Introduction to Computer Vision

KKY/RVB Lecture Computer Vision

Ing. Petr Neduchal

Department of Cybernetics
Faculty of Applied Sciences
University of West Bohemia

ESF projekt Západočeské univerzity v Plzni
reg. č. CZ.02.2.69/0.0/0.0/16_015/0002287

What is Computer Vision?

Image processing

Lower level. Image processing addresses techniques that process image in some way. e.g. noise filtration, image compression, image sharpening or basic searching for objects.

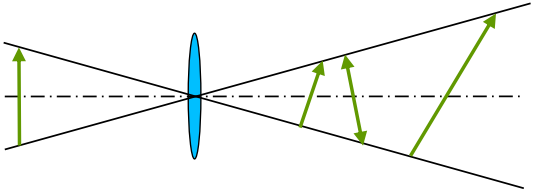
Computer Vision

Computer Vision is the science that focus on techniques that enable computer to understand high level information in digital images (scene and its 3D representation). It uses wide range of techniques such as Feedback, Modeling of the world and objects, methods from Artifical Inteligence, ...

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
DEPARTMENT OF CYBERNETICS

Why is it hard?

- Loss of information due to perspective projection, can be solved by using more than one camera.



- The basic unit – brightness – depends on multiple factors such as camera pose and orientation, light source properties or an object material reflectivity.
- A presence of the noise in the real world data.
- An amount of data especially in the case of video stream.
- Local window from the global view

Recognition vs. Reconstruction

Recognition

- Scene is classified objects into classes.
- Classes are usually known in advance.

Reconstruction

- Searching for the physical parameters of the scene
  - orientation
  - color
  - depth
  - light and surface properties
- Searching for a relations between objects in the scene.

Basic concepts

Image Function

- The result of perspective projection with respect to geometry of the scene.
- Usually denoted as f(x,y) or f(x,y,t)

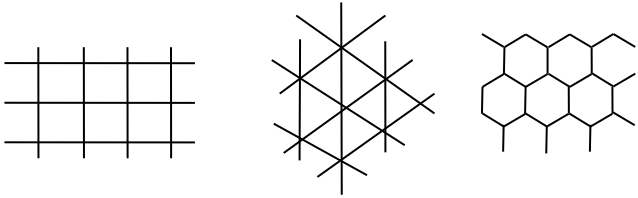
$$f(x, y, t) = \int_0^\infty e(x, y, t) \cdot S(\lambda)\, d\lambda, \tag{1}$$

where x and y are spatial coordinates of the pixel, t is a time.

- The value of the continous image function is a intensity (grayscale image) or color (color image) of the point in the scene.

Image digitization

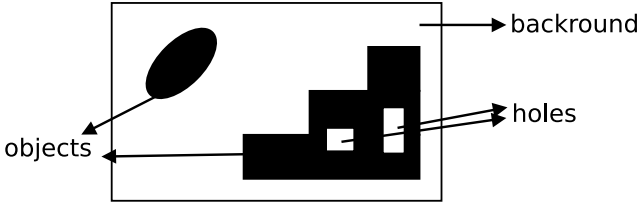
- **Sampling** – image points layout



- R**ectangular grid** can cause problems based on the different distance between points.
- **Sampling theorem** is crucial – unsufficient density of points $\Rightarrow$ aliasing.
- **Quantization** – Results of image function are quantized – usually 8 bits
- **Note** – Human eye is capable to recognize approximately 50 levels of grayscale intensity,

Basic concepts - area, object, background

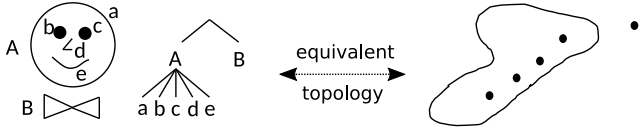
- Image can be divided into areas which belongs either to objects R_i or to the background R_c of the image:



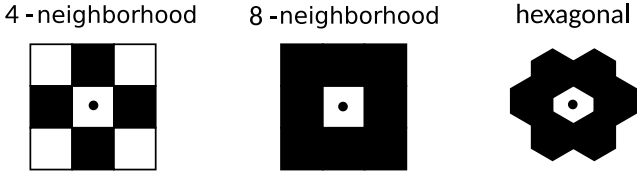
- Area – continuous set of points
- Objects – all objects $R \cup_i R_i$
- Background – R_c – can be composed from multiple parts (holes in objects)

