Lecture 22:
Segmentation I:
Classical Methods

Contents

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Introduction

What is Segmentation?

- Partition of the image domain Ω into connected regions $X_1,\ldots,X_n$.
- In the ideal case, every region X_i represents an object in the real world.
- The notion of segmentation is not uniform in the literature:
  Sometimes already edge detection is regarded as segmentation, even if the contours are not closed.
- One of the most difficult areas in image analysis:
  illumination differences, occlusions, texture, lack of a priori knowledge
- No method works well in all situations.

Introduction (2)

Many books distinguish two different general strategies:

- **Region Based Segmentation**
  - Pixels that belong to the same segment have similar grey values.
  - Global decision that is often fairly stable (due to integration), but may give unpleasant boundaries.

- **Edge Based Segmentation**
  - There is a jump in the grey values between two adjacent regions.
  - Local decision that may be more unstable (due to differentiation), but can give nicer boundaries.
  - Example: Zero crossings of the Laplacian yield an edge based segmentation with closed contours as segment boundaries (Lecture 18).

This distinction is somewhat artificial and focuses only on the extreme cases.
Thus, we do not consider it further.

Introduction (3)

What About Textures ?

- Segmenting textures requires a preprocessing step:
  - apply one or multiple texture descriptors, e.g. central moments averaged within some neighbourhood (cf. Lecture 21)
  - creates an image with one or multiple feature channels that are almost homogeneous results within one segment

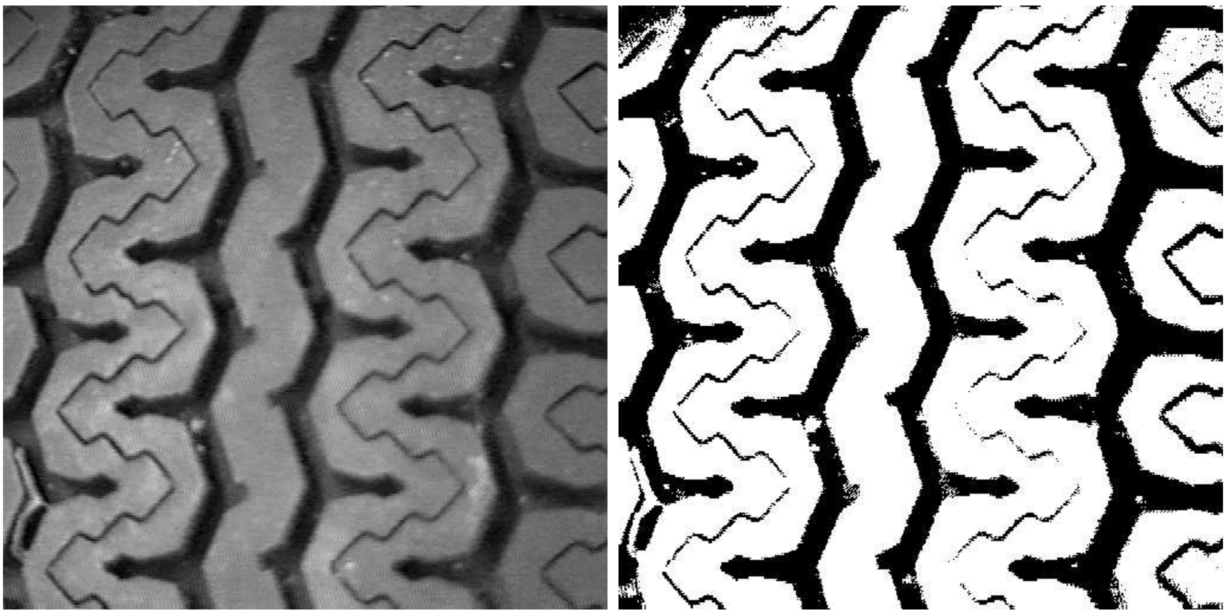
- Afterwards some standard segmentation method for scalar- or vector-valued images can be applied.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Thresholding (Schwellwertbildung, Binarisierung)

- simplest segmentation method
- creates a binary image:
  Grey values at one side of the threshold are assigned to one class,
  while the ones on the other side belong to the other.
- Problems:
  – ignores spatial context
  – selection of the threshold parameter

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27



(a) Left: Original image with 256 greyscales. **(b) Right:** Binarised at $T = 87$. Author: J. Weickert (2000).

