

$$\underbrace{s^*(n, m)}_{\text{output image}} = \underbrace{s(n, m)}_{\text{input image}} ** \underbrace{h(n, m)}_{\text{impulse response}}$$

The image is bounded:

- linear convolution (continuation with zeros)
- periodic convolution (continuation periodic)
- circular convolution (continuation periodic, half of the filter width)
- reflection convolution (continuation with the reflection, half of the filter width)

Linear Transformations

Spatial Domain:

$$s(n, m) = \sum_k \sum_l S(k, l) \cdot \underbrace{g(n, m, k, l)}_{\text{kernel}}$$

Transformed Domain:

$$S(k, l) = \sum_n \sum_m s(n, m) \cdot g'(n, m, k, l)$$

Different Transformations:

Fourier, Cosine, Wavelet, Haar ...

Discrete Fourier Transformation:

$$S(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} s(n, m) \cdot e^{-j2\pi(\frac{nk}{N} + \frac{ml}{M})}$$

$$s(n, m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} S(k, l) \cdot e^{j2\pi(\frac{nk}{N} + \frac{ml}{M})}$$

Speciality:

The circular/periodic convolution in spatial domain corresponds to a multiplication of the DFTs in frequency domain.

The circular/periodic convolution in frequency domain corresponds to a multiplication in spatial domain.

Discrete Cosine Transformation:

- DCT is *not* the real part of the DFT
- For correlated input signals the energy is stored in a few coefficients
- Reason why it is used for lossy data compression... Quantisation Table... (e.g. JPEG)

Digital Filters

- linear filters:
 - low-pass
 - high-pass
 - 2d Convolution
 - frequency domain methods
- non-linear filters:
 - rank order filters
 - adaptive procedures
 - morphology

Wavelet Transformation

- kernel is a wavelet (small wave)
- controllable trade-off between spatial and frequency resolution is possible
- often better for compression

Employment of Digital Filters:

- de-noising
- image enhancement and restoration
- edge enhancement, reconstruction and detection

Types of FIR filters:

- low pass filter (noise suppression)
- high pass filter (edge enhancement)

4.3.1 Low Pass Filters

Box Filter:

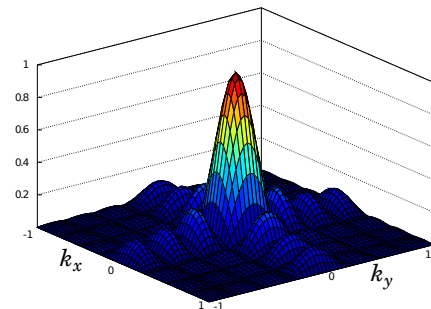
A box filter calculates the mean grey value over the region specified by the filter. A box filter with size N is defined as:

$$B_N = \frac{1}{N^2} \underbrace{\begin{bmatrix} 1 & \dots & 1 \\ \dots & 1 & \dots \\ 1 & \dots & 1 \end{bmatrix}}_N \Bigg\}^N$$

- + easy to implement
- not all high frequencies are suppressed
- an-isotropic (direction sensitive)
- causes phase shift in certain frequency ranges

Example:

$$B_8 = \frac{1}{64} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



Binominal Filter: A binominal filter can be looked upon as an approximation of the Gauss filter:

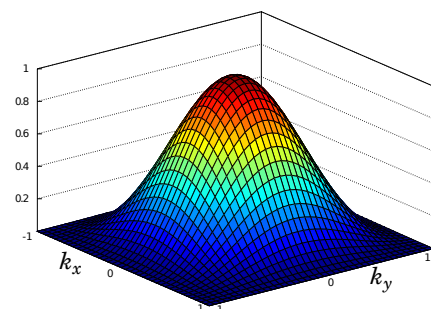
$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

In discrete case the impulse response is approximated by the binominal coefficients

- coefficients can be determined: pascal's triangle
- + filters are nearly isotropic
- + no phase shift
- more computation effort

Example:

$$B_8 = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$



Example:



original

box filter 5×5

binominal filter 5×5

Generally speaking, low pass filters perform a weighted averaging over the area of the filter mask.

4.3.2 High Pass Filters

High Pass Filters:

As the grey values at the edges in an image fluctuate strongly, their differential quotients in the regions are large. This fact is exploited in high pass filters. The filters which have a differentiating property. An edge in an image consists of high frequencies in the fourier domain.

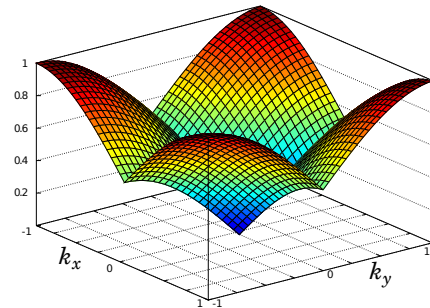
Difference Operator:

Simplest form of high pass filter which approximates the differential quotient in the x and y directions.

$$D_1 = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad D_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

- + Very Simple
- sensitive to noise
- an-isotropic

Transmission Function:

