MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Accurate and Efficient Stereo Processing by Semi-Global Matching and Absolute Difference, Squared Difference and Normalized Cross Correlation

Teacher: Florin Oniga

Students: Cristian – Ioan Blaga

Vlad – Cristian Buda

Group: 30433

Contents

1. Introduction

The objective of this project is the development of a stereo processing algorithm capable of creating a disparity map out of two rectified images. The main inspiration for our project is the *"Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information"* article written by Heiko Hirschmuller from the Institute of Robotics and Mechatronics, Germany.

The general idea behind the project is that, given two rectified images, one representing a left view of the scene, and the other one representing the right view of the same scene, depth information regarding the scene can be extracted. There are multiple algorithms that can be used, but for this particular project we have decided to make use of the Semi-Global matching algorithm.

The Semi-Global Matching (SGM) method is based on the idea of pixelwise matching of different matching costs and approximating a global, 2D smoothness constraint by combining many 1D constraints. The algorithm is described in distinct processing steps, assuming a general stereo geometry of two or more images with known epipolar geometry.

2. Algorithms

The prerequisites of Semi-Global Matching algorithm are the following:

- two rectified images representing the left and right view of a scene; the images need to be grayscale
- a cost volume representing the cost of the pixelwise matching of the two images

Considering the fact that the two given images are rectified, the corresponding pixel of a pixel from the left image in the right image always lies to its left, on the same row. More specifically, if the pixel from the left image is on the position (x,y1) and the corresponding pixel from the right image is on the position (x,y2), we know that $y2 \leq y1$. Moreover, the difference between the y coordinates represents **the disparity** (d) between the two pixels, i.e. $d = y1 - y2$.

Having this in mind, the cost volume for the Semi-Global Matching algorithm is defined as the cost of the pixelwise matching at every disparity level between 0 and the maximum disparity chosen depending on the context. For this project, three different cost functions have been used in order to compute the pixelwise matching cost. Each cost function is implemented using a configurable size window. The window is first set in the left image and based on the current disparity level the corresponding window is found in the right image. Then, the cost function is applied on these two windows. The cost functions used for this project are the following:

a) **Absolute difference**

The absolute difference cost is defined as follows:

$$absoluteDifference = \sum \frac{|left(x,y) - right(x,y-d)|}{width * height}$$

b) **Squared difference**

The squared difference cost is defined as follows:

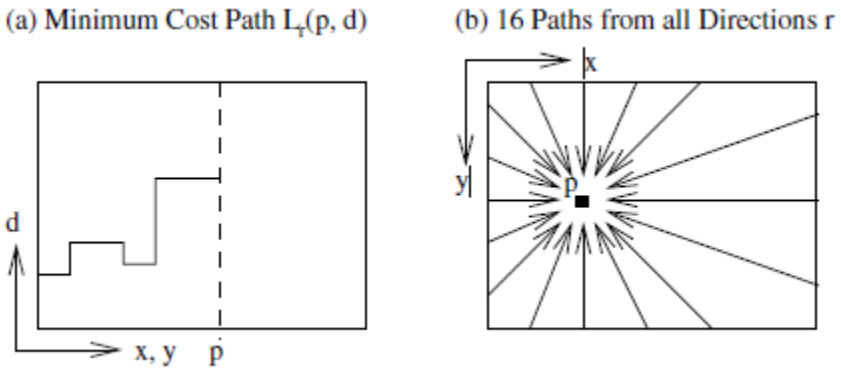
$$squaredDifference = \sum \frac{(left(x,y) - right(x,y-d))^2}{width * height}$$

c) **Normalized cross-correlation**

The normalized cross-correlation is defined as follows:

$$normalizedCrossCorrelation = \frac{1}{width * height} \sum \frac{(left(x,y) - mean(left)) * (right(x,y-d) - mean(right))}{\sqrt{var(left) * var(right}}$$

Having computed the cost volume, the Semi-Global Matching algorithm represents the aggregated (smoothed) cost S(p,d) for a pixel p and disparity d, which is calculated by summing the costs of all 1D minimum cost paths that end in pixel p at disparity d. It is noteworthy that only the cost of the path is required and not the path itself.



$$L_{\mathbf{r}}(\mathbf{p},d) = C(\mathbf{p},d) + \min(L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},d),$$
$$L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},d-1) + P_1, L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},d+1) + P_1,$$
$$\min_i L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},i) + P_2) - \min_k L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},k)$$

Let Lr be a path that is traversed in the direction r. The cost Lr(p,d) of the pixel p at disparity d is defined recursively. The C parameter represents the pixelwise matching cost. The remainder of the equation adds the lowest cost of the previous pixel p−r of the path, including the appropriate penalty for discontinuities. The values of Lr permanently increase along the path,

which may lead to very large values. However, by subtracting the minimum cost of the previous pixel from the whole term, an upper limit is established.

