NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Computer and Information Science

Examination paper for TDT4265 Computer Vision

Academic contact during examination: Christian Schellewald
Phone: 94123453

Examination date: 21.5.2013
Examination time: 15:00-19:00
Permitted examination support material:

No written or printed materials allowed.
No calculator allowed.

Other information:

Time allowed: 4 hours = 240 minutes.

The point scores for each question part are shown in the text. Use these point scores to help you budget your time and effort in answering.

READ EACH QUESTION THROUGH CAREFULLY BEFORE BEGINNING YOUR ANSWER.
ANSWER ALL QUESTIONS. SHOW ALL YOUR WORKING.

Total point score: 70
Number of questions: 22

Language: English
Number of pages: 8
Number of pages enclosed: 8

Checked by:

____
Date Signature

1 Theme: Image Processing Fundamentals

1.1 Filtering/Convolution

1.1.1 Provide a binomial approximation of a 1D Gaussian filter with 4 elements.

(a) [2 points] Give the filter-mask along with their correct normalization factor.

(b) [1 point] What is the disadvantage of such an even sized filter?

1.1.2 [2 points] Padding: What is meant by zero-padding in image processing and which disadvantages do you see in this padding method?

1.1.3 [2 points] Use symmetric padding for the 1d array and convolve it with the given filter mask.

	3	2	4	4	4	2	3	3	

filter:			
	−1	0	1

1.1.4 [2 points] Name one non-linear filter that can help to reduce salt and pepper noise. Why is this a nonlinear filter?

1.1.5 Consider the following 5x5-filter:

$$\frac{1}{50}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

(a) [2 points] What are the two main effects on an image after a single filtering with this filter?

(b) [1 point] What result image do you expect to see if you apply the filter over and over again?

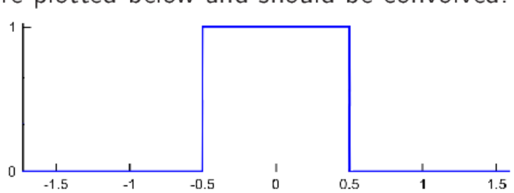
(c) [1 point] Make an estimate how often you can apply this filter until the result does not change any longer?

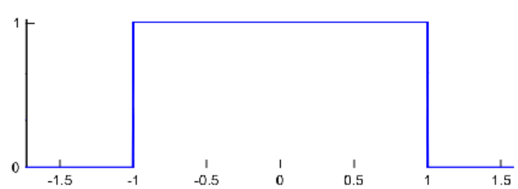
(d) [2 points] Can you think of ways that can be used to speed up the convolution? What is the computational effort of this faster convolution on a NxN gray-value image? And what would be the computational effort of a regular, non-optimized convolution?

1.1.6 The two rectangular functions $f(x)$ and $g(x)$

$$f(x) = \begin{cases} 0 & \text{if } |x| > 0.5 \\ 1 & \text{if } |x| \leq 0.5 \end{cases} \qquad g(x) = \begin{cases} 0 & \text{if } |x| > 1.0 \\ 1 & \text{if } |x| \leq 1.0 \end{cases}$$

are plotted below and should be convolved.





(a) [2 points] Draw a sketch of the function $h(x) = (f * g)(x)$ which is the result of the convolution of f and g. There is no need to compute the convolution explicitly but indicate the length of function-segments in you sketch.

(b) [2 points] Is this the same as the convolution $(g * f)(x)$? Is that allways the case?

1.2 Smoothing and Edge-detection

1.2.1 Assume that you have recorded 100 noisy images of exactly the same static scene (there is no change of the camera position, too). Each image has a noise level with standard deviation σ.

(a) [1 point] What can you do to obtain an image with a reduced noise level ?

(b) [1 point] About what factor will the noise be reduced?

1.3 Histograms

1.3.1 The two images shown in figure 1 below have both the size of 512x512 pixels. A uniform bright or dark colored square in the left image has the size of 256x256 pixels while a uniform colored square in the right image has a size of 128x128 pixels.

Figure 1: Two chess patterns

(a) [1 point] Can you distinguish the histograms of the shown images? Explain!

(b) [1 point] Does that change when you look at the histograms of smoothed versions (you can assume smoothing with a 3x3 box-filter) of the images ? Explain why!

2 Theme: Filtering in the Frequency Domain

2.1 Fourier transform

2.1.1 [4 points] Consider the 1D rectangular function shown below.

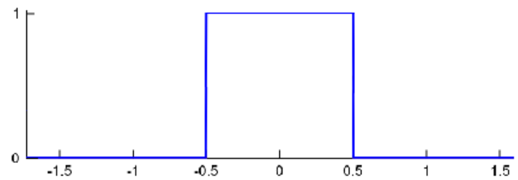


Figure 2: A 1-dimensional rectangular function

The function is defined by the equation below:

$$f(x) = \begin{cases} 0 & \text{if } |x| > 0.5 \\ 1 & \text{if } |x| \leq 0.5 \end{cases}$$

Calculate the continuous 1D Fourier transform of this rectangular function.
Reminder: The 1D Fourier transform is defined by the integral:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} \, \mathrm{d}x \tag{1}$$

You might although find Euler's formula useful:

$$e^{ix} = \cos x + i \sin x \tag{2}$$

2.1.2 [2 points] If you compute the Fourier transform of the two scaled versions of a pattern shown in figure 3 can you distinguish the amplitude images of their Fourier transform $|\mathcal{F}\{f\}|$?