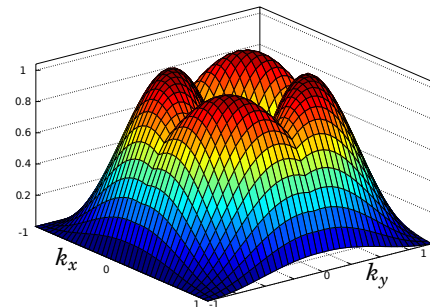


Sobel Operator:

an integration (low pass characteristic) is performed in the direction perpendicular to the direction of differentiation. This decreases the noise sensitivity considerably. Also a direction information is there.

$$S_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Transmission Function:



Kirsch Operator:

8 filter masks are employed and the maximum is taken. The direction information is lost.

$$K_1 = \begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix} \quad K_2 = \begin{bmatrix} 1 & -5 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & 3 \end{bmatrix} \quad K_3 = \begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix} \quad K_4 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{bmatrix}$$

$$K_5 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \quad K_6 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \quad K_7 = \begin{bmatrix} -5 & 3 & -3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \quad K_8 = \begin{bmatrix} -5 & -5 & 3 \\ -5 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}$$

Laplacian-of-Gaussian:

One important filter for edge detection is the LoG operator. The edge is detected using the second derivative ($=0$). At first the image is smoothed by a gaussian filter. Because both filters are linear they can be combined to a single filter.

The LoG delivers always closed loops of the edges (segmentation)

Examples:



Difference

Sobel

Kirsch

Laplace

Filter Design in Frequency Domain:

Setting out from the desired behavior in the frequency domain, the filter is designed by inverse transformation of the resulting frequency response.

Constraints in frequency domain lead to masks with infinite ranges. Windows are necessary → modification in frequency domain.

Windows commonly used:

Hamming, Hanning, Cosine windows.

5 Morphology and Non-Linear Filters:

5.1 Rank Order Filters / Median Filter:

Median Filter:

- linear filter: every pixel value (even an outlier!) contributes to the result
- median filter removes outliers

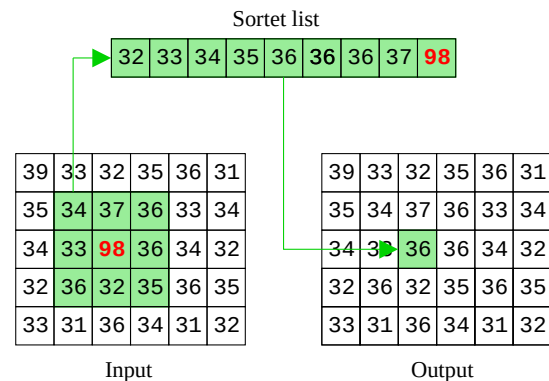
How it Works:

1. The pixel values in the region of interest (ROI) are listed in ascending order
2. The pixel value in the centre position of the list is taken to be the result of filtering

Pro and Contra:

- + *Salt and Pepper Noise* is eliminated
- + Outliers do not have any effect on the result
- + Edges remain sharp (are not blurred cp. low pass)
- fine structures are eventually lost
- corners are rounded
- computationally intensive

Example:



Example: Lena with Outliers



5.2 Binary Image Morphology:

Morphology:

Morphology is the study of forms. The idea is the interpretate images as set of structures. The basic operations are Erosion and Dilation. Mopology operators can be used for binary, grey value and coloured images.

Operation on Sets:

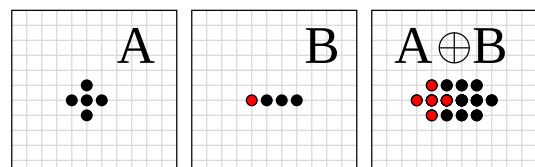
- Translation: $(A)_x = \{p \mid p = a + x, a \in A\}$
- Union: $A \cup B = \{p \mid p \in A \vee p \in B\}$
- Intersection: $A \cap B = \{p \mid p \in A \wedge p \in B\}$
- Complement: $A^c = \{p \mid p \notin A\}$
- Transposition: $A^\sim = \{-p \mid p \in A\}$

Dilation: (Ausdehnen, Erweitern)

$$A \oplus B = \bigcup_{a \in A} (B)_a$$

Each point in A generates a copy of the structuring element B . The dilation is the union of the copies.

Example:

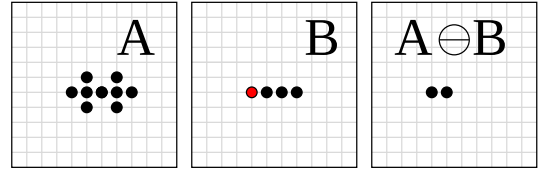


Erosion: (Abtragen, Aushöhlen)

$$A \ominus B = \{p \mid (B)_p \subseteq A\}$$

The result are all points where the structuring elements fit in.

Example:



Properties of Dilation:

- Commutative: $A \oplus B = B \oplus A$
- Asspcoative: $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
- Translation invariant: $(A)_x \oplus B = (A \oplus B)_x$
- Distributive: $(A \cup B) \oplus C = (A \oplus C) \cup (B \oplus C)$

The form of the structuring element influences the result of the operation!