Basic concepts - Topology, neighborhood

- Topology – arrangement of the image parts



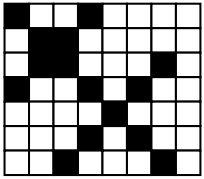
- Neighborhood



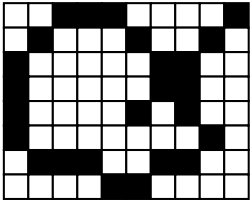
Distance and Paradoxes

- Distance between pixels
  - Euclidean $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
  - City block $| x_1 - x_2 | + | y_1 - y_2 |$
  - Chessboard $max\,(| x_1 - x_2 |, | y_1 - y_2 |)$
- Paradoxes

line paradox



Jordans paradox



Digital image processing

- **input:** image $f(i,j)$
- **output:** usually processed image $g(i,j)$
- Preprocessing method are in general los-making w.r.t. information contained in the input image. $\Rightarrow$ The best preprocessing is **NO** preprocessing.
- **GOAL:** suppress distortion, highlight parts of the image
- types of preprocessing methods based on the neighborhood:
  - brightness transformations
  - local operations
  - geometrix transformations
  - frequency analysis

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
DEPARTMENT OF CYBERNETICS

Image Brightness Characteristics

Histograms

Basic statistic description of the image brightness.

- Histogram:

$$H(p) = \sum_{i,j} h(i,j,p) = \begin{cases} 1 & \text{for } I(i,j) = p \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

  where I is an input image, i and j are spatial coordination of pixel and p is a pixel value.

- Relative histogram:

$$H_R(p) = \frac{H(p)}{\sum_p H(p)} = \frac{H(p)}{i \cdot j}, \quad \sum_p H_R(p) = 1 \tag{3}$$

Image Brightness Characteristics

Histograms

Basic statistic description of the image brightness.

- Cumulative Histogram:

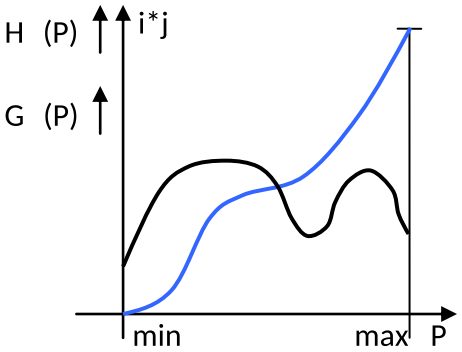
$$G(p) = \sum_{q=min}^{p} H(q) \tag{4}$$

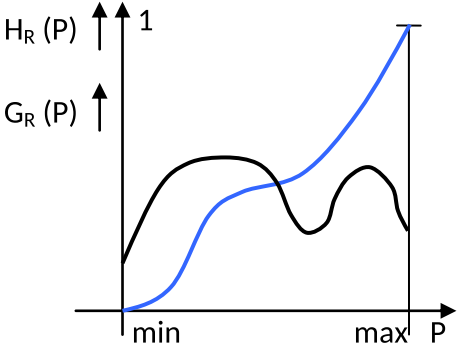
$$G(max) = \sum_{p} H(p) = i \cdot j \tag{5}$$

- Relative cumulative histogram:

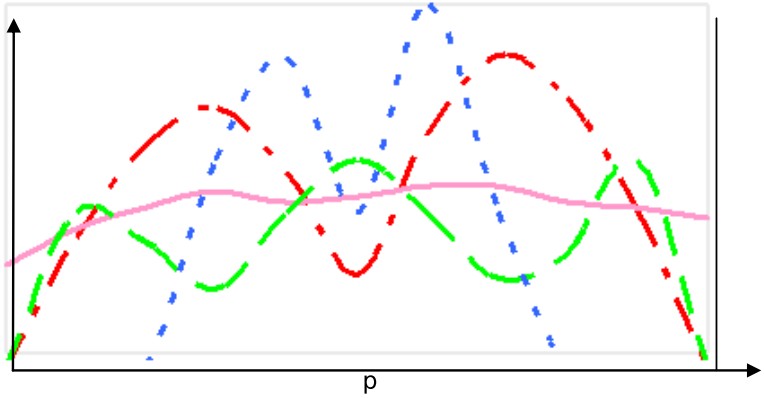
$$G_R(p) = \frac{G(p)}{\sum_p G(p)} = \frac{G(p)}{i \cdot j}, \quad \sum_p G_R(p) = 1 \tag{6}$$

