Student Name: Federico Gelain
Student ID: 2076737

UNIVERSITÀ DEGLI STUDI DI PADOVA
3D Data Processing
Lab 1: Stereo Matching SGM + Monocular

Task 1: Computation of the path cost

The path cost for each pixel p_i (i refers to the position of the pixel in the path) has been computed using the following equation:

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

with

$$E_{smooth}(p, q) = min \begin{cases} E(q, f_q) & if\ f_p = f_q \\ E(q, f_q) + c_1 & if\ |f_p - f_q| = 1 \\ min_{0 \leq \Delta \leq d_{max}} E(q, \Delta) + c_2 & if\ |f_p - f_q| > 1 \end{cases} \tag{2}$$

The three terms of equation (2) are called, respectively, **no penalty term**, **small penalty term** and **big penalty term**.
Referring to the code provided, given a pixel $\boldsymbol{p}$ with coordinates $\boldsymbol{(cur\_x, cur\_y)}$, its path cost for the path of index $\boldsymbol{cur\_path}$ is equal to $\boldsymbol{path\_cost\_[cur\_path][cur\_x][cur\_y][d]}$, $\forall d \in [0, disparity\_range\_)$
We have two cases depending on what position the pixel is in the path:

- if it's the first in the path, meaning that it's located in the border of the image, than there's no previous pixel, and so its path cost is just equal to the first term $E_{data}(p, d)$;

- if it's not the first in the path, then the 2^{nd} and 3^{rd} terms of equation (1) have to be computed as well. The latter one, stored in **best_prev_cost**, is obtained by iterating through all disparities and finding the minimum value of $E(p_{i-1}, d)$, where p_{i-1} is the previous pixel of p in the current path. It's computed before E_{smooth} to optimize the running time of the algorithm, since **big penalty term** is simply equal to $best\_prev\_cost + c_2$ ($p2\_$ in the code) and there is no need to recompute it inside the loop used to compute the other two penalty terms for each disparity d since it's always the same.
  As mentioned what's missing are **no penalty term** and **small penalty term**. The former is simply equal to the path cost of the previous pixel of p in the path with the same disparity d. The small penalty term varies depending on the value of d:

  - if $d = 0$ or $d = disparity\_range\_ - 1$, then it's simply equal to the path cost of the previous pixel of p in the path at disparity $d + 1$ and $d - 1$ respectively;
  - otherwise both path costs for the previous pixel of p at disparity $d - 1$ and $d + 1$ have to be considered. Since at the end the minimum is taken, we can already choose the smaller one between the two;

  p1_ is then added to get the correct value of the small penalty term. By choosing the minimum among those penalty costs $E_{smooth}(p_i, p_{i-1})$ is obtained and finally the path cost of p can be computed following equation (1).

Task 2: Aggregation

In this function, to correctly compute the path cost, the choice of the start, end and step value for both x and y direction is crucial. To determine them, it's enough to check separately the value of the directions dir_x and dir_y and handle all their 3 possible values (+1, 0, -1) using a series of if conditions:

- if $dir\_x = 1$, it means that the path goes from left to right, so it starts from position pw_.west and ends at pw_.east, with a step of +1. Similarly for $dir\_y = 1$, meaning a top to bottom path, which starts from pw_.north and ends at pw_.south, with a step of +1. This is also the initialization for the cases $dir\_x = 0$ and $dir\_y = 0$;
- if instead $dir\_x = -1$, it means that the path goes from right to left, so it starts from pw_.east and ends at pw_.west, with step -1. Similarly for $dir\_y = -1$, meaning a bottom to top path, which starts from pw_.south and ends at pw_.north, with step -1.

Task 3: Computation of scale factor for monocular disparity

To compute the linear coefficients **h** and **k** to scale the monocular disparity values, the good confidence disparities estimated from SGM and the corresponding unscaled disparities from the right-to-left initial guess of the disparity map are stored in two separate arrays of the dimension n_valid, called d_{sgm} and d_{mono} respectively. With those vectors, h and k can be computed in the following way:

$$\begin{bmatrix} h \\ k \end{bmatrix} = (A^T A)^{-1} A^T b \tag{3}$$

where $b = d_{sgm}$ and $A = [d_{mono}\ \overline{1}]$ is the matrix obtained by stacking horizontally the vector of monocular disparities and the vector of ones of size n_valid.