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Thresholding (3)

Quantile Method for Selecting the Threshold

- The p *quantile* of a histogram is the grey value below which one has the fraction p of all pixels.
  Example: 70 % of all grey values are smaller than the 70 % quantile.
- Application:
  For optical character recognition one knows typical quantiles that are useful for separating characters from the background. This allows to select the threshold parameter automatically.

Mode Method for Selecting the Threshold

- For a bimodal histogram (two peaks) one chooses the threshold parameter of the minimum between the two peaks.
- also possible for more peaks
- problematic when noise is present;
  reqires smoothing then

Thresholding (4)

Double Thresholding

- used e.g. in the Canny edge detector (hysteresis thresholding, Lecture 18).
- A larger threshold T_2 yields "seeds" of the segmentation, which are accepted in all cases.
- They grow in all directions where grey values can be found that exceed some smaller threshold T_1.
- method does take into account spatial context
- often better than a single threshold

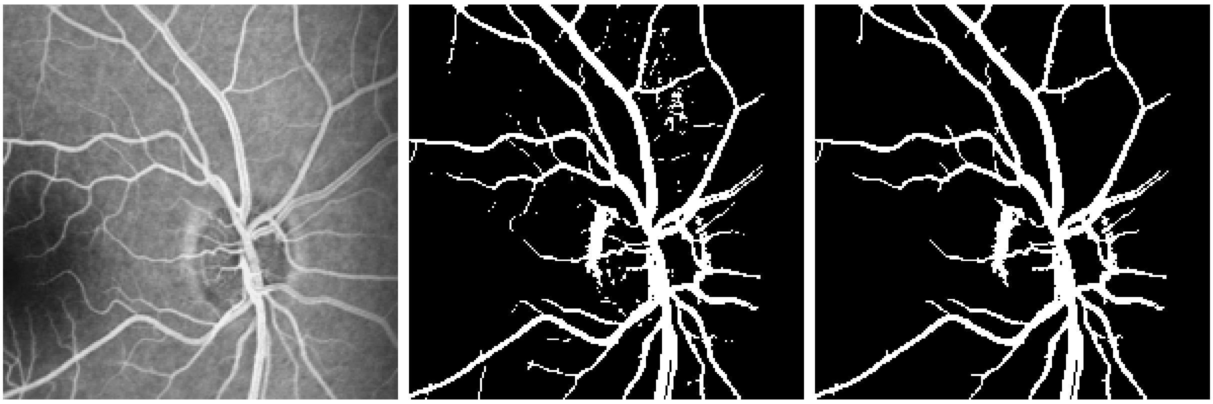


Illustration of double thresholding. **(a) Left:** Original image, 200×200 pixels. **(b) Middle:** Binarisation with a single threshold $T = 160$. **(c) Right:** Double thresholding with $T_1 = 160$ and $T_2 = 210$. Author: J. Weickert (2006).

Region Growing (Regionenwachstum)

Basic Idea

- Start with a manually marked seed within the structure to be segmented.
- Let this seed grow as long as the neighbouring grey values differ by less then some specified threshold T.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Region Growing (2)

Advantages

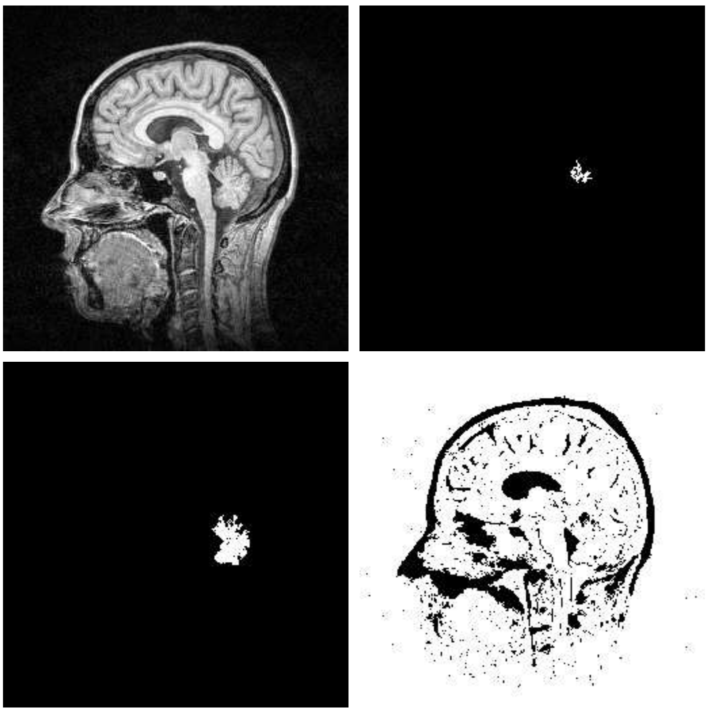
- interactive: loved by clinicians
- indepedent of the selection of the seed within one segment
- uses spatial context and homogeneity