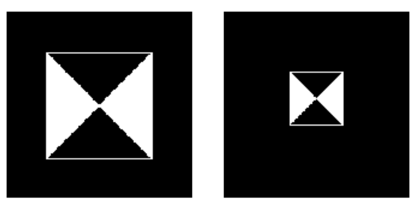


Figure 3: Scaled pattern

2.1.3 Assume that the 1D Fast Fourier transform (FFT) for an 1D-array with n elements has exactly the computational effort of $n \log_2(n)$.

(a) [2 points] Describe how one can perform the 2D Fast Fourier transform of 2D arrays (images) with the help of the 1D Fast Fourier transform when one knows that the Fourier transform is separable.

(b) [2 points] With this knowledge make a good estimate for the computational effort of a 2D Fast Fourier transform of a gray-value image with size MxN.

3 Theme: Morphological Image Processing

3.1 Set Operations

3.1.1 Mathematical morphology is concerned with operations on sets of pixels/points. The rectangular set A and the L-shaped set B are arranged as shown in the center of figure 4.

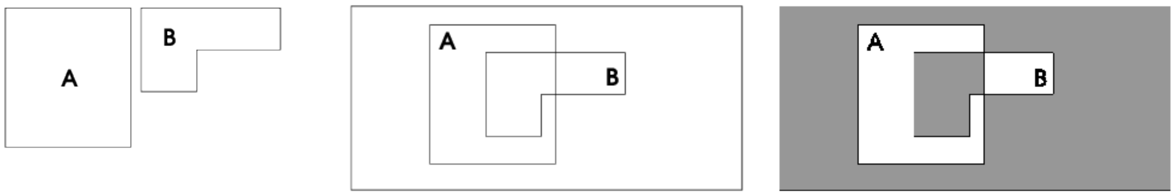


Figure 4: Arrangement of the sets A and B. Find an expression that defines the gray-shaded areas shown on the right side.

(a) [3 points] Using basic set-operations like intersection ($\cap$), union ($\cup$) or the complement (A^C) find a mathematical expression that defines the gray-shaded areas shown on the right side of figure 4.

3.2 Basic Operations

3.2.1 The *hit-and-miss transform* can be used to locate specific patterns of background- and foreground pixels in an image. It does so by employing a structuring element containing both 1's (indicating a foreground pixel), 0's (indicating a background pixel) and "don't care" values. Only positions matching both the foreground and background pattern of the structuring element are kept after the transform.

Below are four structuring elements shown (rotated versions of each other) that can be used for finding certain structures in an image. The reference pixel is located in the center of the structuring elements and a pixel is kept (=marked) if at least one of the four masks fits at a considered position.

0	1	0
0	0	0

0	0	
0	1	
0	0	

0	0	0
0	1	0

	0	0
	1	0
	0	0

Figure 5: Four structuring elements identifying four different rotations of the same structure

(a) [2 points] What type of structure are these strucuring elements able to identify?

(b) [1 point] Is the described operation invariant against rotations of the image about 90 degrees? Explain!

3.3 Skeletonization

3.3.1 When discussing skeletons here, we mean the *medial axis*, defined as the set of points of an object having more than one closest points on the object's boundary. Skeletonization reduces a figure to a set of thin lines. Under certain conditions, these lines can be used to reconstruct the original figure.

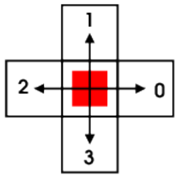
(a) [1 point] What additional information (beside the lines) is needed to form the original figure from its skeleton?

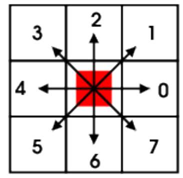
(b) [2 points] If you have this information, how would you go about to reconstruct the figure?

4 Theme: Image Structure and Shape Description

4.1 Chain-Codes

4.1.1 Assume that the chain-codes for a 4 and a 8 neighborhood system are defined as follows:





You are given the following **4 neighborhood** chain-code:

0 1 0 3 0 3 2 3 2 1 2 1

(a) [3 points] Determine the **8 neighborhood** chain-code describing the same shape. (Start the description at the same pixel element.)

4.1.2 Fourier boundary descriptors can be used to describe the boundary of a shape using Fourier methods. By removing different components of the descriptor, we can control what type of information about the shape we keep, and what type of information we disregard.

(a) [1 point] What is the effect of dropping high-frequent components of the Fourier descriptor?

(b) [1 point] Why can it be useful to drop $a(0)$, the most low-frequent component of the descriptor?

5 Theme: Medium/Higher Level Computer Vision Tasks

5.1 Classification

5.1.1 k Nearest Neighbor (kNN) classification is a way to classify new samples based on previously observed (and learned) instances.

(a) [2 points] Assume you wanted to use a kNN classifier to classify shapes as circular or rectangular. Briefly describe what you would need to do to train and use this classifier.

