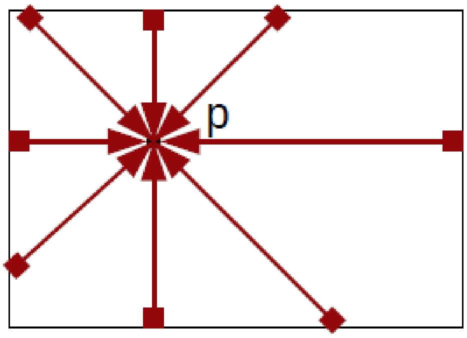
3DP LAB 1 - STEREO MATCHING

In this first laboratory, the goal was to extend the provided C++ software implementing the computation of the path costs and the following cost aggregation, so basically perform the disparity maps estimation of stereo images.

The first thing I did in this homework is figure out how the path was calculated and the number of it. Basically, there are 8 paths per scan, as you can see in the image below, and each path is specified by a combination of direction (***-1***, ***0***, or ***1***). For example, ***direction_x = 0*** and ***direction_y = 1*** means a top to bottom vertical path, north to south.



The second step was defining the variable ***start_x, start_y, end_x, end_y, step_x*** and ***step_y***, using the structure ***pw_***, in which are stored the minimum and maximum horizontal and vertical coordinates, as you can see in the table below.

path	***direction_x***	***direction_y***	***start_x, end_x, step_x***	***start_y, end_y, step_y***
0 ↑	0 →	-1 ↑	pw_.west, pw_.east, 1	pw_.south, pw_.north, dir_y
1 ↓	0 →	1 ↓	pw_.west, pw_.east, 1	pw_.north, pw_.south, dir_y
2 ←	-1 ←	0 ↓	pw_.east, pw_.west, dir_x	pw_.north, pw_.south, 1
3 ↖	-1 ←	-1 ↑	pw_.east, pw_.west, dir_x	pw_.south, pw_.north, dir_y
4 ↙	-1 ←	1 ↓	pw_.east, pw_.west, dir_x	pw_.north, pw_.south, dir_y
5 →	1 →	0 ↓	pw_.west, pw_.east, dir_x	pw_.north, pw_.south, 1
6 ↗	1 →	-1 ↑	pw_.west, pw_.east, dir_x	pw_.south, pw_.north, dir_y
7 ↘	1 →	1 ↓	pw_.west, pw_.east, dir_x	pw_.north, pw_.south, dir_y

The third step was the one of computing the right cost for each path, which is updated after each iteration of ***compute_path_cost()***. To do it I use the formula that was given in the slide, which is the showed in the figure below, which as the purpose of restrict the range of resulting values without affecting the minimization procedure.

$$E(p_i, d) = E_{data}(p_i, d) + E_{smooth}(p_i, p_{i-1}) - \min_{0 \le \Delta \le d_{\max}} E(p_{i-1}, \Delta)$$

$$E_{smooth}(p, q) = \min \begin{cases} E(q, f_q) & \text{if } f_p = f_q \\ E(q, f_q) + c_1 & \text{if } |f_p - f_q| = 1 \\ \min_{0 \le \Delta \le d_{\max}} E(q, \Delta) + c_2 & \text{if } |f_p - f_q| > 1 \end{cases}$$

The last step was the one of aggregating the costs for all direction into the ***aggr_cost_*** tensor.

Since the value of MSE error was pretty high, I also tried the formula below, which is the simplified version of the formula just seen above, but the result was pretty similar.

$$E(p_i, d) = E_{data}(p_i, d) + E_{smooth}(p_i, p_{i-1})$$

The result obtained from the source code that you find in the file ***sgm.cpp***, so the disparity maps and the error reported by the application are the following, with ***disparity range*** equal to ***85***:

Right Image MSE error: 122.464

Right Image MSE error: 475.166

Right Image MSE error: 557.735