TDT4195: Visual Computing Fundamentals

Digital Image Processing - Assignment 3

September 12, 2016

Aleksander Rognhaugen
Department of Computer and Information Science
Norwegian University of Science and Technology (NTNU)

- **Delivery deadline: September 30, 2016** by 22:00.
- **This assignment counts towards 4 % of your final grade.**
- You can work on your own or in groups of two people.
- Deliver your solution on *itslearning* before the deadline.
- Please upload your report as a PDF file, and package your code into an archive (e.g. zip, rar, tar).
- The programming tasks may be completed in the programming language of your choice, however, it might be a good idea to select one that supports matrix and image processing operations, e.g. MATLAB or Python with NumPy. *The lab computers at IT-S 015 support Python and MATLAB.*
- Your code is part of your delivery, so please make sure that your code is well-documented and as readable as possible.
- For each programming task you need to give a brief explanation of what you did, answer any questions in the task text, and show any results, e.g. images, in the report.
- In this assignment, you can use an existing implementation of morphological operations. For example, for the morphological *closing* operation you can use the `imclose()` function from MATLAB or the `scipy.ndimage.morphology.binary_closing()` function from SciPy (Python).

Learning Objectives: Gain experience with (a) how an image can be segmented into a set of disjoint regions, and (b) how mathematical morphology can be used to manipulate the contents of images.

1 Theory [0.5 points]

1. [**0.1 points**] What are the main aims of segmentation in computer vision? Give two reasons for why segmentation is difficult.

2. [**0.1 points**] Describe how the region-based segmentation algorithm *split and merge* works. Segment the 8×8 binary image in Figure 1 (a) using the split and merge algorithm and show the resulting quadtree. Let the homogeneity criteria $H(R_i)$ of a region R_i be *true* if all pixels in R_i have the same intensity value.

3. [**0.1 points**] Are morphological operations linear? Explain your reasoning.

4. [**0.1 points**] Define *opening* and *closing* in terms of *erosion* and *dilation*. What happens when *opening* and *closing* are applied multiple times on the same image?

5. [**0.1 points**] Determine the *dilation* $A \oplus B$ of the 6×6 binary image in Figure 1 (b). Use the 3×1 structuring element next to the image. The reference pixel is indicated by a circle.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1
0	0	1	0	0	0	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1

(a)

0	0	0	0	0	0
0	1	0	0	1	0
0	1	1	1	1	0
0	0	1	0	1	0
0	0	1	0	0	0
0	0	0	0	0	0

1
●
1

(b)

Figure 1: Two binary images and a structuring element. The foreground is coloured white and given the symbol 1. The background has the symbol 0 and is coloured black. The reference pixel in the structuring element (b) is indicated by a black circle.

2 Programming [3 points]

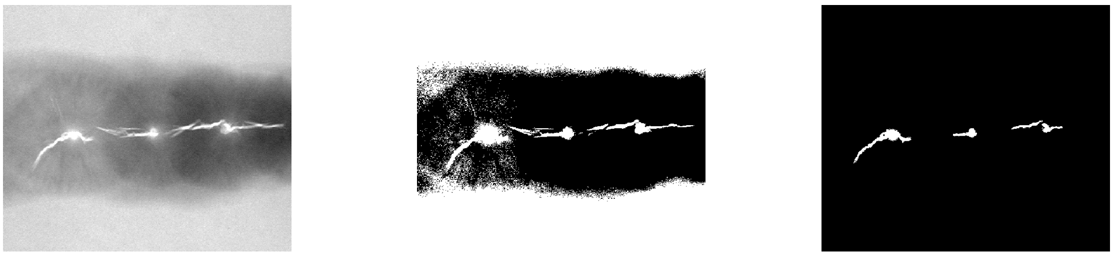
Task 1: Segmentation [1.5 points]

Selecting a good value for thresholding is a cumbersome process that is best left up to a threshold detection algorithm. One iterative algorithm for selecting a good threshold in a greyscale image by Gonzalez et al. [1] can be seen in Algorithm 1. Given an initial threshold estimate, the algorithm computes the mean values of the pixel intensities below and above the estimated threshold, respectively. The two means are averaged and set to be the new threshold estimate. This operation continues until the difference between two consecutive estimates are smaller than ΔT. Naturally, the algorithm works best when the histogram of the original image is roughly bimodal. The result of segmenting the image in Figure 2 (a) using this thresholding algorithm can be seen in Figure 2 (b).

a) **[0.5 points]** Implement a function that takes a greyscale image and calculates a threshold using the global thresholding algorithm outlined in Algorithm 1. Segment an image using the threshold and show the result in your report. What initial threshold estimate and ΔT did you select and why?

Algorithm 1: Basic global thresholding algorithm.