Task 4: Replacement of low confidence disparities with the corresponding scaled monocular ones

With the coefficients computed in the previous task, the low confidence disparities (obtained by simply iterating through all disparities and checking the ones that satisfy the negation of the condition used to find the good ones) can be replaced with the corresponding scaled monocular disparities as:

$$d_{sgm} = h * d_{mono} + k \tag{4}$$

Results

Below are listed all the MSE values for the disparity maps obtained by the implementation of SGM with and without the refinement step, with a disparity range value of 85:

Data Item	Aloe	Cones	Plastic	Rocks1
MSE without refinement	122.464	475.166	820.049	557.735
MSE with refinement	13.7291	17.4342	348.223	34.6984

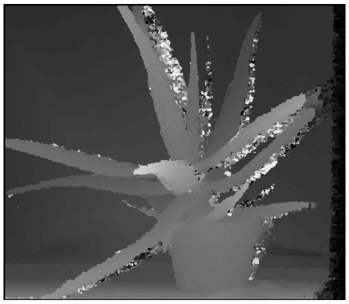

Figure 1: Aloe disparity map computed with SGM without refinement

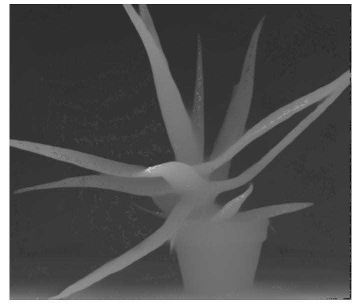

Figure 2: Aloe disparity map computed with SGM refined with scaled initial guess disparities

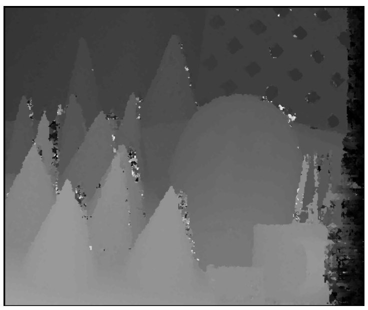

Figure 3: Cones disparity map computed with SGM without refinement

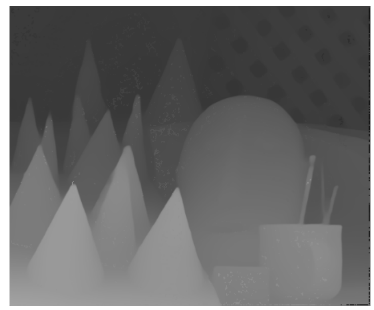

Figure 4: Cones disparity map computed with SGM refined with scaled initial guess disparities

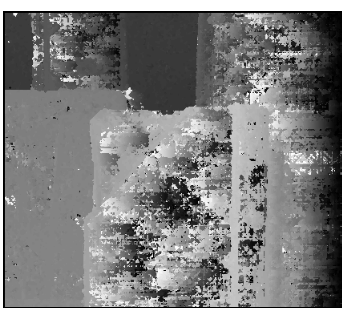

Figure 5: Plastic disparity map computed with SGM without refinement

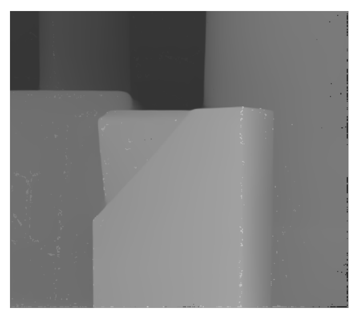

Figure 6: Plastic disparity map computed with SGM refined with scaled initial guess disparities

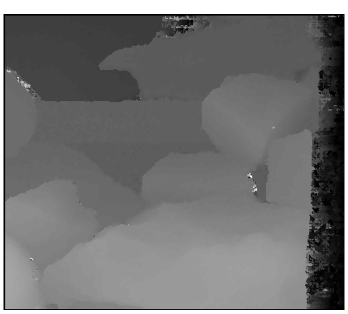

Figure 7: Rocks1 disparity map computed with SGM without refinement

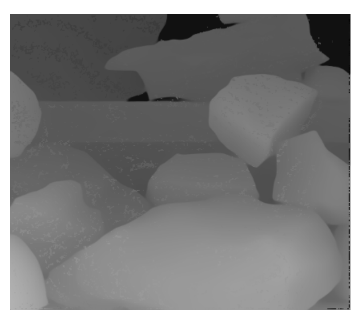

Figure 8: Rocks1 disparity map computed with SGM refined with scaled initial guess disparities