Disadvantages

- sensitive w.r.t. noise
- homogenity is only requested locally, not within two pixels of the entire segment
- consequence: unsharp boundaries lead to rapidely growing segments

Region Growing (3)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27



(a) Top left: Original image. A point within the cerebellum is chosen as seed. **(b) Top right:** Region growing with difference $T = 8$. **(c) Bottom left:** $T = 16$. The cerebellum is well segmented. **(d) Bottom right:** $T = 17$. The segmentation is of no use. Author: J. Weickert (2000).

Region Merging (Regionenverschmelzen)

Basic Idea

- merge all pixels, where the neighbours differ by less than T

Properties:

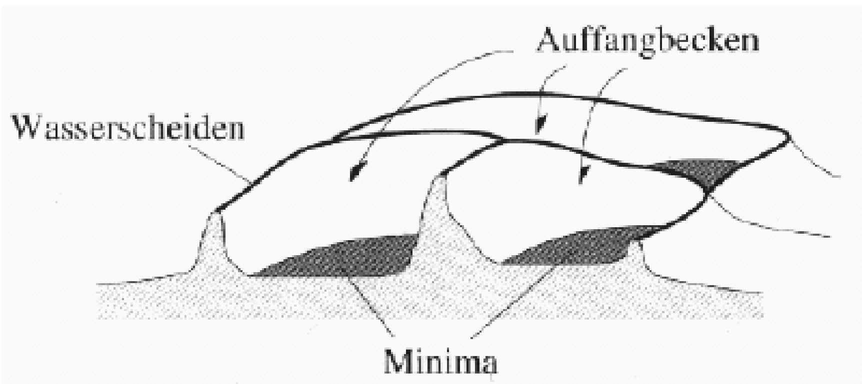
- hierarchical segmentation:
  increasing T merges more and more segments
- same disadvantages as region growing

Modification:

- introduce homogeneity criterion within one segment:
  e.g. the deviation from the mean must be below some threshold
- method becomes more complicated in this case

MIA 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

The Watershed Transformation (Wasserscheidentransformation)



Watersheds, bassins and minima of an image in topographic representation. Author: P. Soille (1998).

The Watershed Transformation (2)

Basic Idea

- morphological segmentation method
- extracts edges as watersheds in the gradient magnitude image $|\nabla f|$
- Consider $|\nabla f|$ in a topographic representation (landscape with hills and valleys).
- A water drop moves downwards to the valley, while choosing the direction of steepest descend.
- Regions where the rain flows to the same valley belong to the same bassin (segment).
- boundary between two bassins: watersheds

The Watershed Transformation (3)

A toboggan is highly efficient for sliding downhill. This idea is also useful for image segmentation. Source: `http://www.mkweb.co.uk/Xscape/images/toboggan.jpg`.

MIA
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

A Toboggan Watershed Algorithm

- Track a toboggan on the image $g = |\nabla f|$ on its way down to the valley.
- Put all points along its path on a stack.
- Once the toboggan reaches the minimum in (p, q) of the *gradient* image $g = |\nabla f|$, one assigns the grey value $f(p, q)$ of the *original (!)* image f to all pixels on the stack.
- Creates a piecewise constant image f:
  Every region attains the grey value correponding to its extremum or saddle point. This image enhancement procedure is also called *toboggan contrast enhancement*.
- Watersheds are given by the discontinuities in the final image.
- What about efficiency?
  - Points of the stack are marked and treated only once.
  - If the toboggan reaches a pixel that has been treated already, there is no need to follow its path downhill any further.
  - Thus the algorithm is highly efficient: linear complexity in the pixel number

(a) Top left: Original image. **(b) Top right:** Gradient magnitude, with gamma correction ($\gamma = 3$) for better visibility. **(c) Bottom left:** Toboggan contrast enhancement creates piecewise constant patches. **(d) Bottom right:** Watersheds as boundaries of these patches. Author: J. Weickert (2006).

The Watershed Transformation (6)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Main Drawback of the Watershed Transformation

- Oversegmentation:
  numerous small bassins due to noise, image quantisation and irrelevant local minima
- Also saddle points of f lead to an oversegmentation with the toboggan watershed algorithm.

