

## Laboration in Dynamic programming

### Getting started

- This lab consists of the following files:  
"Lab Dynamic Programming.pdf" – the lab instructions  
dp.m, run.m, run2.m – Matlab files
- If you type run you will get the following message: "You need some code here in dp.m!".  
Your job is to modify the file *dp.m* by implementing the dynamic programming algorithm.

### Modify *dp.m*

- Use the dynamic programming algorithm that is described in the lecture notes and set the smoothness cost to  $2\Delta y$ .
- The return vector should contain the y-coordinates of the detected edge, .i.e. the edge points  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  gives  $\text{edge} = [y_1 \ y_2 \ \dots \ y_n]$ .
- The resulting cost accumulation matrix (CAM) with backtracing arrows can be seen in Figure 1. Hint: implement the backtracing arrows in a separate matrix.
- **NOTE:** The edge in this example is easy to detect and can therefore be correctly detected with a lot of different edge detection algorithms, but the purpose of this laboration is to implement the dynamic programming algorithm. This means that you will get a return if the implementation is incorrect, even if it detects the correct edge. Make sure that both the CAM-matrix and the backtracing matrix is correct!

2	4	8	10	10
4	4	5	7	8
1	3	8	13	9
2	4	6	10	12

Figure 1. The resulting CAM-matrix, with backtracing arrows, for "run.m".

### Test *run2.m*

- When you are sure that your version of *dp.m* is correct, you can take a look at “*run2.m*”. It detects the horizon in the image *trees.tif* (included in the Image Processing Toolbox).
- Start by looking at the code and try to understand what it does. Do you understand how the cost matrix (see Figure 2) is computed? Why are `WEIGHT_EDGE` and `WEIGHT_INTENSITY` negative?

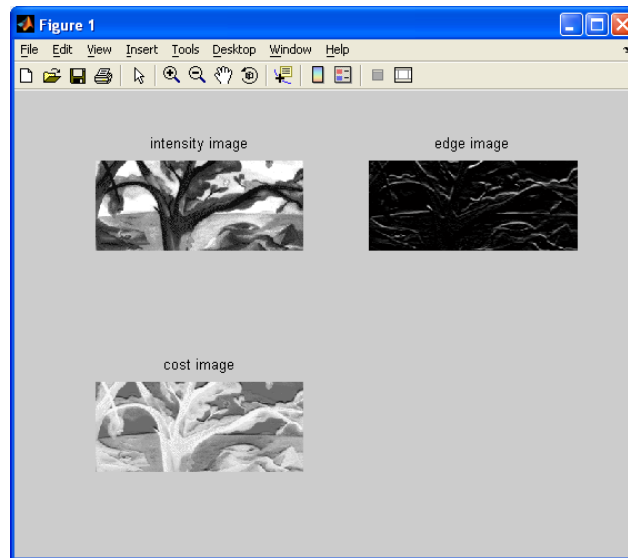


Figure 2. The intensity, edge, and cost image (cost matrix) of “*run2.m*”.

- Run *run2.m* to see if works with your version of *dp.m*. See Figure 3 for the correct result.

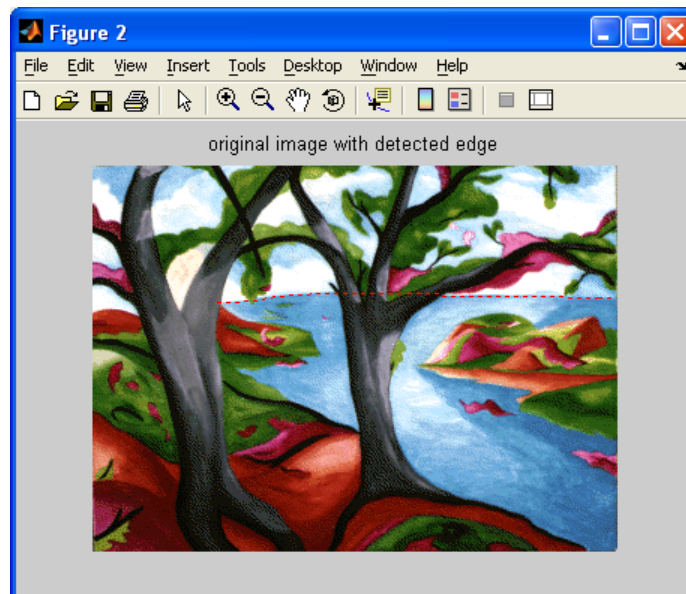


Figure 3. The correct result from “*run2.m*”.

### Labreport

Send your **commented** code for *dp.m* to: see the *Image Analysis SSY095 course homepage* !