Stefano Binotto 2052421

REPORT STEREO MATCHING

The first method I implemented was **compute_path_cost**. This method aims to compute **path_cost_** $\boldsymbol{E(p_i, d)}$, the cost relative to a given pixel $\boldsymbol{p_i}$, given a disparity $\boldsymbol{d}$, following a direction depending on the path we are considering.

Firstly, I implemented the computation of path_cost_ for the first pixel of the path. This was easy since the path cost, in this case, is equal to the matching cost $\boldsymbol{E_{data}(p_i, d)}$:

$$E(p_i, d) = E_{data}(p_i, d)$$

For all the other pixels, I had to compute the whole formula to find the path cost:

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

where $\boldsymbol{p_{i-1}}$ is the pixel immediately previous the current pixel in the considered path.

To do so, initially I found the **best_prev_cost**, $\boldsymbol{min_{0 \le \Delta \le d_{max}} E(p_{i-1}, \Delta)}$, which is the minimal cost related to the previous pixel among all the possible disparities.

Then I just had to find the **smooth term** $\boldsymbol{E_{smooth}(p_i, p_{i-1})}$. I know that the smooth term is the following:

$$E_{smooth}(p_i, p_{i-1}) = min \begin{cases} E\left(p_{i-1}, f_{p_{i-1}}\right) & if\ f_p = f_{p_{i-1}} \\ E\left(p_{i-1}, f_{p_{i-1}}\right) + c_1 & if\ \left|f_{p_i} - f_{p_{i-1}}\right| = 1 \\ min_{0 \le \Delta \le d_{max}} E(p_{i-1}, \Delta) + c_2 & if\ \left|f_p - f_{p_{i-1}}\right| > 1 \end{cases}$$

So in the first case, I found the **no_penalty_cost** which is equal to the path cost of the previous pixel $\boldsymbol{E\left(p_{i-1}, f_{p_{i-1}}\right)}$, in the case where the disparities of the two pixels were the same ($\boldsymbol{f_p = f_q}$).

In the second case I had to find the **small_penalty_cost** $\boldsymbol{E\left(p_{i-1}, f_{p_{i-1}}\right) + c_1}$, when $\boldsymbol{\left|f_p - f_q\right| = 1}$. This was a little bit more complex. Since the difference between the two disparities is only 1, then three different cases can occur:

1. If the disparity $\boldsymbol{f_p}$ is zero, then $\boldsymbol{f_{p_{i-1}}}$ can only be equal to $\boldsymbol{f_p + 1}$, so the **small_penalty_cost** is equal to the path cost of $\boldsymbol{p_{i-1}}$ given the disparity $\boldsymbol{f_p + 1}$**:**
$$E\left(p_{i-1}, f_p + 1\right) + c_1$$
2. If the disparity $\boldsymbol{f_p}$ is the maximum disparity value, then $\boldsymbol{f_{p_{i-1}}}$ can only be equal to $\boldsymbol{f_p - 1}$, so the **small_penalty_cost** is equal to the path cost of $\boldsymbol{p_{i-1}}$ given the disparity $\boldsymbol{f_p - 1}$:
$$E\left(p_{i-1}, f_p - 1\right) + c_1$$
3. If the disparity $\boldsymbol{f_p}$ is not zero and not the maximum disparity value, then $\boldsymbol{f_{p_{i-1}}}$ can be equal to both $\boldsymbol{f_p - 1}$ and $\boldsymbol{f_p + 1}$, so there are two **small_penalty_cost** values**:**
$$E\left(p_{i-1}, f_p - 1\right) + c_1$$
$$E\left(p_{i-1}, f_p + 1\right) + c_1$$