Histograms





Histogram examples



Coocurence matrix

Definition

$$S = [s_{pq}] \qquad s_{pq} = \sum_{i,j} g(i,j,p,q) \tag{7}$$

$$g(i,j,p,q) = \begin{cases} I0 & f(i,j) \neq p \\ 0 & f(i,j) = p \wedge \nexists [k,l] \in O_8 \text{ where } f(k,l) = q \\ num\ of\ points & f(i,j) = p \wedge \forall [k,l] \in O_8 \text{ where } f(k,l) = q \end{cases} \tag{8}$$

where $O_8(i,j)$ is 8-neighborhood of pixel i,j

Coocurence matrix

Properties

- matrix element - expresses how many times the brightness p is neighbor to the brightness q
- is symetric
- elements on the diagonal - the brightness is neighbor to itself - a measure of the size of continuous areas
- the sum of the elements in a given row outside the diagonal - the measurement of angularity

Brightness transformations

Brightness corrections

- the new point brightness is a function of position and brightness

$$g(i,j) = FUNC(i,j,f(i,j)) \tag{9}$$

- most often: $g(i,j) = FUNC(i,j) \cdot corr(i,j)$ where corr is a matrix of correction coefficients.
- use: correction of systematic errors of the digitization process
- approach:
  - Calibration - we scan an image with known values on a scanning device. From these known correct values and from the measured values, we calculate a matrix of correction coefficients. Multiply each image by this matrix