Input : Image $\mathbf{X}$, inital threshold $\mathbf{T}_{initial}$, and ΔT
Output : Threshold $\mathbf{T}$
$\mathbf{T}_{previous} \leftarrow 0$
$\mathbf{T} \leftarrow \mathbf{T}_{initial}$
while $|\mathbf{T} - \mathbf{T}_{previous}| > \Delta T$ **do**
    $S_1 \leftarrow$ set of pixels where $\mathbf{X} > \mathbf{T}$
    $S_2 \leftarrow$ set of pixels where $\mathbf{X} \leq \mathbf{T}$
    $\mu_1 \leftarrow$ mean value of S_1
    $\mu_2 \leftarrow$ mean value of S_2
    $\mathbf{T}_{previous} \leftarrow \mathbf{T}$
    $\mathbf{T} \leftarrow \frac{1}{2}(\mu_1 + \mu_2)$
end



(a) X-ray image of a defective weld.

(b) Segmented using the *global thresholding* algorithm from Algorithm 1.

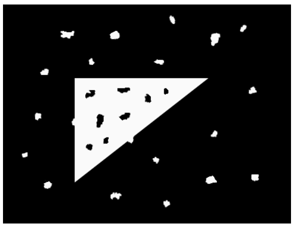
(c) Segmented using *region growing* with four manually selected seed points.

Figure 2: Illustrates two different ways to segment the image in (a).

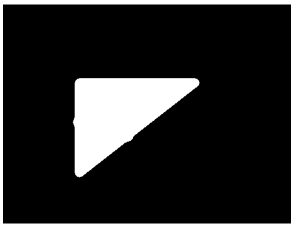
Region growing is a region-based segmentation algorithm that uses a set of seed points and a homogeneity criteria $H(R_i)$ to perform segmentation. For each seed point, a region is grown by inspecting neighbouring pixels and evaluating whether or not to include them in region R_i using $H(R_i)$. The neighbouring pixels that are currently being evaluated are typically called *candidate* pixels. The growing of a region stops when there are no more candidate pixels to inspect. The simplest homogeneity criteria is a threshold, where the threshold defines the maximum difference in intensity between the seed point and the current candidate pixel. Segmenting the image in Figure 2 (a) using region growing with four manually selected seed points can be seen in Figure 2 (c).

b) **[1 point]** Implement a function that segments a greyscale image using the region growing method outlined above. When growing the region around each seed point you may expand your set of candidate pixels using either a von Neumann neighbourhood (4-connectedness) or a Moore neighbourhood (8-connectedness). Apply it on an appropriate image and show the result in your report. Which seed points did you select? *Tip: To select seed points, visualise the original image and pick out the location of pixels which are clearly within the region(s) you want to segment.*

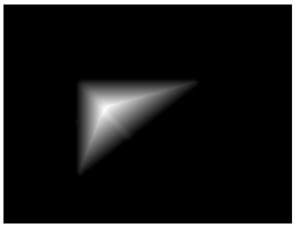
Task 2: Mathematical Morphology [1.5 points]



(a) A noisy binary image. This could, for example, be the result of having applied thresholding on a noisy greyscale image.



(b) An almost noise free version of (a).



(c) A distance transform of the binary image in (b) using chessboard distance.

Figure 3: Two binary and one greyscale image.

a) **[0.5 points]** Use what you know about *erosion*, *dilation*, *opening*, and *closing* to remove the noisy elements from the image in Figure 3 (a). Your result should look something like the one in Figure 3 (b). Explain your noise removal process and show the result in your report. *Tip: Turn the image into a binary image by using thresholding before applying morphological operations.*

The distance transform is an operation typically applied on a binary image and creates a greyscale image where each foreground pixel shows the distance to the closest boundary pixel. An example of this using chessboard distance can be seen in Figure 3 (c). One inefficient way of calculating the distance transform is to use erosion. Intuitively, by using erosion the distance transform for a pixel is simply the number of erosion operations it took to remove it from the foreground of the original image.

b) **[0.8 points]** Implement the distance transform using the erosion method explained above. You can use a 3×3 structuring element of all ones to get chessboard distance. Test the function on the noise free binary image you got from task (a) and show the result in your report.

In previous assignments we have seen how convolution can be used to approximate the gradient of an image. Mathematical morphology operations can also be used for this. One way to extract edge information from images can be seen in Equation 1, where $\ominus$ is morphological erosion. This operation is typically called *boundary extraction* and is usually applied on binary images.

$$A_{\text{boundary}} = A - (A \ominus B) \tag{1}$$

c) **[0.2 points]** Implement a function that extracts the boundary from a binary image as defined in Equation 1. Show the boundary of the noise free binary image from task (a) in your report.

3 Exam Style Q&A [0.5 points]

Create *one* “exam” style question and answer with the following theme: *Segmentation and Morphology*. It should be possible to answer the question in 5-7 minutes. The answer *must* contain all the steps required to get to a solution, and not just the answer itself. Challenging questions are more likely to receive the full amount of points.

You will be asked to add your question and answer – corrected based on any feedback – to a separate document later in the semester.

References

[1] Rafael C Gonzalez, Richard Eugene Woods, and Steven L Eddins *Digital image processing using MATLAB* Gatesmark Publishing, 2009 (cit. on p. 2)