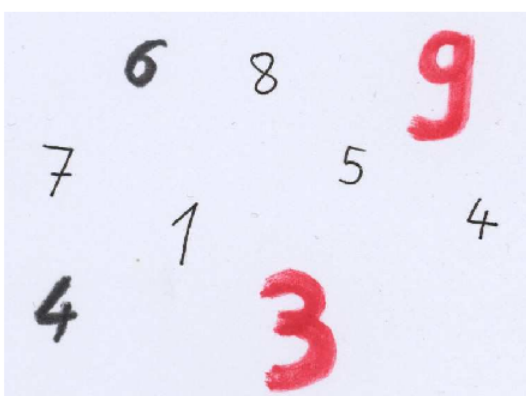
(b) [2 points] The value of k will greatly affect how new instances are classified. What is the effect of varying k on the performance of the classifier? Tip: Describe what happens to the *decision boundary* for large and small values of k.

5.2 Higher Level Tasks

5.2.1 [4 points] Describe roughly the steps you would use to determine the location of vanishing points in images like the one shown below:

5.2.2 [4 points] Briefly explain how edge detection by second-derivative zero-crossing is performed in images. Explain shortly the function of the filters you use.

5.2.3 [4 points] Describe roughly which basic image processes and representations might be used to recognize handwritten digits $0, 1, ..., 9$ like in the example image below:



6 Theme: Graph-Based Methods in Computer Vision

6.1 Graphs

6.1.1 Five regions in an image are segmented and labeled by numbers $1, .., 4$ as indicated. Create a region adjacency graph

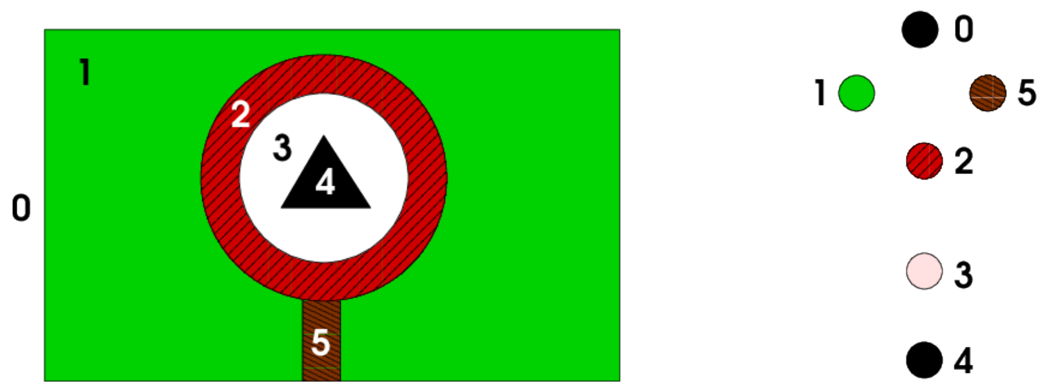


Figure 6: Segmented and labeled image

(a) [2 points] Describe a general method to create a region adjacency graph.

(b) [1 point] Draw the region adjacency graph and use the above indicated node-setup.

6.2 Dynamic Programming

6.2.1 Dynamic programming helps to efficiently solve some problems which are difficult and time-consuming when solved by brute-force methods. One of these problems is finding a cheapest path through a weighted graph with connected layers like the one shown in figure 7. There the weights (=costs) to move from one node to another are indicated by the integer-weights above the edges.

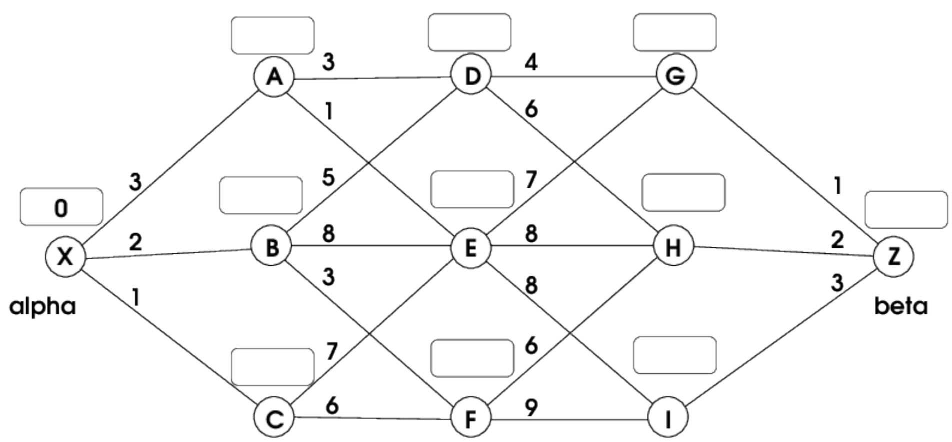


Figure 7: Find the cheapest path through the graph.

(a) [3 points] Determine for each node in the graph of figure 7 the cheapest cost to reach that node from the source node X (=alpha).

(b) [1 point] Which is the cheapest way to go from X (=alpha) to Z (=beta)?

——————— Congratulations this is the last page of the exam ! ———————