Properties of Erosion:

- Translation invariant: $(A)_x \ominus B = (A \ominus B)_x$
- Distributive: $(A \cup B) \ominus C = (A \ominus C) \cup (B \ominus C)$

Properties:

Erosion and Dilation are dual operators, an erosion does not invert a dilation and vice versa

$$(A \ominus B)^c = A^c \oplus B^\sim$$

Opening:

Opening *opens* connected structures

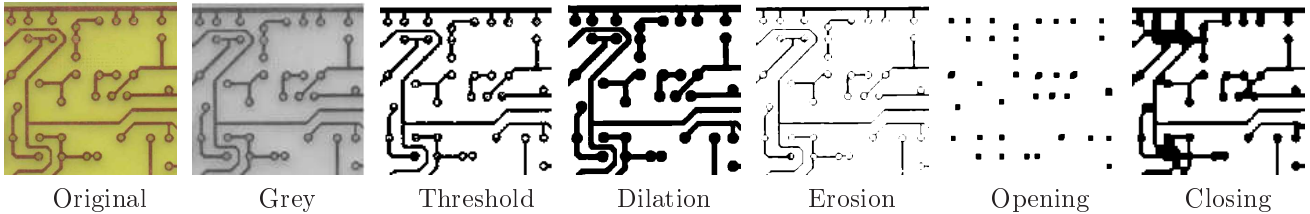
$$A \circ B = (A \ominus B) \oplus B$$

Closing:

Closing *closes* near structures

$$A \bullet B = (A \oplus B) \ominus B$$

Examples:



With opening we can find the drill holes in the circuit board.

Remarks:

$$\begin{aligned} A \circ B &= A \\ A \bullet B &= A \end{aligned}$$

$$\begin{aligned} (A \circ B) \circ B &= A \circ B \\ (A \bullet B) \bullet B &= A \bullet B \end{aligned}$$

$$(A \bullet B)^c = A^c \circ B^\sim$$

A is open/close w.r.t. B Opening and closing are idempotent Opening and closing are dual operations

Hit or Miss Transformation: Extension of $\circ$ & $\bullet$

The HOM transformation takes in addition to the foreground pixels also the background pixels into account! The structuring element is composed from two sets: X_f and X_b

$$A \odot X = \{p \mid (X_f)_y \subseteq A, (X_b)_y \subseteq A^c\}$$

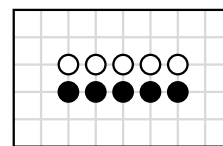
A pixel is part of the result if at this position the elements of X_f are completely within the object and the elements X_b are completely in the background.

Remarks:

$$X_f = 0 \rightarrow A \odot X = A \ominus X$$

$$X_b = 0 \rightarrow A \odot X = A \oplus X$$

Example: Circuit Board:



Structuring Element



Used as end point finder

5.3 Applications of Morphology

Morphological Gradient:

$$\begin{aligned}\nabla_i(A) &= A - (A \ominus B) && \text{inner edge of the object} \\ \nabla_o(A) &= (A \oplus B) - A && \text{outer edge of the object}\end{aligned}$$

Morphological Laplace-Operator:

$$\Delta_o - \Delta_i = (A \oplus B) + (A \ominus B) - 2A$$

Example:



Example:



Do $\rho_B(A)$ while something changes...

Morphological Reconstruction:

Conditional dilation: $A \oplus_B X = (A \oplus X) \cap B$
Dilate A and cut off all points which are in B.

Dilation with unity structuring element: $\delta_B^{(1)}$

n-th geodesic dilation: $\delta_B^{(n)} = \delta_B^{(1)}(\delta_B^{(n-1)})$

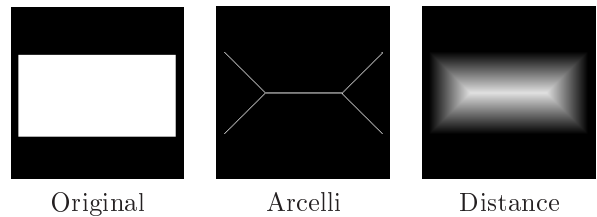
Reconstruction: $\rho_B(A) = \bigcup_{n \geq 1} \delta_B^{(n)}$

Skeletonisation: Arcelli Method:

$$s_1 = \begin{bmatrix} 1 & * & 0 \\ 1 & 1 & 0 \\ * & * & 0 \end{bmatrix} \quad s_2 = \begin{bmatrix} * & 0 & 0 \\ 1 & 1 & 0 \\ * & 1 & * \end{bmatrix}$$

Delete points when HOMT with s_1 and s_2 and 6x turn by 90° matches.

Example:



Skeletonisation: Distance Transformation:

The distance Transformation represents the mapping of a real number for each pixel which is the minimal distance to the edge

Granulometry:

The image is opened with structuring elements of increasing size. The difference between the original and the opened images is calculated after every stage (e.g. count of black pixels). Finally the results are normalised and plotted in a *Histogram of Particle Size*.

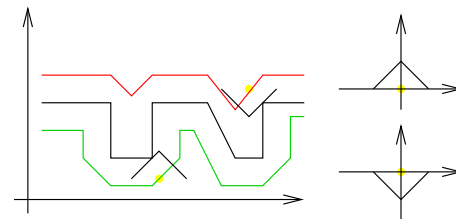
- similar to sieves with increasing diameters of holes
- determines the distribution of particle size
- useful to determine object sizes and number

Grey Value Morphology:

The image is a function (topographic relief) and the structuring element (also a function).