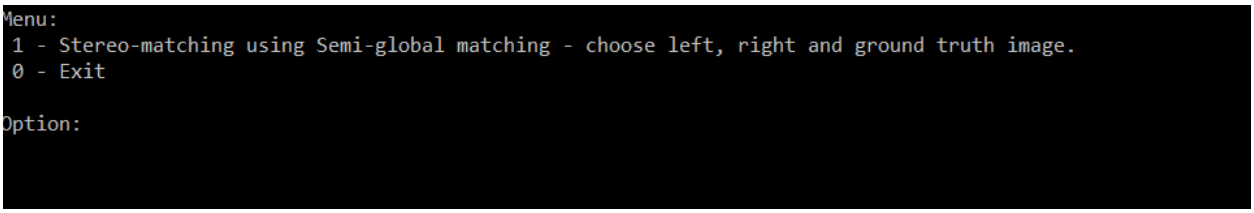
Afterwards, the costs Lr are summed over paths in all directions r:

$$S(\mathbf{p},d)=\sum_{\mathbf{r}}L_{\mathbf{r}}(\mathbf{p},d)$$

Finally, the disparity map can be computed by choosing, for each pixel **p**, the disparity **d** that minimizes the cost previously defined.

3. Implementation

The main entry point of the application is represented by the command line prompt that pops up when the executable is run. In this command line prompt, a menu is shown for the user to interact with the application:



By choosing the first option, the process for computing the disparity map image starts. The user first has to select three images representing:

- The left view of the scene
- The right view of the scene
- A ***ground truth*** image representing a disparity map image previously computed for the scene that will be used in order to test the accuracy of our algorithm

The first step of the process is represented by the conversion of the two views of the scene from an RGB representation (3 channels) to a Grayscale representation (1 channel). This can be done by computing, for each pixel, the average of these three channels.

Afterwards, the SGM algorithm is applied for each cost. Thus, the steps for each iteration are the following:

- Compute the cost volume using the chosen cost (squared, absolute, normalized cross correlation). The window size is set to 7. When centering the window, the pixels that would otherwise be outside of the image are set to 0.
- Afterwards, apply the SGM algorithm on the cost volume previously computed. We have chosen our algorithm to follow eight paths (the 8 neighbors around the

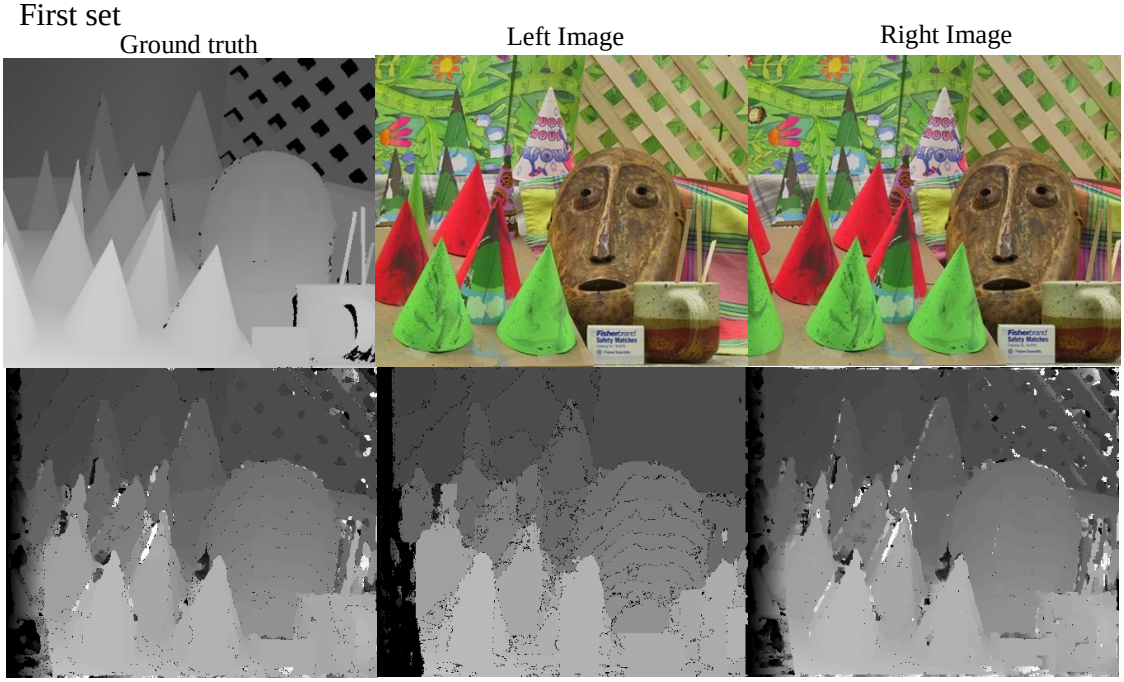
pixel). By applying the algorithm, we will have computed the **S** cost volume defined in the previous section.

