Computer Exercise: Implementation of Distance Transforms

Introduction

Distance transforms are very useful tools in digital image analysis. Some areas of application are distance measurement between sets (objects), morphological operations, skeletonization, path finding and template matching. Although distances are only approximations of the true Euclidean distances the use of digital distance transforms are often motivated by the low computational cost as well as the ease of implementation.

In this exercise we will consider images consisting of object, A, and background A^c. We will only compute the distance transform of the object not the background. Moreover, for simplicity, we will change the sign convention from the lectures. We will use positive distances inside the object, see Figure 1 where white (large positive values) indicates large distances.

The standard implementation is basically a dynamic programming approach. The original image is tranformed so that A^c is equal to zero and A to infinity (in practise a high enough number). This image is then updated in two scans over the image, a forward scan followed by a backward scan. The weights for different directions are illustrated in Figure 2. Here *a, b* and *c* are the costs when moving from the center to the new position. When using 3×3 neighborhoods c equal zero. The forward mask positions are boldface and the backward positions are italic. In pseudo code the algorithm looks like this:

```
Initialize A^c to zero and A to infinity

Forward scan:
for i = 1, 2 , ...  , nr of lines; do
   for j = 1, 2,..., nr of columns; do
     v^1_{i,j} = min_{(k,l)∈fmask}{v^0_{i+k,j+l} + fmask(k,l)}

Backward scan:
for i = nr of lines, nr of lines - 1, ...  , 1; do
   for j = nr of columns, nr of columns - 1, ..., 1; do
     v^2_{i,j} = min_{(k,l)∈bmask}{v^1_{i+k,j+l} + bmask(k,l)}
```

Figure 1: Distance transform (City Block) of a hand object.

Assignment

You should write a Matlab program that implements the three distance transforms $a = 1, b = 1$, $a = 3, b = 4$ and $a = 5, b = 7, c = 11$. Test your code on the images **square.mat**, **phantom.mat** and **rabbit.mat**. For each of these images a Euclidean distance image has been computed **esquare.mat**, **ephantom.mat** and **erabbit.mat**. These images is found at `www.cb.uu.se/~ola/distanslab/`. When your code is working correctly you should compare your result with these images by plotting the absolute value of the difference images. Use for example the Matlab command *surf* or (and) visualize the result as a new image. Write a short report with your results and your code and send it to me.

Good luck,

Ola Weistrand
weistrand@math.uu.se

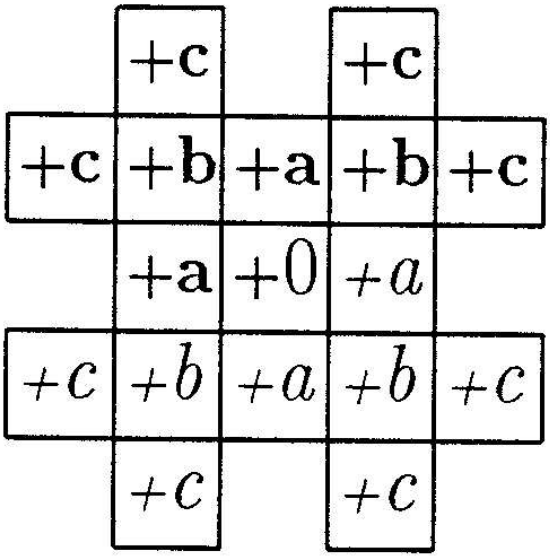

Figure 2: The different steps.