University of Applied Sciences
Department of
Design, Computer Science, Media
Prof. Dr. D. Richter

Transilvania University Brasov
Faculty of
Mathematics and Informatics
Dr. I. Plajer

ISIP 2019
International School on Image Processing
July 15th - 26th, 2019

detlef.richter@HS-RM.de

University of Applied Sciences
Department of
Design, Computer Science, Media
Prof. Dr. D. Richter

Lectures on Image Processing

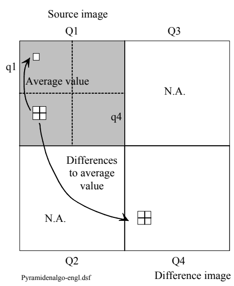
Chapter 8A
Selected Algorithms and Applications

Einige Bildbeispiele gem. UrhG §52a nicht enthalten

Notice to Copyright

8.1 Selected Algorithms and Applications

8.2 Selected Algorithms and Applications, Pyramid Algorithm

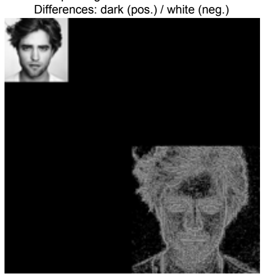


- Define 512² array
- Use 256² source image Q1, differential image in Q4, Q2 und Q3 unused
- Calculate mean value for 4 pixel
- Store mean value in q1 of Q1
- Store differences in homologue positions in Q4
- Iteration with q1 and q4 in Q1
- Procedure terminates when mean value of source image is in pixel (0,0).
- Reconstruct source image after transmission until desired resolution.

8.2 Selected Algorithms and Applications, Pyramid Algorithm

Source Image 512 x 512, Array 1024 x 1024

1. Step: Image 512 x 512 → 256 x 256
Differences: dark (pos.) / white (neg.)

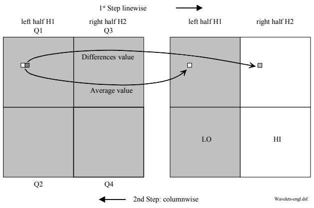


8.3 Selected Algorithms, example for Pyramid Algorithm

Resolution from 1^2 to 256^2

Rasterung.jpg

8.3 Selected Algorithms and Applications, Wavelet Algorithm

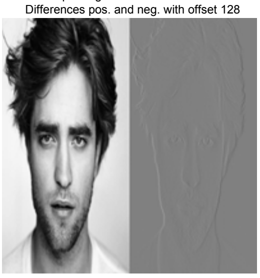


- Define two 512² arrays, write 512² source image into left array.
- Calculate mean value of 2 adjacent pixels in a line, store mean value in left half (LO) of 2nd array and store the difference value in right half (HI) of 2nd array.
- Repeat procedure column wise from right array and store results in upper and lower half of left array.

8.3 Selected Algorithms and Applications, Wavelet Algorithm

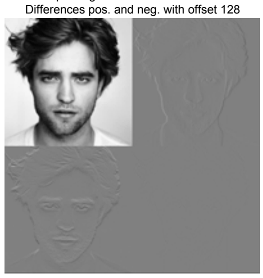
Source Image 512 x 512

1. Step: Image 512 x 512 → 512 x 256
Differences pos. and neg. with offset 128

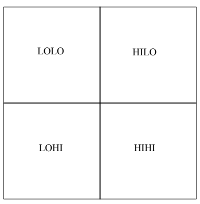


8.3 Selected Algorithms and Applications, Wavelet Algorithm

2. Step: Image 512 x 256 → 256 x 256
Differences pos. and neg. with offset 128

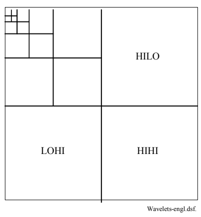


8.3 Selected Algorithms and Applications, Wavelet Algorithm



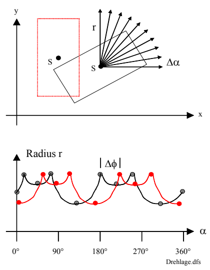
- Result:
- 4 quadrants LOLO, LOHI, HILO, HIHI in left array.
- LOLO contains a frequency reduced image.

