Assignment 1 - Stereo Matching

Introduction

This assignment involves implementing the Semi-Global Matching (SGM) algorithm to estimate the disparity map from a pair of rectified stereo images.

Implementation

The focus of this assignment is on the aggregation function of the SGM algorithm, which comprises two steps: (i) computation of the path cost for all pixels along all paths, and (ii) aggregation of the costs over the paths.

Computation of the path cost

The `compute_path_cost` function computes the cost for a pixel along a given direction. It implements the following dynamic programming equation:

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

where p_i is the current pixel, p_{i-1} is the previous pixel along the path, d is the disparity value, E_{data} is the precomputed cost volume, and E_{smooth} is the smoothness cost, computed as follows:

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

If the given pixel is the first one on the path, the function assigns it a smoothness cost of 0. In the code, this check is performed by comparing the pixel coordinates with absolute cardinal points, and by considering the current directions.
For all other pixels, the above formulas apply. In this case, the smoothness cost is computed as the minimum among four quantities: (i) the path cost of the previous pixel on the current path for the current disparity *d*; (ii) the path cost of the previous pixel on the current path at disparity *d-1*, plus a small regularization factor *p1_* (if *d-1* is not properly defined, this is set to infinity); (iii) the path cost of the previous pixel on the current path at disparity *d+1*, plus a small regularization factor *p1_* (if *d+1* is not properly defined, this is set to infinity); (iv) the minimum path cost at the previous pixel on the current path, plus a big regularization factor *p2_*. Finally, to ensure numerical stability, the best path cost at the previous pixel on the current path is subtracted from all the path costs.
In comparison to the original formula, this implementation differs in term (iv). Instead of computing the minimum costs of the previous pixel for only the disparity values that differ at least 2 from the current disparity, the minimum costs are computed for all disparity

values. Mathematically, this is equivalent to the original formulation because the seemingly additional costs in the minimum of (iv) are offset by a smaller regularization cost in the other terms of the minimization (i), (ii), and (iii). Therefore, they do not impact the calculation.

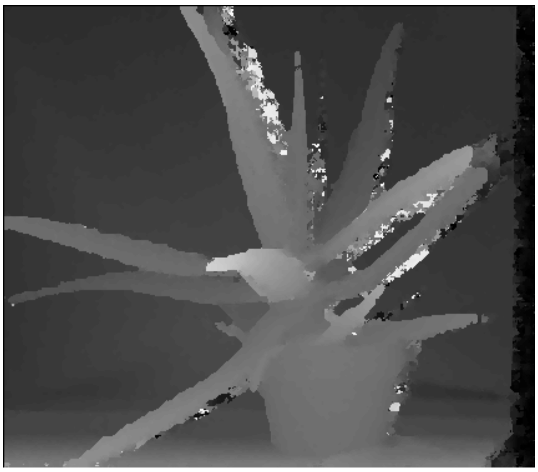
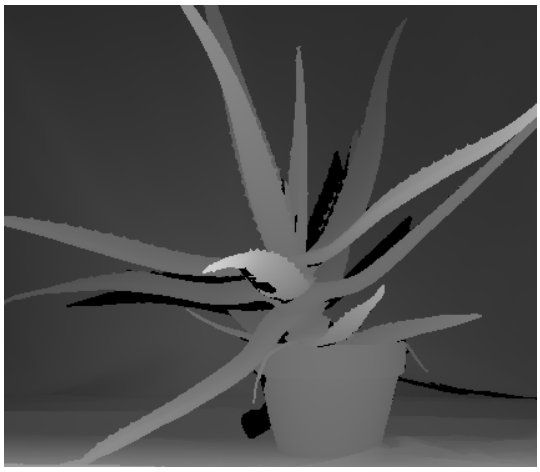
Aggregation