For erosion the structuring element wanders along this relief. Closing and Opening are similar to binary imaging.

Example:



Examples:



Original



Dilation
makes lighter



Erosion
makes darker



Open
suppress small
white structures



Close
suppress small
dark structures

Top Hat Transformation:

Suppresses coarse structures and highlight fine structures: $A - (A \circ B)$

E.g. projectile weapon comparison.

6 Segmentation and Edge Extraction: (1st Step in Information gathering)

Segmentation:

Segmentation is the process of transforming an image into a collection of disjunct elements which have some common property.

$$f \xrightarrow{\text{Segmentation}} \{f_n \mid n = 1, \dots, N\}$$

$$\bigcup_{n=1}^N f_n = f \quad f_\mu \cap f_\nu = \emptyset \quad \forall \mu \neq \nu$$

The individual seg. should be disjunct if possible.

Common Properties:

- Colour
- Grey value
- Texture
- Geometrical
- Parameters of Transformations
- ...

Applications: Separation of regions of an image based on colour; Separation of two textures...

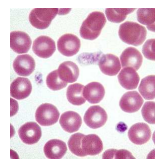
6.1 Property Procedures:

Segmentation by Thresholding:

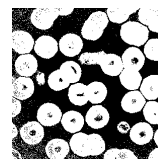
- + very simple technique
- automatic thresholding (e.g. Otsu)
- performance depends on brightness of the image
- eventually insufficient separation due to overlapping segments
- disturbances and shading
- elimination of disturbance with morphology

Thresholding methods are suited for segmentation of grey value and coloured images for which a histogram has been calculated.

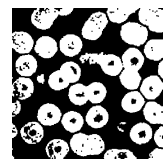
Example:



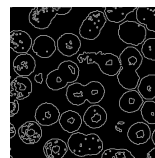
Blood cells



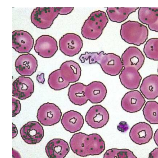
Threshold



Morphology



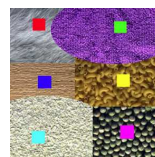
Edge detection



Overlay

Segmentation by Texture:

For each texture a feature parameter vector is defined. A texture is then assigned to a pixel depending on the range to which the texture parameters of the area around the pixel belong to.



Texture



Segmentation

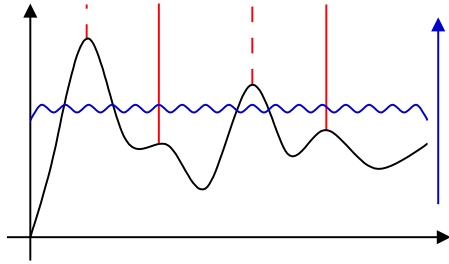


Morphology

6.2 Region Growing Procedures:

Watershed Algorithm:

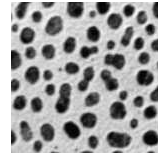
One can visualise that water is inundated gradually into a terrain which is formed by representing heights by grey values of the pixels. Lakes are formed which increase in area with the volume of water. A boundary is set when two watersheds meet.



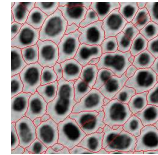
Region Growing with Distance Transform.

- distance transformation
- thresholding
- labeling
- growing (morph. reconstruction) w.r.t: **binary mask** or **statistics**

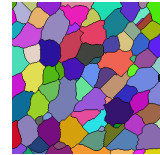
Example Watershed:



Sponge



Segmentation



Labeling

6.3 Clustering Methods:

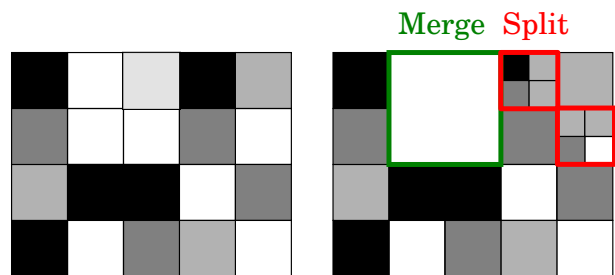
Split and Merge:

Split: Calculate the characteristics of a region (e.g. variance) if region not fulfill a criterion split it.

Merge: Compare a pixel with the neighbourhood and fuse if sufficiently similar.

Advantage: an erroneously assigned pixel can be removed from the region again.

Example:



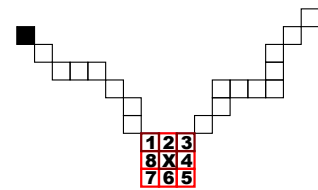
6.4 Edge Extraction:

Chain Code Freeman:

Representation for contour using direction data.