8.3 Selected Algorithms and Applications, Wavelet Algorithm



- Iteration of the procedure out of LOLO-image in Q1 of left array.
- Procedure terminates when mean value of source image is in pixel (0,0).
- Reconstruct source image after transmission until desired resolution.

8.4 Selected Algorithms and Applications, Angular Orientation

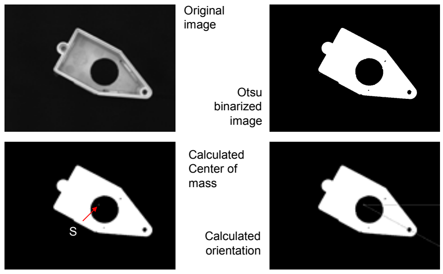


Teach-in Phase :
- Calculate center of binary object S in defined position (red).
- Determine lengths of radial vectors in steps of Δα.
- Store function $r_1 = f_1(\alpha)$ in an appropriate array.

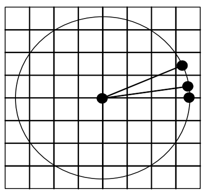
Working Phase :
- Calculate S (black).
- Determine lengths of radial vectors in steps of Δα, get $r_2 = f_2(\alpha+\Delta\Phi)$.
- Shift function f_2 against f_1 until difference of f_1 and f_2 is minimum to define ΔΦ.

Independency of object position

8.4 Application of Angular Orientation for Robotics



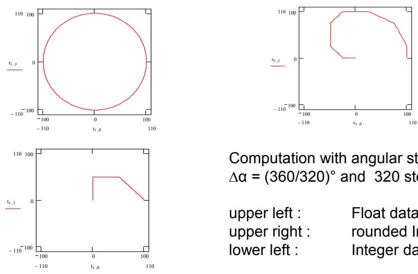
8.5 Selected Algorithms and Applications, Error Propagation



Rotationsmatrix.dsf

- Calculate circle by applying a rotational matrix R in arbitrary steps of Δα.
- Plot a circle line by repeated application of the matrix on last point.
- Use different data types (integers without rounding (truncation), integers with correct rounding, float or double)
- Choose correct center of image for the circle, e.g. M = (383.5, 287.5)
- choose correct radius to plot the line, e.g. R = xxx.5

8.5 Selected Algorithms and Applications, Error Propagation



Computation with angular steps of Δα = (360/320)° and 320 steps

upper left :	Float data
upper right :	rounded Integer data
lower left :	Integer data

Result dependent on resolution of Δα

8.7 Selected Algorithms and Applications, Connect Component Labelling

P_3	P_2	P_1
P_4	P(x,y)	

CCL.dsf

Separation of disjoint objects in binary images. (Lit. : Rosenfeld A., Kak A.C., Digital Picture Processing, Vol. 2, Academic Press, New York, 1982)

1st step : scanning binary image from upper left to lower right by given mask for white pixel. For white pixel do:
- None of the pixels P_1 to P_4 is labelled : then : $P(x,y) \leftarrow$ new label $L_i = \min\{$set of not assigned labels $L_i\}$ (Start condition)
- At least one of the pixels P_1 to P_4 is labelled with label L_m : then $P(x,y) \leftarrow L_p = \min\{L_m\}$. In case of two different labels L_m and L_n are occurring : Note it within a list of equivalencies.

8.7 Selected Algorithms and Applications, Connect Component Labelling

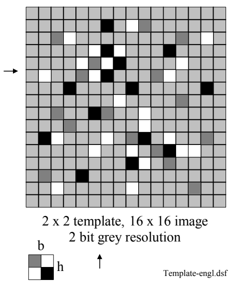
P_3	P_2	P_1
P_4	P(x,y)	

CCL.dsf

2nd step : replace equivalent labels in the list of equivalencies by their smallest equivalent.