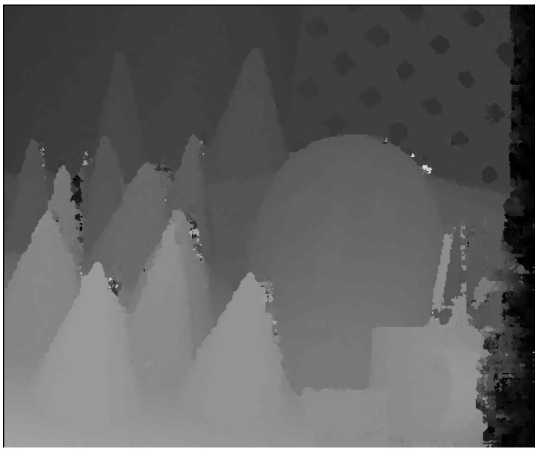
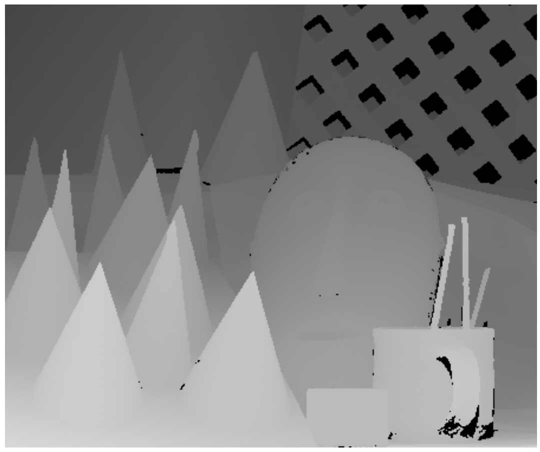
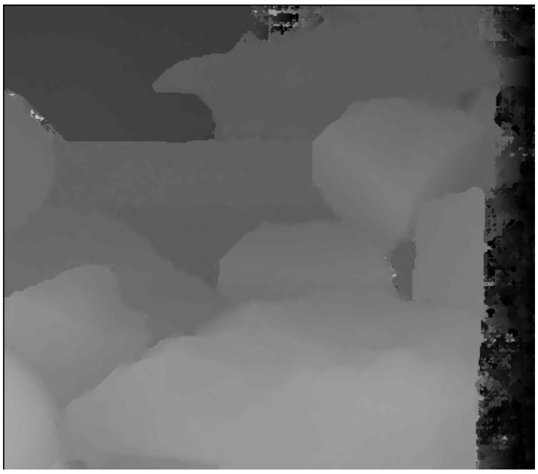
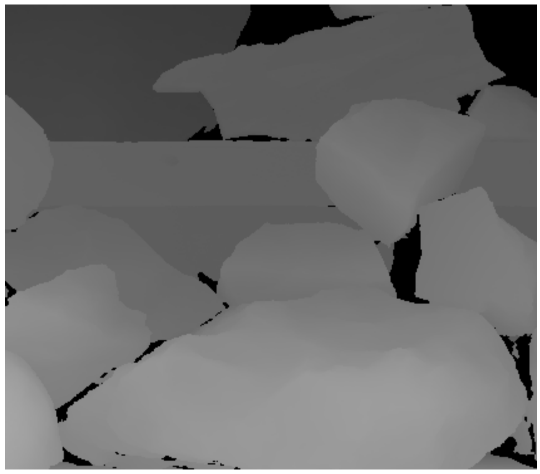
The `aggregation` step computes the final cost of each pixel at all possible disparity values. This is achieved by summing up the following terms: the data term, stored in the `cost_` tensor; the smoothness term, obtained by summing all path costs stored in the `path_cost_` tensor, minus the corresponding data term. This final subtraction's purpose is to consider the data cost only once per pixel, independently of the number of paths used to compute the smoothness cost.

Results

The table below summarizes the obtained results, providing a quick qualitative and quantitative analysis. The output disparity map can be compared with the ground truth disparity map, while the Mean Squared Error (MSE) offers a quantitative measure of the difference between the output and the ground truth disparity maps.

All reported results refer to the disparity map of the right image and were obtained using disparity levels ranging from 0 to 85. This range has been empirically proven to be reasonable.

Input	**Output disparity**	**Ground truth**
Aloe		
	MSE: 97.3548	

<table>
<tr><td rowspan="2">Cones</td><td></td><td></td></tr>
<tr><td colspan="2">MSE: 449.287</td></tr>
<tr><td rowspan="2">Rocks1</td><td></td><td></td></tr>
<tr><td colspan="2">MSE: 323.356</td></tr>
</table>

The results are in line with expectations: they capture depth quite well while being affected by some noise.

One interesting observation is that the output disparity maps have a black vertical band on the right side. This may be because the rightmost pixels in the right stereo image have no corresponding pixels in the left image.

Additionally, there is a very thin black frame along the sides of the image. This is likely due to the Block Matching approach used to compute the cost volume, where a window is slid across the two images. The window size doesn't allow computation for border pixels, resulting in a cost volume of 0 for those pixels. This hypothesis is supported by the fact that using a larger window size in the algorithm results in a wider black band along the border.