$$corr(i,j) = \frac{correct(i,j)}{measured(i,j)} \tag{10}$$

Brightness transformations

Brightness transformations

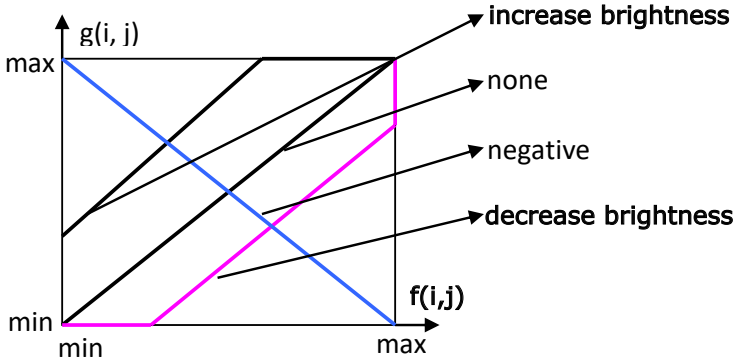
- function the same for all pixels

$$g(i,j) = FUNC(f(i,j)) \tag{11}$$

- FUNC does not depend on i, j
- LookUp Table

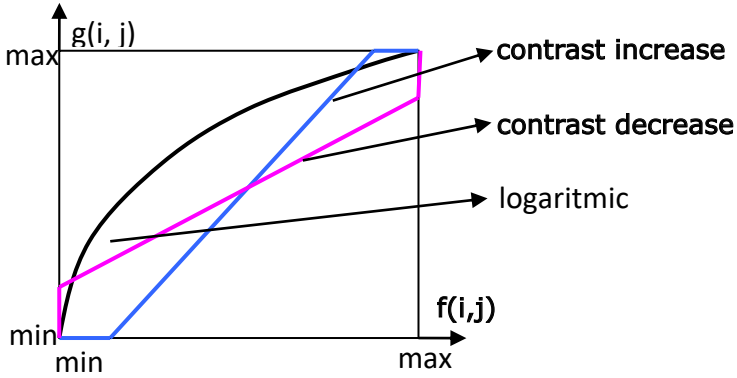
Brightness transformations

Brightness transformations - Brightness



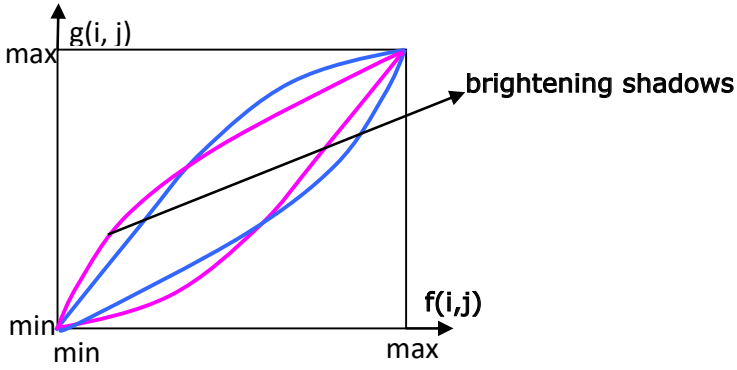
Brightness transformations

Brightness transformations - Contrast



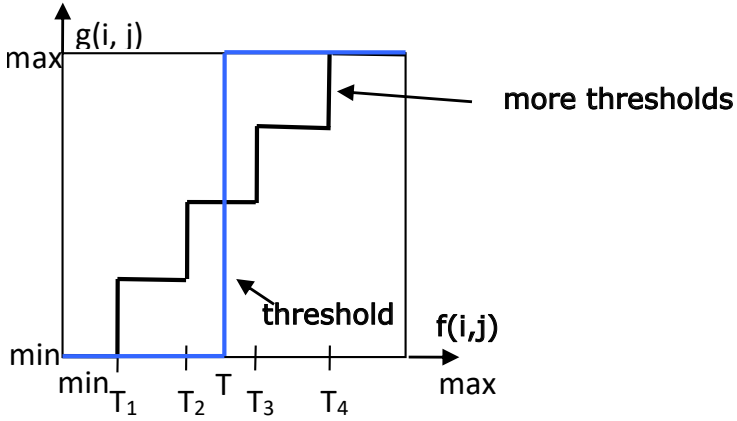
Brightness transformations

Brightness transformations - Shadows



Brightness transformations

Brightness transformations - Thresholding



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

DEPARTMENT OF CYBERNETICS

Brightness transformations

Brightness transformations - Histogram equalization

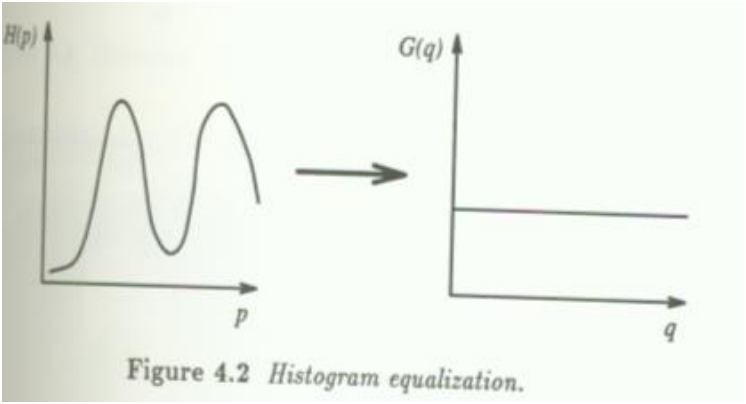


Figure 4.2 *Histogram equalization.*

Brightness transformations

Brightness transformations – Histogram equalization – Example

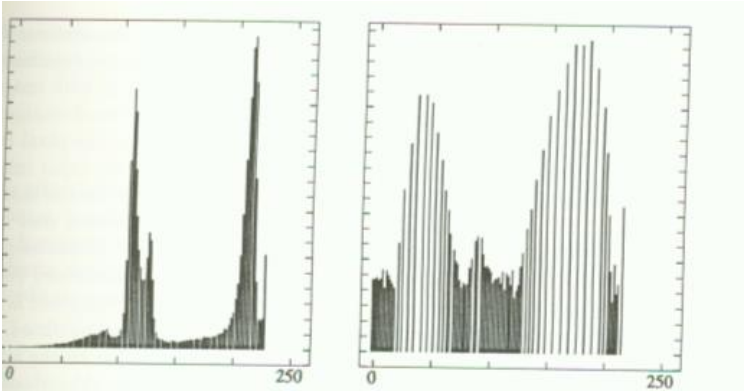


Figure 4.4 *Histogram equalization: Original and equalized histograms.*

Geometric Transformations

- Transformation T_G

$$i' = u(i,j) \quad j' = v(i,j) \Rightarrow g(i',j') = f(i,j) \tag{12}$$

- transformation relationship is known - e.g. rotation, translation, resize,...
- the relationship can be found based on the original and transformed image
- e.g. correspondence of coordinates on a satellite image and on a map (so-called fitting points are used)