Visualize all individual objects by bounding boxes or by different grey levels.

8.8 Selected Algorithms and Applications, Template Matching

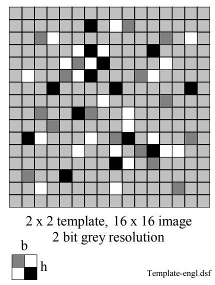


- Searching of defined patterns by templates
- Simple algorithm even for complex patterns
- Scanning of image using a template T = t(i,j) of size b x h
- b and h odd,

$$-\frac{b-1}{2} \le i \le +\frac{b-1}{2}$$

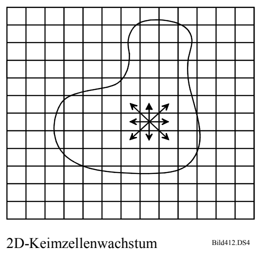
$$-\frac{h-1}{2} \le j \le +\frac{h-1}{2}$$

8.8 Selected Algorithms and Applications, Template Matching



- Calculation of similarity measure d between image and template : $d(x,y)=\sum\sum(g(x+i,y+j)-t(i,j))^2$
- If $d(x,y) \Rightarrow$ minimum and $d(x,y) \le T$, then template pattern matches image
- Thresholding is necessary, because there exists always a minimum

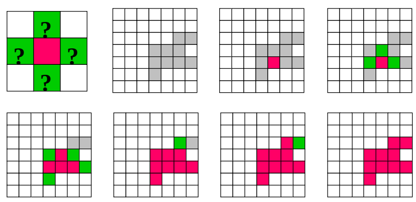
8.9 Selected Algorithms and Applications, Seed-Algorithm



- Search for connected structures or tissues, defined by special properties.
- Thresholding mostly not applicable (e.g. medical images).
- 2 voxel belong to the same object, if there exists a chain of connected voxels with same properties (homogeneity criterion).
- Normally check two adjacent voxels.
- Problems :
  concave structures (2D, 3D);
  partial effects (area, volume) : pixel or voxel belongs to different objects.

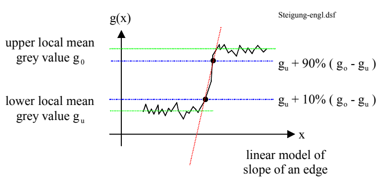
8.9 Selected Algorithms and Applications, Seed-Algorithm

4-neighborhood in case of 2-dim image



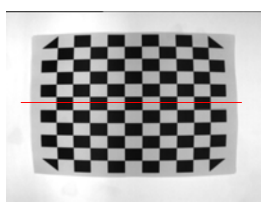
Add pixel to requested structure according homogeneity criterion

8.10 Selected Algorithms and Applications, Grey level slope at edges



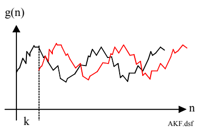
- Suppose to have a bimodal histogram (Chapter 2) in full image or in region of interest. Determine mean upper g_o and lower g_u grey level.
- Define a linear grey level transition function between 10 % and 90 %
- Calculate a subpixel precise position of grey values : $g(x)<g_{calculated}<g(x+1)$
- Determine the slope of linear grey level transition function (sharpness)
- Define the edge at 50 % range.

8.10 Selected Algorithms and Applications, Grey level slope at edges



- On-line measurement of grey level slope assists optimal focusing for automatic camera calibration either with manually operated or computer operated objectives.

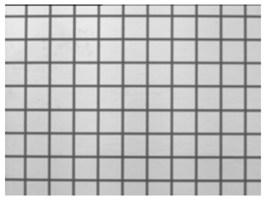
8.11 Selected Algorithms and Applications, Autocorrelation funct.