Remedy

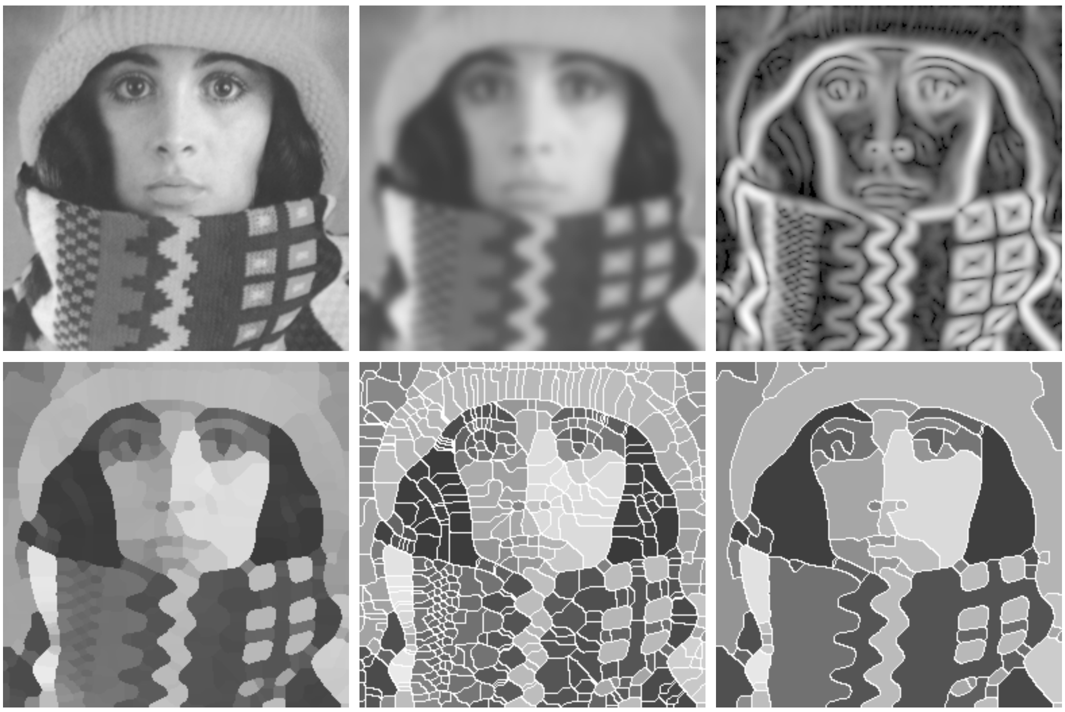
- preprocessing, e.g. by Gaussian convolution
- postprocessing, e.g. by region merging

However:

- Both steps introduce additional parameters.
- Often it is not sufficient to apply only one of these steps.
- Gaussian convolution may also delocalise structures.
  This can be addressed by downfocussing in scale-space.

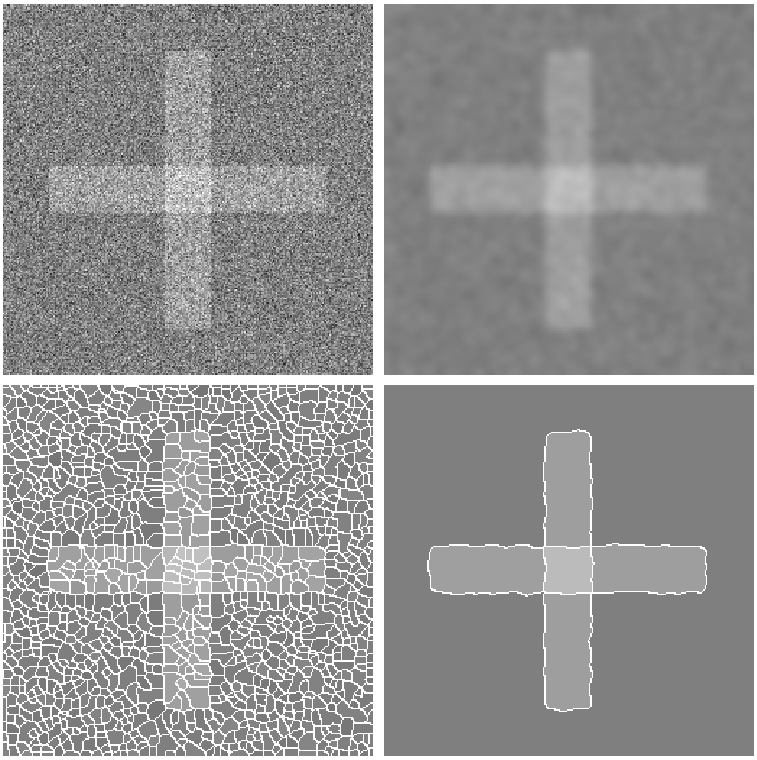
The Watershed Transformation (7)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27