- + Compressed representation
- No direct coordination information available
- No rotation invariance (except 45° steps)

Example:



5-5-4-4-5-5-6-5-5-3-3-3-2-4-4-4-2-2-3-3

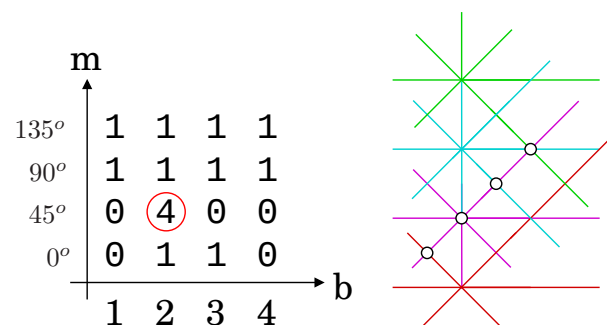
Straight line Extraction with Accumulator array:

$$\text{Straight line: } y = m \cdot x + b$$

Collect in an accumulator array how often the parameters m and b match to points in the image. The maximum refers to a straight line.

In practice the parameter space is discretized.

Example:



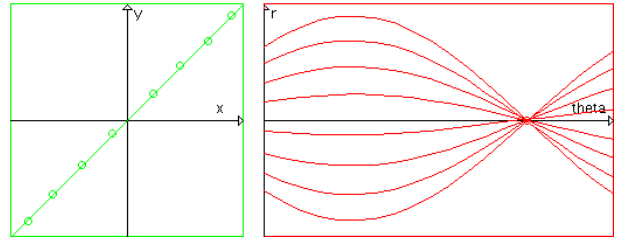
Hough Transformation:

$$x \cdot \cos(\Theta) + y \sin(\Theta) - r = 0$$

The parameters Θ and r specify the straight line to be represented.

- point $\rightarrow$ sinusoidal curve
- line $\rightarrow$ array of sinusoidal curves in a point
- point $\leftarrow$ line

Example:



6.5 Active Contours

Splines:

- compound of polynomes
- employed for representation of connected lines and contours
- employed for completing broken lines

Balloons:

- Extension of the snake model
- A balloon force is introduced which makes the snakes to grow

Snakes:

- employed for detection of edges and boundaries
- snake minimises the energy functional

One can imagine a spring system with forces:

- *internal energy:*
 - smoothness
 - elasticity
- *external energy:*
 - potential function: lines, edges, gaps
 - adjusting

7 Texture Analysis

Texture:

A texture is a structural appearances of surfaces such as those of textiles or of a wall or a chess board. They are characterized by a certain regularity *or* repetition.

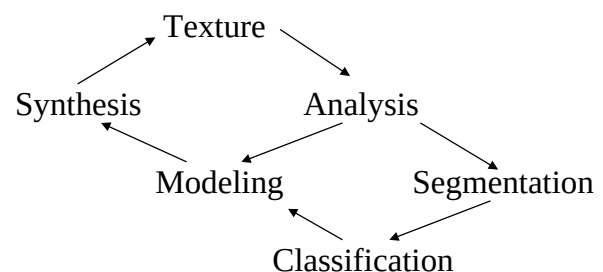
- consists of deterministic or stochastic patterns
- elementary structures, also named textons or texels which occur repeatedly in an image
- micro-/ macro pixels

Applications:

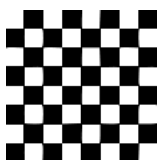
- Life science (tissue characterization in medicine)
- Quality control (surface inspection)
- Evaluation of satellite images ...

Tasks of Texture Analysis:

- *Modeling:* analytical description of texture
- *Segmentation:* division of a textured image into regions which have the same features of texture
- *Classification:* assignment of a texture to one of n texture classes



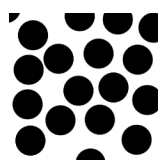
The three classes of texture models:



structural



stochastic



structured
stochastic

Structured Texture Model:

- are made of a defined elementary pattern g
- the configuration follows a strict scheme
- are not suited for modeling natural textures

$$f(x, y) = \left[\sum_{i=-b}^{+b} \sum_{j=-d}^{+d} \delta(x - N_1 \cdot i, y - N_2 \cdot j) \right] ** g(i, j)$$

Stochastic Texture Models:

- stochastic process: joint density function
- statistical parameters (1st and 2nd Order) can be used to characterize the texture

Difficult as the joint density function is not known and only one realization is available:

- use co-occurrence matrices or autocorrelation function

Structured Stochastic texture models:

- A texture primitive which is somewhat deterministic is randomly distributed w.r.t. space coordinates.
- Texture is not specified on the basis of the configuration but on that of the relative frequency of occurrence of elementary forms
- the situation is well suited for correlation analysis

Global texture features with stat. methods:

- First Order:
 - Mean value
 - Variance
- Second Order:
 - Auto Correlation (detection of periodicities)
 - Gauß Process
 - Poisson Process
 - Fractal model

Co-Occurance Matrices:

A co-occurrence matrix gives an estimate of the joint density function with a shift vector: (dx, dy) . The dimension of the co.oc. matrix is the count of different grey values. The normalization factor is the count of the transitions. (↘)

$$C_{dx,dy}(x, y) = \sum_{i=0} \sum_{j=0} \delta(A(i, j) - x) \cdot \delta(A(i + dx, j + dy) - y)$$

Examples Co-Occurrence Matrix:

$$A = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 2 & 2 & 2 & 1 \\ 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \nwarrow \hat{=} \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$C_{dx,dy} = \frac{1}{9} \begin{matrix} & \begin{matrix} 0 & 1 & 2 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 3 & 1 \end{bmatrix} \end{matrix}$$

Haralick Features:

Information in the co.oc. matrix can be condensed to features:

- Homogeneity: $m_1 = \sum_i \sum_j (C(i, j))^2$
- Contrast: $m_2 = \sum_i \sum_j C(i, j) \cdot (i - j)^2$
- Entropy: $m_3 = \sum_i \sum_j C(i, j) \cdot \log(C(i, j))$
- Correlation ...