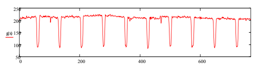


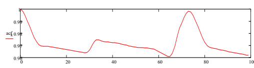
- Suppose to have the given cut out of an image line :
- g(n), n=1, …, N, (black curve)
- Calculate $R(0) = \sum_{n=1}^{n=N} g(n) * g(n)$
- Shift g(n) for k pixel (red curve)

  $R(k) = \sum_{n=0}^{n=N} g(n) * g(n+k)$
- Normalize R(k) : $r(k) = \frac{R(k)}{R(0)}$
- In case of an existing periodicity with period k r(k) shows a local maximum in the approximate size of 1.

8.11 Selected Algorithms and Applications, Autocorrelation funct.

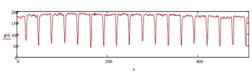

Original Image (576 x 786)

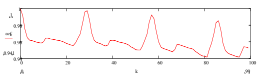

g(x, y=270)


r(k) = R(k)/R(0) for k = 0 … 99
max{r(k = 73)} = 0.998…

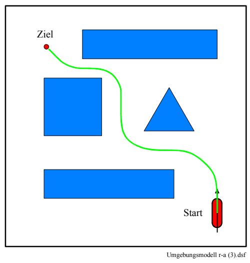
8.11 Selected Algorithms and Applications, Autocorrelation funct.


Original Image (512 x 512)


g(x, y=40)


r(k) = R(k)/R(0) for k = 0 … 99
max1{r(k = 29)} = 0.998…
max2{r(k = 57)} = 0.993…
max3{r(k = 86)} = 0.986…

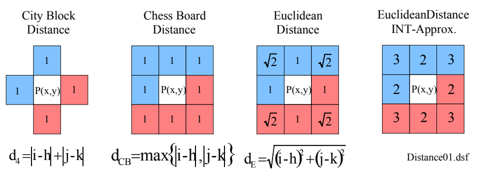
8.14 Mobile Roboter, Distanzkarten, Distance Maps



To find a path for an autonomous moving robot from starting point to destination it is necessary to find the most distant points from any obstacles in an environment. These points are calculated by distance maps. To calculate distance maps different masks are used,e.g.

- City-Block-mask,
- Chess-Board-mask,
- Euclidean mask.

8.14 Mobile Roboter, Distanzkarten, Distance Maps



$d_4 = |i-h| + |j-k|$ $\quad d_{CB} = \max\{|i-h|, |j-k|\}$ $\quad d_E = \sqrt{(i-h)^2 + (j-k)^2}$

Distance d between any two points: $P_1 = (i, j)$ and $P_2 = (h, k)$

8.14 Mobile Roboter, Distanzkarten, Distance Maps

Define:

White objects or obstacles (g=255) → preset distance map with 0 (distance from object = 0)

Black background means no object (g=0) → preset distance map with 1 (distance not yet calculated)

8.14 Mobile Roboter, Distanzkarten, Algorithmus

Example: Euclidean Distance

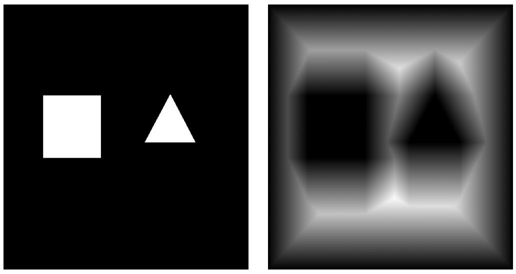
Forward calculation (blue mask elements) from top left to bottom right:
$d(x,y) = \min\{\text{mask elements} + \text{value at pixel position}\}$ if $d(x,y) = 1$
e.g.
$d(x,y) = \min\{\sqrt{2} + d(x-1,y-1),\ 1 + d(x,y-1),\ \sqrt{2} + d(x+1,y-1),\ 1 + d(x-1,y)\}$

Backward calculation (red mask elements) from bottom right to top left:
$d(x,y) = \min\{\text{mask elements} + \text{pixel position}\}$ if $d(x,y) \neq 0$
e.g.
$d(x,y) = \min\{d(x,y),\ 1 + d(x+1,y),\ \sqrt{2} + d(x+1,y+1),\ 1 + d(x,y1),\ \sqrt{2} + d(x-1,y+1)\}$

City Block- und Chess Board Distance are calculated accordingly.