Geometric Transformations

Spatial transformation

$$x' = \sum_{r=0}^{m} \sum_{k=0}^{m-r} a_{rk} x^r y^k \qquad y' = \sum_{r=0}^{m} \sum_{k=0}^{m-r} b_{rk} x^r y^k \tag{13}$$

polynomial of the mth degree

billinear

$$x^i = a_0 + a_1 x + a_2 y + a_3 xy \tag{14}$$

$$y^i = b_0 + b_1 x + b_2 y + b_3 xy \tag{15}$$

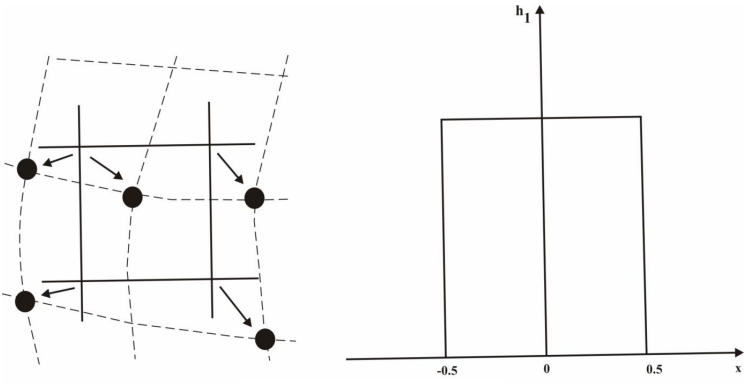
affine – rotation, translation, resize, ...

$$x^i = a_0 + a_1 x + a_2 y \tag{16}$$

$$y^i = b_0 + b_1 x + b_2 y \tag{17}$$

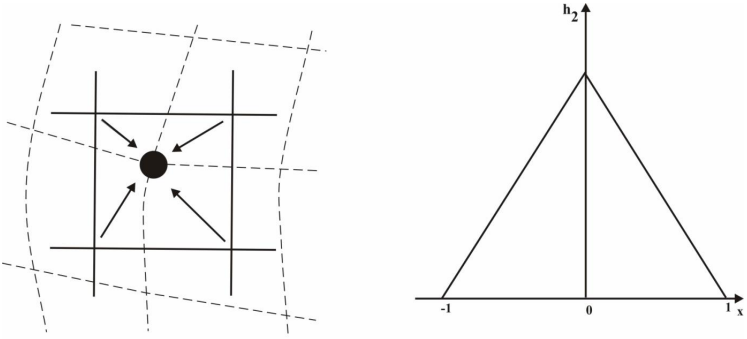
Geometric Transformations

Brightness interpolation – Nearest Neighbor usually the inverse transformation is performed and the nearest value is subtracted in the original image.



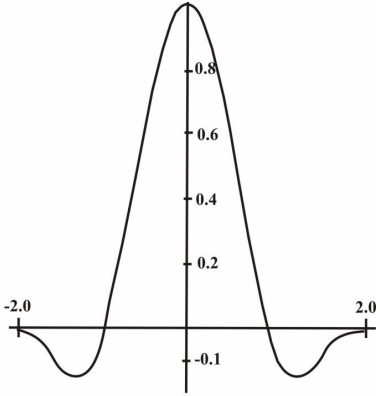
Geometric Transformations

Brightness interpolation – Billinear



Geometric Transformations

Brightness interpolation – Bicubic



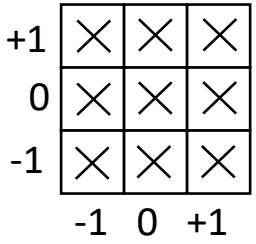
EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
DEPARTMENT OF CYBERNETICS

Local preprocessing

Discrete convolution

$$g(i,j) = \sum_{(m,n)\in o} \sum f(j-m, j-n) \cdot h(m,n) \tag{18}$$

where h is a mask



Local preprocessing

Smoothing

- **Goal:** noise reduction
- global smoothing trough more images $\Rightarrow$ averaging over more images

$$g(i,j) = \frac{1}{n}\sum_{k=1}^{n} f_k(j,j) \tag{19}$$

where f_k is the image function of k-th image. Image number n is usually 30-50.