Texture Filters:

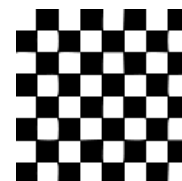
Texture filters are employed to highlight certain structures or features in the texture and are useful for feature extraction as a preliminary step for classification.

The input image is filtered with each of N filters. The impulse response of the filters correspond to the features which are expected in the input image.

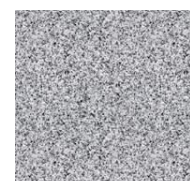
Examples Haralick:



H = 1
E = 1.2



H = 0.3
E = 9.0



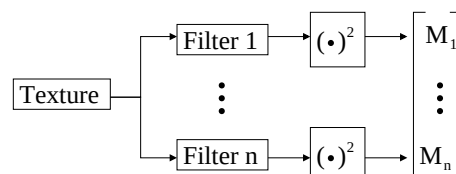
H = 0.0004
E = 0.05

Laws Filter:

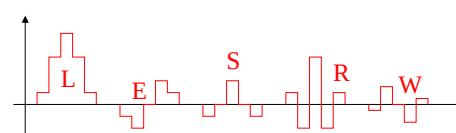
- Empirical filter choice for dominant features
 - 5 one dimensional filter kernels
- 25 separable 2d filters, **may be correlated**

L: homogeneous, **E**: edge, **S**: spot & points,
R: ripples, **W**: waves

Example Texture Filters:



Example Texture Filters:



Ade Eigenfilter:

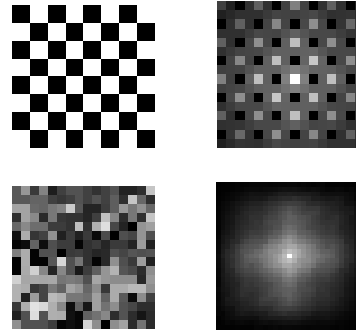
Filter design by eigenvectors of covariance matrix. The filters are orthogonal and uncorrelated!

Chen-Dai Least Square Filter:

The original image is assumed to consist of the superposition of R different known and mutually disjunct textures t_i . Then filters are designed based on these textures.

Auto Correlation:

Auto correlation is a powerful tool for analyzing features in structured stochastic images. Autocorrelation detects periodicities.

Auto-correlation:

8 Feature Extraction:

Feature Extraction:

The term feature implies a quantitative measure of some property of the object. Features are the basis for *classification*. A feature should have following properties:

- **Separation Sharpness:** The feature should be distinctly different for different objects
- **Reliability:** The feature should vary little for one and the same object
- **Independence:** The variations of the features of two similar objects should be statistically independent. This ensures that every feature has some additional information
- **Low number:** dependence increases with the number of features, therefore the number of features should be small

8.1 Geometrical Features:

Area:

The Area A of a labeled object L is given by the number of pixels belonging to it.

$$A = \sum_{x=0}^N \sum_{y=0}^M \delta(g_L - g(x, y))$$

Perimeter:

The perimeter U of an object is given by the number of pixels in the chain code. A better result is obtained to consider the count of straight transitions k_s and diagonal transitions k_d :

$$U = k_s + \sqrt{2} \cdot k_d$$

Compactness:

The compactness K gives the degree of similarity to a circle. ($K = 1$ is a circle)

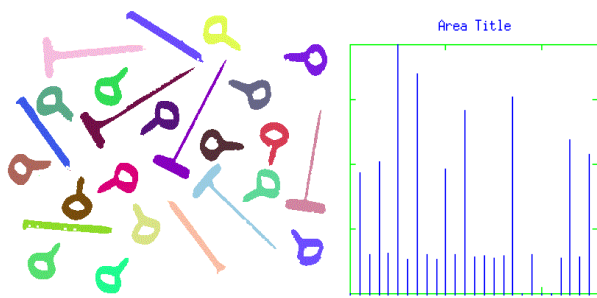
$$K = \frac{U^2}{4\pi A}$$

Euler Number:

The euler number is a topographical feature, which is defined as the difference between the connected areas C and holes H

$$E = C - H = K - V + F$$

Skeleton: K = edges, V = corners, F =areas

Example Area:**Example Euler:**

$$E = C - H = 9 - 2 = 7$$

8.2 Moments:

Moments:

Moments describe a figure completely, in the sense they suffice to reconstruct the object completely. The $(p + q)$ -Moment of an object $f(x, y)$ is defined as:

$$m_{p,q} = \int \int f(x, y) x^p y^q dx dy$$

Reconstruction is possible:

$$f(x, y) = \int \int e^{-j2\pi(x\psi_1 + y\psi_2)} \left[\sum_{p=0}^{\infty} \sum_{q=0}^{\infty} m_{p,q} \cdot \frac{(j2\pi)^{p+q}}{p!q!} \psi_1^p \psi_2^q \right] d\psi_1 d\psi_2$$

Eccentricity: (Unwucht)

$$\varepsilon = \frac{(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}}{A}$$

$$\varepsilon_{circle} = 0 \quad \varepsilon_{line} = 1$$

Center of Gravity:

$$\bar{p} = (\bar{x} | \bar{y}) = \left(\frac{m_{0,1}}{m_{0,0}} \mid \frac{m_{1,0}}{m_{0,0}} \right)$$

Invariant Moments:

Translation invariance can be archived by calculating the central moments:

$$\mu_{p,q} = \int \int f(x, y) \cdot (x - \bar{x})^p \cdot (y - \bar{y})^q dx dy$$

Invariance to scale and rotation can also be achieved...