Hochschule RheinMain - University of Applied Sciences

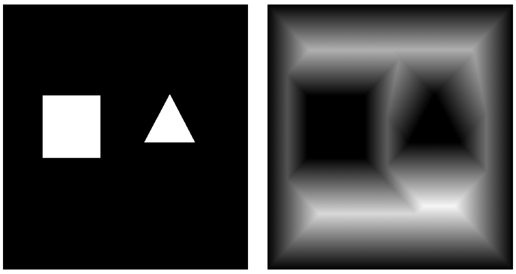
8.14 Mobile Robots, Distance Maps, City-Block-Distance



Prof. Dr. D. Richter Department [Design>Computer Science>Media]

Hochschule RheinMain - University of Applied Sciences

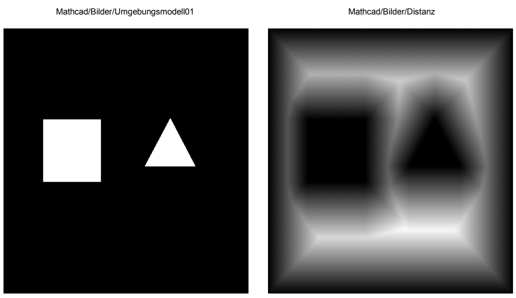
8.14 Mobile Robots, Distance Maps, Chess-Board-Distance



Prof. Dr. D. Richter Department [Design>Computer Science>Media]

Hochschule RheinMain - University of Applied Sciences

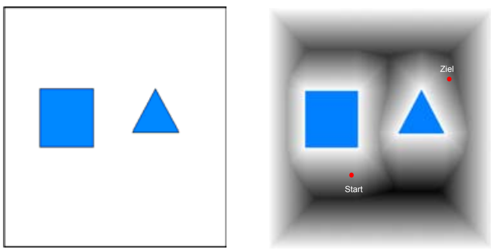
8.14 Mobile Robots, Distance Maps, Euclidean-Distance



Prof. Dr. D. Richter Department [Design>Computer Science>Media]

Hochschule RheinMain - University of Applied Sciences

8.14 Mobile Robots, Distance Maps, Path-Finding

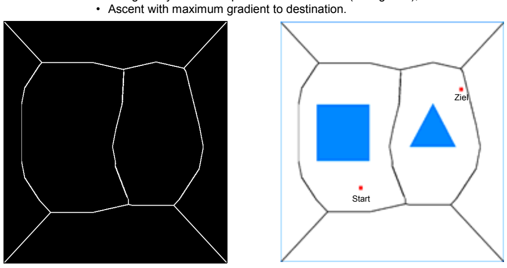


Prof. Dr. D. Richter Department [Design>Computer Science>Media]

Hochschule RheinMain - University of Applied Sciences

8.14 Mobile Robots, Distance Maps, Path-Finding

- Decent from start with maximum gradient down to valley,
- Along valley to nearest position to destination (ambiguous),
- Ascent with maximum gradient to destination.



Prof. Dr. D. Richter Department [Design>Computer Science>Media]

8.16 Object Tracking

Insert Special Chapter
Kalman Filter

Review of Chapter 8

- Automatic binarization according to Otsu
- Pyramid and the wavelet algorithm for adaptive image size
- Calculation of the angular position of objects in binary images
- Error propagation due to wrong data formats
- Tracing of isophotes in grey level images

Review of Chapter 8 (continued)

- Labeling of disjoint objects in binary images by Connect Component Labeling
- Template Matching for search of complex patterns in images
- Segmentation by seed algorithm in medical images
- Automatic focus adjustment by analysis of grey level slopes at edges
- Search for periodic structures by autocorrelation function

Review of Chapter 8 (continued)

- Performance optimizing by real time measurement of algorithms
- Producing one-pixel wide edges in grey level images by Canny filter
- Distance maps for robot path planning