- local smoothing – edges are blurred, details are lost.
  - The size of the mask should be smaller than the smallest detail in the image we want to keep.

Local preprocessing

Local smoothing masks

even mask

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

center point advantage

$$h = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

advantage of the center point and the main axes

$$h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

disadvantage of the center point

$$h = \frac{1}{8}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
DEPARTMENT OF CYBERNETICS

Local preprocessing

Maximum incidence smoothing - example

$$f(i,j) = \begin{bmatrix} 22 & 31 & 31 \\ 22 & 25 & 31 \\ 27 & 30 & 36 \end{bmatrix}, \quad g(i,j) = 31 \tag{20}$$

Problems:

$$f(i,j) = \begin{bmatrix} 21 & 22 & 23 \\ 24 & 25 & 26 \\ 27 & 28 & 29 \end{bmatrix} \qquad f(i,j) = \begin{bmatrix} 8 & 19 & 26 \\ 8 & 18 & 26 \\ 8 & 19 & 26 \end{bmatrix} \tag{21}$$

Local preprocessing

Quantile sellection - median smoothing - example

$$f(i,j) = \begin{bmatrix} 100 & 90 & 85 \\ 93 & 99 & 110 \\ 154 & 86 & 79 \end{bmatrix}, \quad g(i,j) = 31 \tag{22}$$

Sorted:

$$79 \;\; 85 \;\; 86 \;\; 90 \;\; [93] \;\; 99 \;\; 100 \;\; 110 \;\; 154 \Rightarrow g(i,j) = 93 \tag{23}$$

- solves the problem of outlier/s – i.e., biased values
- is nonlinear
- breaks thin lines and corners

Local preprocessing

Gradient operators

- gray level discontinuity detection in the image
- can be used for segmentation
- requirements: size and orientation of the gradient
- Gradient (continuous)

$$| \, grad(g) \, | = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} \tag{24}$$

$$\varphi = arctg\left(\frac{\partial g}{\partial y} / \frac{\partial g}{\partial x}\right) \tag{25}$$

Local preprocessing

Gradient operators

- Gradient (discrete)

$$\Delta_x g(i,j) = g(i,j) - g(i,j-1) \qquad \Delta_y g(i,j) = g(i,j) - g(i-1,j) \tag{26}$$

$$| \, grad(g) \, | = \sqrt{(\Delta_x g)^2 + (\Delta_y g)^2} \tag{27}$$

$$\varphi = arctg \, (\Delta_y g / \Delta_x g) \tag{28}$$

- three types of gradient operators
  - approximation of derivatives by differences (1st and 2nd order)
  - comparison with parametric edge model
  - zero crossings of 2. derivation of image function (Marr's theory of edge detection)

Local preprocessing

Gradient operators – approximation of derivatives by differences

- Roberts

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{29}$$

$$g(i,j) = | f(i,j) - f(i+1,j+1) | + | f(i,j+1) - f(i+1,j) | \tag{30}$$

- Laplace

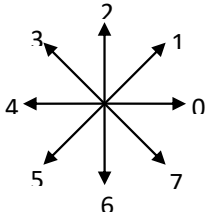
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{31}$$

- It only specifies the size of the edge, not its direction. If we also want to know the direction of the edge, we use directionally dependent gradient operator.

Local preprocessing

Gradient operators – comparison with parametric edge model

- orientations:



$$|\, grad(g) \,| \widehat{=} max_{k=0\ldots7}(g * h_k) \tag{32}$$

$$\varphi \widehat{=} k^* = argmax_{k=0\ldots7}(g * h_k) \tag{33}$$

Local preprocessing

Gradient operators – comparison with parametric edge model

- Prewitt

$$h_0 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad h_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad h_3 = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$h_4 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad h_5 = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad h_6 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad h_7 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

Local preprocessing

Gradient operators – comparison with parametric edge model

- Sobel

$$h_0 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h_1 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \quad h_2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad h_3 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

$$h_4 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad h_5 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad h_6 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_7 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

