Computer Exercise in Mathematical Morphology

Introduction

Mathematical morphology is a frequently used tool in the field of image analysis for pre- and post-processing of image data. Some common operations are filtering, extraction of connected components, pruning and boundary extraction. Morphology can also be used for shape analysis, for example thinning, where the object is reduced to a "skeleton" by applying topology preserving operations. Morphology is not restricted to binary images. The methods are very useful for grey-level images as well. In this exercise we will, however, only work with binary images.

An image can be looked upon as a rectangular subset of $\mathbf{Z}^2$ on which a function is defined. For binary images the range of this function is the set $\{0, 1\}$, where 1 denotes a digital object and 0 the background.

For this exercise you will use Matlab to represent and manipulate images. If your Matlab installation has Image toolbox installed, you will be able to use the commands *imdilate* and *imerode* for morphological operations. You can check if image toolbox is installed by typing

```
>> help imdilate;
```

If Matlab cannot find imdilate.m an error message will be printed, otherwise a description of the command will be presented. If Image Toolbox is not part of your Matlab installation contact me. Observe that in early versions of Image toolbox, imdilate and imerode were called dilate and erode. You should therefore check these commands if you cannot fins imdilate.

The following example (Matlab script) implements a simple dilation using these commands.

```
clf;
x = -2:2; y = -2:2;
[x y] = meshgrid(x,y);
se = sqrt(x.*x + y.*y)<=2   % Create the structuring element

bild = zeros(20,20);
bild(5:15,5:15)=1;          % Create the image
bild_dil = imdilate(bild,se);
```

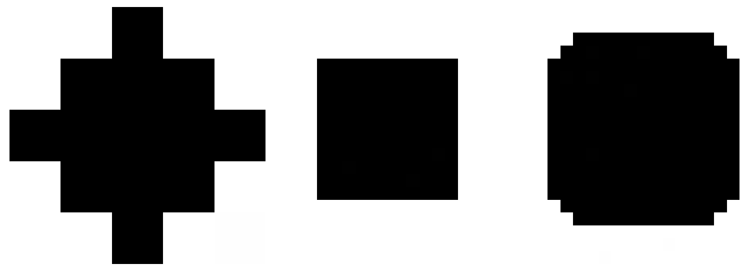

Figure 1: Left: structuring element, Middle: Original image, Right: Dilated image

```
% Plot commands

subplot(1,3,1);
imagesc(bild);
axis('equal');axis('tight');
title('Image');

subplot(1,3,2);
imagesc(se);
axis('equal');axis('tight');
title('Structuring element');

subplot(1,3,3);
imagesc(bild_dil);
axis('equal');axis('tight');
title('Dilated image');
```

Make sure you understand how it works.

When working with the assignments below some useful Matlab commands are:

- imdilate and imerode
- imagesc, imshow, image
- find
- reshape
- load and save (loads and saves Mat-files)

By typing *help command* you can learn about the Matlab command *command*. In the exercises you are asked to test your algorithms on some test images. These are stored as Mat-files and are available at

`http://www.cb.uu.se/~ola/morflab/.`

Figure 2: License plate image to be restaurated.

Assignment 1, Image Restauration

In this assignment you should use morphology for image restauration of a license plate image, Figure 2. The image is taken from an automatic traffic surveillance system. It was extracted from a larger image using (among other things) morphology. As you can see it needs to be restaurated and this is what you should do. Try to remove as much of the noise as possible. Also remove "Dubai" to the left in the image. Your goal is to come as close as possible to the image in Figure 3.

Construct a suitable set of structuring elements and perform morphological operations with these elements. Useful operations are

- Dilations, $A \mapsto A \oplus B, A + B$
- Erosions, $A \mapsto A \ominus B$
- Openings, $A \mapsto A \circ B, A_B$
- Closings, $A \mapsto A \bullet B, A^B$.

Write a Matlab function called *licplate.m*, which takes the image as input argument and returns the restaurated image. You should write a short report which includes

- The restaurated image.
- The source code, licplate.m.
- The structuring elements you used.
- Which morphological operations you used, and in which order.

Figure 3: Restaurated image, only characters and numbers left.

Assignment 2, Image Labelling

A very common task in Image analysis is *image labelling*. This means that given a binary image the connected components should be extracted. Of course the result of this will depend on the definition of connectedness that you use. In this exercise you should implement an algorithm (a Matlab function) that takes a binary image as input and produces a *labelled* image as output. Write two versions, one that extracts four connected objects and one that extracts eigth connected components. Of course you can write one function which takes the connectedness as a parameter. A labelled image is an image where each connected component is represented by a unique number as illustrated below for four connectedness.

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

The morphological approach to this is simple. Let A and Y be sets such that $Y \subset A$ and $p \in Y$ is known. Y is the connected component to be extracted. The following iterative procedure yields all elements of Y

$$X_k = (X_{k-1} \otimes B) \cap Y$$

where $X_0 = p$ and B is the structuring element. To extract all components you have to repeat this until the whole image is labelled. Test your algorithm on the images *eightpattern.mat*, *f16.mat* and *licenseplate.mat*.

Write a short report including

- The source code, labelling.m
- The result of your algorithm on the test images
- A short description of your code.

Good luck!!!

Ola Weistrand ola@cb.uu.se