Orientation:

Orientation is given by the angular position of the main axis of an object:

$$\Theta = \frac{1}{2} \cdot \arctan \left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right)$$

8.3 Edge Based Features:

Complex Fourier Descriptors:

Any contour of length N can be described as a sequence $(x(k), y(k))$ $k = (0 \dots N)$. We can write this as complex notation:

$$s(k) = x(k) + j \cdot y(k)$$

Then transform the function $s(k)$ in fourier domain to get the complex fourier descriptors a :

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \cdot e^{-\frac{2j\pi uk}{N}}$$

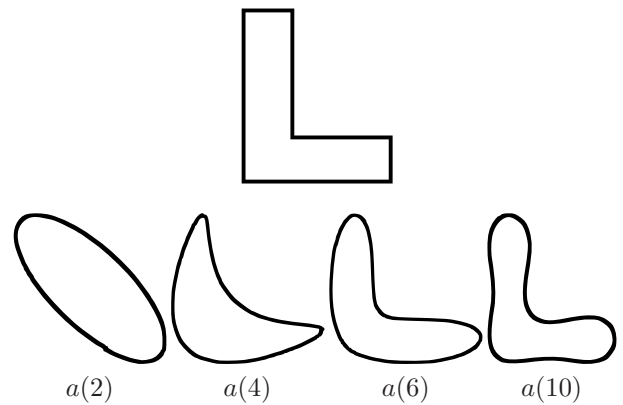
The inverse transform with M fourier descriptors leads to a closed curve contour:

$$\hat{s}(u) = \sum_{k=0}^{M-1} a(u) \cdot e^{\frac{2j\pi uk}{N}}$$

Invariant Complex Fourier Descriptors:

Translation Invariance can be archived by the first F.D. pair which gives the center of mass. The rotation is given by the second pair. So the other coefficients can be modified.

Example:



Other Edge Based Approaches:

- Ψ -S-Diagram
- Polygons
- Real Fourier Descriptors (with angle...)

9 Classification:

Classification:

Classification is the process of assigning a given object to a class of similar objects. The metric used for similarity is based on features of the object.

Procedure:

First feature parameter are collected using methods of feature extraction and represented as feature vectors. Feature vectors are then taken to characterize the objects.

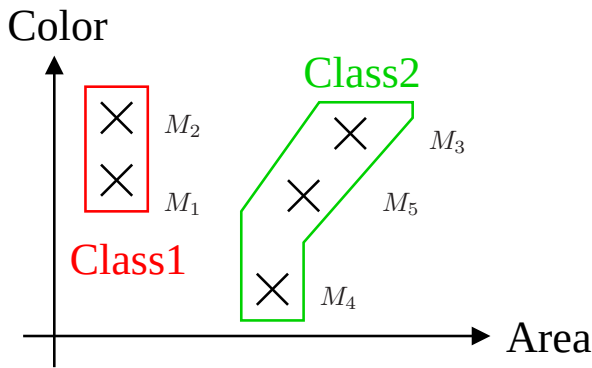
Feature Vector:

Possible entries in feature vector:

$$\vec{M} = \begin{pmatrix} \text{perimeter} \\ \text{area} \\ \text{compactness} \\ \text{euler number} \\ \text{variance} \\ \text{colour} \\ \text{texture} \\ \text{contour} \\ \dots \end{pmatrix}$$

Feature Space:

The N-features in a feature vector span an N-dimensional feature space:



Properties:

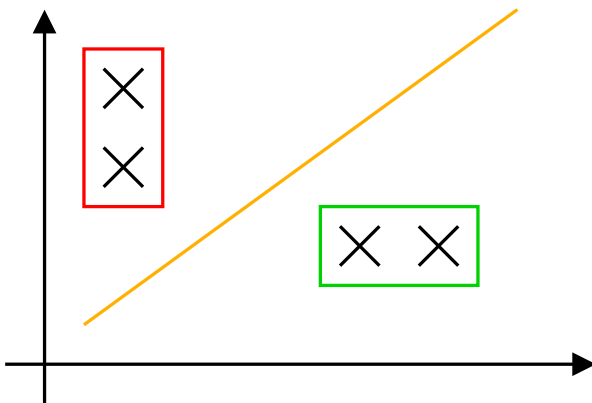
Objects with similar features lie closely in f.s.. The goal is to achieve a unique separation of the classes.

Often the N-features are not linearly independent or are not sharply separated. In such cases the dimension of the feature space is reduced to the number of distinctly different features. (e.g. Karhunen-Loeve Transformation)

An alternative is to optimize the features w.r.t their use for classification.