Local preprocessing

Gradient operators – comparison with parametric edge model

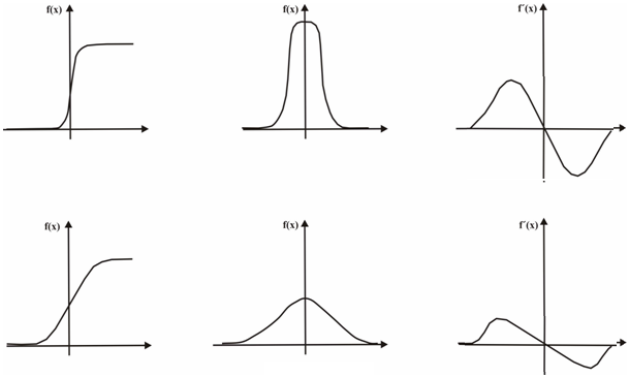
- Kirsch

$$h_0 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \quad h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{bmatrix} \quad h_2 = \begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix} \quad h_3 = \begin{bmatrix} 3 & -5 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & 3 \end{bmatrix}$$

$$h_4 = \begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix} \quad h_5 = \begin{bmatrix} -5 & -5 & 3 \\ -5 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix} \quad h_6 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \quad h_7 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix}$$

Local preprocessing

Gradient operators – Marr's theory of edge detection



zero crossings of 2. derivation of image function (1D case)

Gradient operators – Marr's theory of edge detection

- edge detection mask

$$h_0 = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad h_7 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad h_3 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

- point detection mask

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Segmentation

INPUT: INTENSITY IMAGE

OUTPUT: IMAGE DIVIDED INTO AREAS RELATED TO REAL WORLD OBJECTS

Complete segmentation:

- distinguishable areas related to real world objects.
- necessary to use knowledge about solved problem.
- special case: constant objects in front of constant background of known brightness – good results without any further knowledge of the problem.

Examples: text in scanned document, blood cells, counting screws

Partial segmentation:

- areas are homogeneous with respect to particular selected properties (brightness, color, texture, ...)
- areas can overlay each other
- necessary to use knowledge about solved problem.

Examples: scene with field and forest viewed from window – areas can not be related to real world objects

Segmentation - Task knowledge

It is necessary to use knowledge about the problem to obtain the best results. For example:

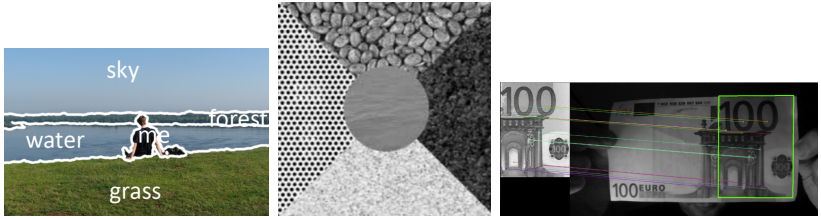
- defined shape
- defined position and orientation
- defined start and end point of the object edge
- relation between object and other areas

Examples:

- *searching for ships on the sea (example property: color of background)*
- *searching for railways and highways in the map (example property: maximum curvature)*
- *searching for rivers in the map (example property: rivers does not intersect)*

Segmentation - Segmentation approaches

- brightness approaches – thresholding
- edges based approaches
- areas based approaches



Segmentation - Thresholding

- ☺ oldest and simplest approach
- ☺ the most common approach
- ☺ low resource and computing requirements
- ☺ fastest approach – capable to run in real-time
- ☹ threshold determination – not a simple task to perform it automatically
- ☹ can be perform only on particular type of input images (objects and background are easy to distinguish)

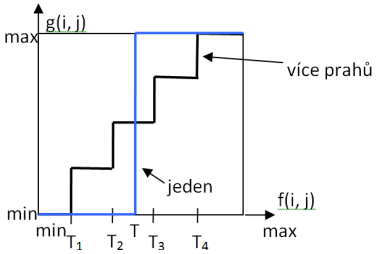
$$g(i,j) = \begin{cases} 1 & \text{for } f(i,j) \geq T \\ 0 & \text{for } f(i,j) < T \end{cases} \tag{34}$$

T – threshold constant

Segmentation - Thresholding

$$g(i,j) = \begin{cases} 1 & \text{pro } f(i,j) \geq T \\ 0 & \text{pro } f(i,j) < T \end{cases} \tag{35}$$

T – threshold constant



Segmentation - Thresholding

Modification: thresholding with a set of known brightness constants

$$g(i,j) = \begin{cases} 0 & \text{for } f(i,j) \in D \\ 1 & \text{otherwise} \end{cases} \tag{36}$$

where D is a set of brightness values representing background.