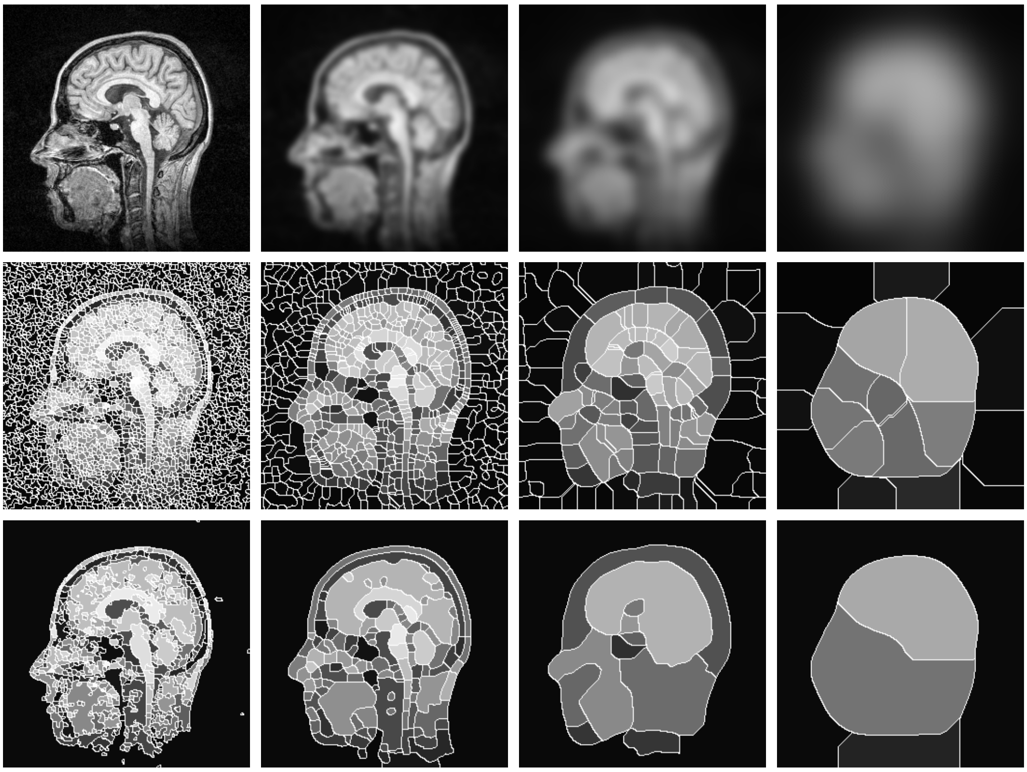
(a) Top left: Original image, 256×256 pixels. **(b) Top middle:** Preprocessing with Gaussian convolution ($\sigma = 3$). **(c) Top right:** Gradient magnitude of (b), with gamma correction ($\gamma = 3$). **(d) Bottom left:** Toboggan contrast enhancement. **(e) Bottom middle:** Watersheds of (d). **(f) Bottom right:** After region merging with threshold $T = 12$. Author: J. Weickert (2006).

The Watershed Transformation (8)



(a) Top left: Noisy test image, 256×256 pixels. **(b) Top right:** Preprocessing by Gaussian convolution with $\sigma = 3$. **(c) Bottom left:** Watershed segmentation of (b). **(d) Bottom right:** Postprocessing by region merging with threshold $T = 10$. Author: J. Weickert (2006).

The Watershed Transformation (9)



Top row: Preprocessing a 256×256 image using Gaussian convolution with $\sigma = 0,\ 3,\ 7$ and 19. **Middle row:** Watershed segmentation. **Bottom row:** Postprocessing by region merging with thresholds $T = 12,\ 12,\ 25$ and 20. Author: J. Weickert (2006).

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Summary

- Segmentation belongs to the most difficult problems in image analysis.
- Thresholding is the simplest segmentation method, but it does not take into account the spatial context.
- Double thresholding may give improved performance and incorporates spatial context.
- Region growing and region merging also exploit the spatial context, but in their simpler implementation only local homogeneity criteria are used.
- The watershed transform is a morphological edge based segmentation method that usually suffers from oversegmentation.
  To get useful results, some pre- and postprocessing is recommendable.