Finally I computed the third case, the **big_penalty_cost**, adding c_2 to the **best_prev_cost** I found above:

$$min_{0\le\Delta\le d_{max}} E(p_{i-1},\Delta) + c_2$$

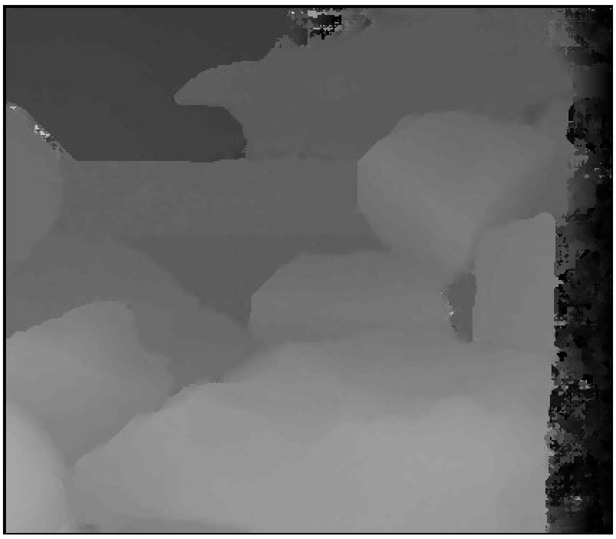
The second method I implemented was **aggregation** to aggregate the costs for all the directions. Firstly I initialized the start_x, end_x, step_x variables depending on the path and the directions I am considering.
- If the **dir_x** is equal to **+1**, then the scanning goes from left (***start_x = west***) to right (***end_x = east+1***), so I set the ***step_x = +1***.
- If the **dir_x** is equal to **0**, then I decided to develope the scanning along the x dimension from right (***start_x = east***) to left (***end_x = west-1***), so I set the ***step_x = -1***.
- If the **dir_x** is equal to **-1**, then the scanning goes from right (***start_x = east***) to left (***end_x = west-1***), so I set the ***step_x = -1***.

Secondly I initialized the start_y, end_y, step_y variables depending on the path and the directions I am considering:
- If the **dir_y** is equal to **+1**, then the scanning goes from up (***start_y = north***) to down (***end_y = south+1***), so I set the ***step_y = +1***.
- If the **dir_y** is equal to **0**, then I decided to develope the scanning along the y dimension from down (***start_y = south***) to up (***end_y = north-1***), so I set the ***step_y = -1***.
- If the **dir_y** is equal to **-1**, then the scanning goes from down (***start_x = south***) to up (***end_x = north-1***), so I set the ***step_y = -1***.

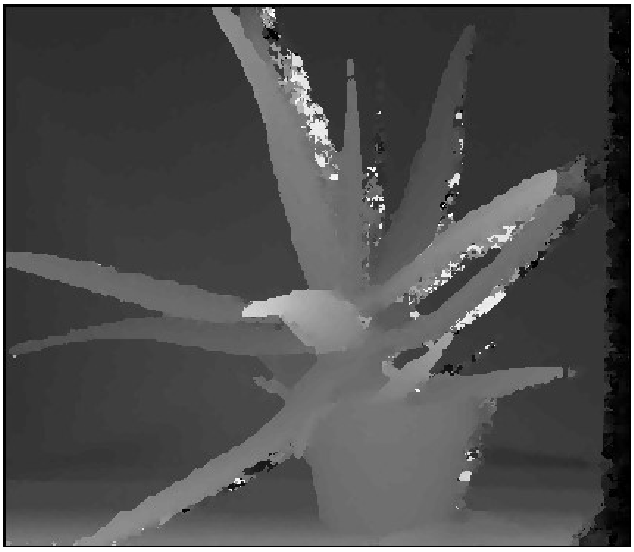
At the end I aggregated the costs into the **aggr_cost_** tensor by adding, for every pixel and for every disparity, all the path costs of every path. Since the data term $E_{data}(p_i, d)$ is repeatedly added eight times, I removed seven repeated data terms.

Results: (with maximum disparity value = 85)

Rocks -> MSE = **346.657**



Aloe -> MSE = **106.987**



Cones -> MSE = **470.417**