Examples: blood cells – cytoplasm has particular set of brightness (background is brighter, core is darker)

Modification: thresholding with multiple threshold

$$g(i,j) = \begin{cases} 1 & \text{for } f(i,j) \in D_1 \\ 2 & \text{for } f(i,j) \in D_2 \\ \vdots & \\ n & \text{for } f(i,j) \in D_n \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

where $D_i \cap D_j = 0 \; i \neq j$

Segmentation - Thresholding

Modification: partial thresholding

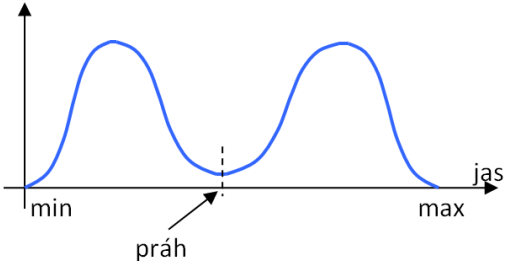
$$g(i,j) = \begin{cases} f(i,j) & \text{for } f(i,j) \in D \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

where D is a set of brightness values related to e.g multiple objects.

- remove background
- preserve brightness values in objects
- $f(i,j)$ **not only** brightness function (e.g. image gradient value, texture property, depth map, color)

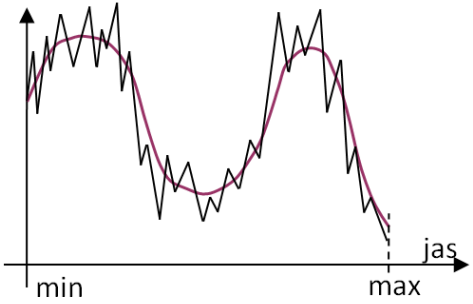
Determination of the thershold

- basic thesholding is based on known (in advance) thershold
- input information for threshold determination is usually image histogram
- how to determine the threshold **automatically**?; "trial and error"approach or;
- bimodal histogram (2 sufficiently distant maximums)
- in this case is possible to determine minimum value between masimums.



Histogram smoothing:

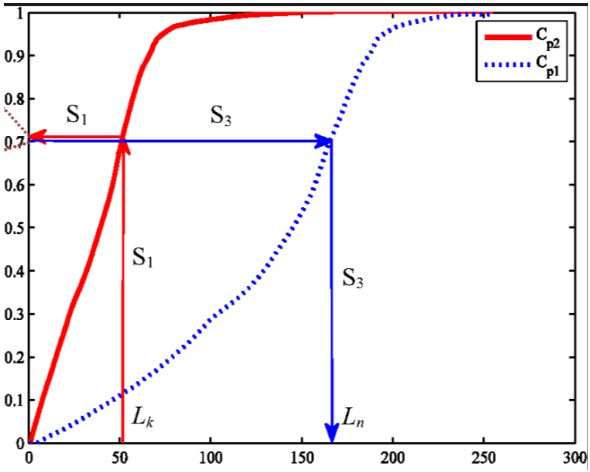
- we are looking for a local minimum between the two largest sufficiently distant local maxima
- but often it is not possible to decide unambiguously about the significance of local maxima and minima
- smoothing suppresses local extremes and ideally provides a bimodal histogram (local averaging - e.g. Gaussian window or median filtering, etc.)



Percentage thresholding

- we have an a priori knowledge of what percentage of the image area is covered by objects.
- e.g. the average text coverage of the page area is around 5 %
- we set the threshold so that just as many percent of the pixels have the color of the objects, the rest the background color.
- see fig. on the next slide: cumulative histogram for 2x differently lit scene, object covers 70 %

Percentage thresholding



Adaptive thresholding

- one global threshold value may not be appropriate for certain cases
- the image may have different lighting conditions in different places
- in this case the adaptive thresholding calculates the threshold for small regions of the image (sliding window)

Thank you for your attention!

Questions?