Literature

- K. D. Tönnies: *Grundlagen der Bildverarbeitung*. Pearson, München, 2005.
  (Chapter 8 given an introduction to simple segmentation approaches.)
- R. Jain, R. Kasturi, B. G. Schunck: *Machine Vision*, Mc Graw-Hill, New York, 1995.
  (for thresholding, region growing, region merging)
- P. Soille: *Morphological Image Analysis*. Springer, Berlin 1999.
  (for watershed segmentation)
- J. Fairfield: Toboggan contrast enhancement for contrast segmentation. *Proc. Tenth International Conference on Pattern Recognition* (Atlantic City, June 16–21, 1990), IEEE Computer Society Press, Los Alamitos, Vol. 1, pp. 712–716, 1990.
  (toboggan watershed algorithm)
- L. Vincent, P. Soille: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 583–598, 1991.
  (popular alternative to the toboggan watershed algorithm)
- J. B. T. M. Roerdink, A. Meijster: The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, Vol. 41, pp. 187–228, 2001.
  `http://www.cs.rug.nl/~roe/publications/parwshed.pdf`
  (survey with numerous watershed algorithms)

Assignment P6 – Programming Work

Please download the required files from the webpage

http://www.mia.uni-saarland.de/Teaching/ipcv07.shtml

into your own directory. You can unpack them with the command `tar xvzf Ex06.tgz`.

Problem 1 (Corner Detection) (4+4+3)

(a) The routine `struct_tensor` within `corners_sub.c` is supposed to compute the structure tensor. Supplement the missing parts in this routine. Please observe: The grey values of the image `u[i][j]` have the index range `i=1,...,nx` and `j=1,...,ny`. With the routine `dummies` it is possible to create dummy boundary values by mirroring.

(b) Try to find a good criterion for detecting corners from the structure tensor, and implement it in the routine `cornerness`. To compile the final programme, use

```
gcc -O2 -o corner corner.o corner_sub.c -lm .
```

(c) Evaluate your corner detector with the images `square.pgm`, `stairs.pgm`, and `acros.pgm`. Try to optimise your result by tuning the noise scale and the integration scale.

M I A
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27

Problem 2 (Double Thresholding) (4+2+3)

(a) The object code `doublethresh.o` contains the main programme for image segmentation by double thresholding. What is missing is a subroutine for double thresholding. To address this problem, supplement `dt.c` with the missing code. It is useful to call the subroutine `dilation (ualt, nx, ny, u)` from the main programme. It computes in every pixel of the image `ualt` of size `nx * ny` the maximum within a 3×3 mask and assigns it to the image `u`. The final programme can be compiled with

```
gcc -O2 -o doublethresh doublethresh.o dt.c -lm .
```

(b) Evaluate your programme with the image `gauss.pgm`. For finding suitable thresholds, it can be helpful to manipulate the intensity in the colour editor of `xv`.

(c) The image `wood.pgm` contains a black (!) defect in the centre. Try to isolate it with a single threshold in `xv`, without getting undesired artifacts from other objects. What do you have to do before you can apply the double thresholding procedure? Try to find good parameters.

M I A
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27

Assignment P6 (3)

Submission

Please remember that up to three people from the same tutorial group can work and submit their results together. For submitting the files rename the main directory `Ex06` to `Ex06_<your_name>` and use the command

```
tar czvf Ex06_<your_name>.tgz Ex06_<your_name>
```

to pack the data. The directory that you pack and submit should at least contain the following files:

- the source code for computing the structure tensor and the cornerness in `corners_sub.c`
- the source code for performing double thresholding in `dt.c`
- the corner maps for the images `square.pgm`, `stairs.pgm` and `acros.pgm`
- the thresholded images for `gauss.pgm` and `wood.pgm`
- a text file README that
  - states the parameters for the corner detection and double thresholding
  - answers the question of task 2 (c),
  - contains information on all people working together for this assignment.

Please make sure that only your final version of the programmes and images are included. Submit the file via e-mail to your tutor via the address:

```
ipcv-xx@mia.uni-saarland.de
```

where `xx` is either `t1`, `t2`, `t3`, `t4`, `w1` or `w2` depending on your tutorial group.

Deadline for submission: Tuesday, January 29, 10 am (before the lecture)