Features with distinctly different means and small variances are good for separating the classes.

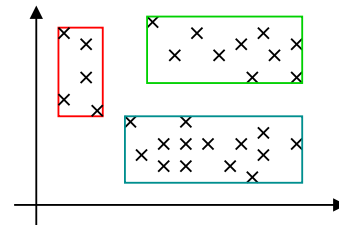
Separation by a Straight Line:



in two dimensional space separation by a straight line is possible

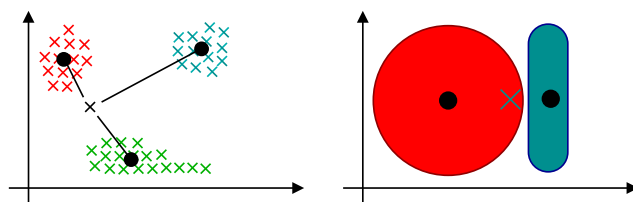
Parallelepiped classifier:

1. training
 2. determine min/max of class limits (boundaries)
 3. classification via inclusion / exclusion
- + simple
 - + boundaries parallel to coordinate axes
 - useless if there is overlap
 - outliers in training can lead to overlap
 - outliers in classification stay unclassified



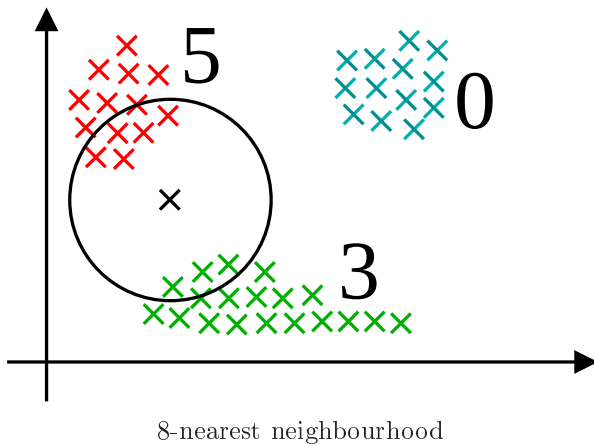
Minimum distance Classification:

1. training
 - 2.
- + simple
 - + efficient by compact classes
 - inefficient when classes have big variance
- mean does not suffice

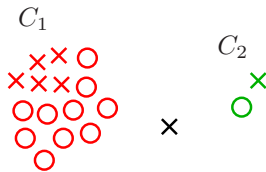


k-Nearest-Neighbour Classification:

1. an object is assigned to a class in which it has the largest number (k) of neighbours in the class.
- + no training
 - + in principle arbitrarily complex class boundaries possible
 - class representants should cover the region well
 - search effort rises with the number of elements in the clusters



Example:



Optimal Statistical Classification (Bayes):

Searched: probability that a feature vector x belongs to a class C_i :

$$p(C_i | x)$$

Wrong assign. to C_j instead C_i is loss: $L_{i,j}$ Mean loss:

$$r_j(x) = \sum_{k=01}^M L_{k,j} \cdot p(C_k | x)$$

Application of Bayes Theorem ($p(a|b) = \frac{p(a)p(b|a)}{p(b)}$):

$$r_j(x) = \sum_{k=01}^M L_{k,j} \cdot \underbrace{p(x | C_k) \cdot p(C_k)}_{\text{known training}}$$

A feature vector is assigned to a Class C_i with the lowest $r_i(x)$.

In many cases: loss is:

$$L_{ij} = \begin{cases} 0 & \text{for } i = j \\ 1 & \text{for } i \neq j \end{cases}$$

→ Vector is assigned to a class C_i if:

$$\underbrace{p(x|C_i) \cdot p(C_i)}_{d_i} > p(x|C_j) \cdot p(C_j) \quad \forall j \neq i$$

Is vector x assigned to class C_1 or to class C_2 ?

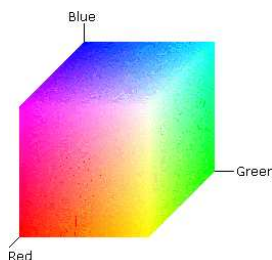
$$d_1 = p(x|C_1) \cdot p(C_1) = \frac{6}{16} \cdot \frac{17}{18} = \frac{6}{18}$$

$$d_2 = p(x|C_2) \cdot p(C_2) = \frac{1}{2} \cdot \frac{2}{18} = \frac{1}{18}$$

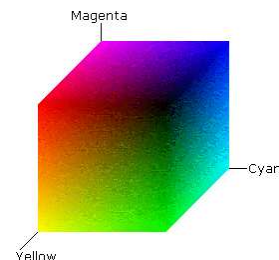
$$d_1 > d_2 \rightarrow x \text{ is assigned to } C_1$$

10 Color Image Processing:

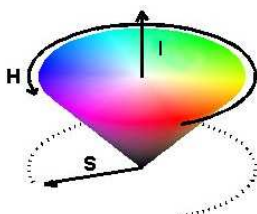
RGB Colour System: Additive colour model:



CMY-Colour System: Subtractive colour model:



HSI-Colour System: (hue, saturation, intensity)



Advantages:

For many applications it would be better to change the colour system (e.g. from rgb to hsi) to get a better separation from the observed objects.