- After computing the **S** cost volume, we can now create the disparity map by choosing, for each pixel **p**, the disparity **d** that minimizes the cost **S.** However, in order to avoid certain uncertainties regarding disparities, we have decided to make use of the second minimum in order to be able to take a decision regarding our chosen disparity. More specifically, if the global minimum is not smaller than a certain threshold from the second minimum, then the disparity is uncertain.
- With the disparity map computed, we can compute the accuracy of our algorithm by comparing it to the ground truth image. More specifically, we can compare each disparity from our disparity map to the disparity from the ground truth. If the disparity difference between the two is smaller than 2, then we consider this to be a **correct** disparity.
- Finally, the result images are shown to the user (and saved locally) together with the accuracy costs.

4. Results

The SGM algorithm (together with the costs) have been applied on the ***Middlebury*** datasets for stereo matching (they also provide ground truth images for these sets).

a) First set

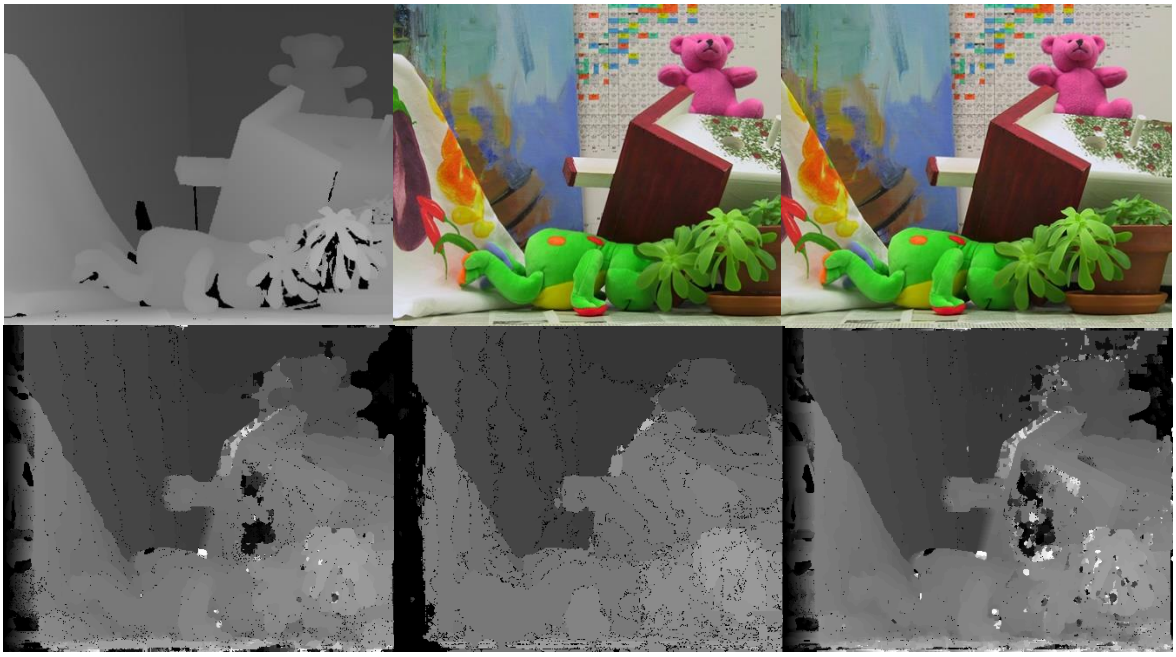


For this set, the accuracy tested against the ground truth is the following:

- Absolute difference accuracy = **79.97%**
- Squared difference accuracy = **77.61%**
- Normalized cross correlation = **79.00%**

b) Second set

Ground truth | Left Image | Right Image



Absolute Difference | Normalized Cross Correlation | Squared Difference

For this set, the accuracy tested against the ground truth is the following:

- Absolute difference accuracy = **76.97%**
- Squared difference accuracy = **74.53%**
- Normalized cross correlation = **77.35%**

5. Conclusion

In conclusion, we have implemented a stereo processing algorithm capable of creating a disparity map out of two rectified images using three different cost functions. Out of the three cost functions, it can be seen that all three of them give quite good accuracy results when compared to the ground truth image. As a further development of this project, what one could do is to try applying different cost functions for this process (such as the mutual information cost presented in the article from where this project was inspired).

6. Bibliography

http://vision.middlebury.edu/stereo/data/scenes2003/

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.8897&rep=rep1&type=pdf