ETH zürich | D INFK

November 15, 2018

Computer Vision

Lab 06: Stereo Matching

Sumeet Gyanchandani

In this lab, we implement a winner-takes-all and a graph-cut based stereo algorithm to recover a disparity map from a set of images. Using the results of dense stereo, we generate a textured 3D model.

1 Stereo Setup

We begin with rectifying the images and reading the camera parameters from the files.

Figure 1. The left and the right images

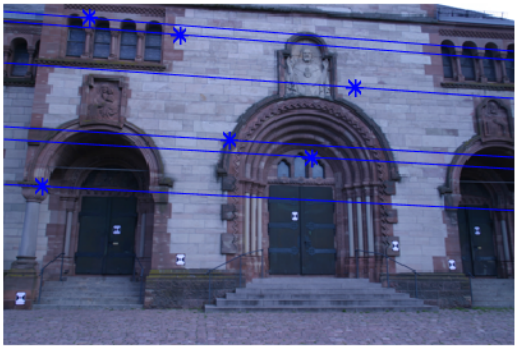

Figure 2. Epipolar lines of the left image

Figure 3. Epipolar lines of right image

Figure 4. The left and the right rectified images

2 Winner-takes-all Stereo Matching

I have implemented a winner-takes-all stereo using SSD. For each disparity in the disparity range, we shift the entire image by disparity value, compute image difference (SSD), convolve with a box filter and remember the best disparity for each pixel. I tried 3 different average filter window sizes (3 x 3, 5 x 5, 7 x 7) for the disparity computation. The following subsections compare the disparity maps among these sizes.

2.1 Filter size 3x3

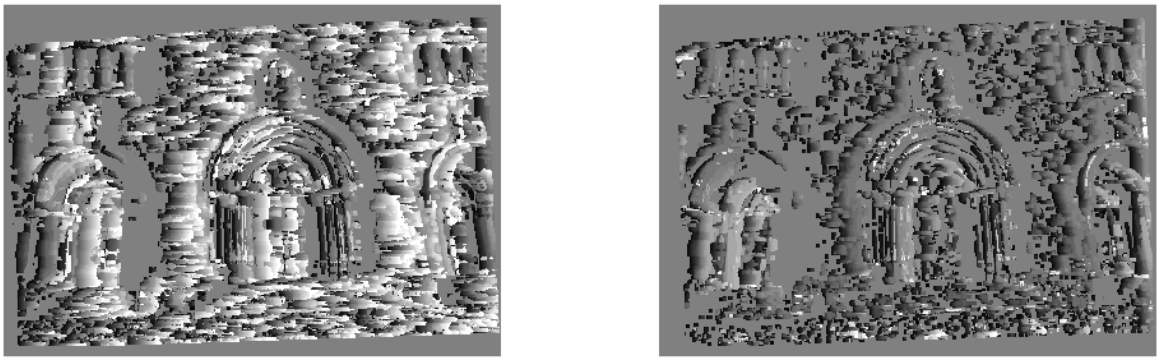

Figure 5. Disparity Map for Filter size 3x3

Figure 6. Image mask for Filter size 3x3

2.2 Filter size 5x5

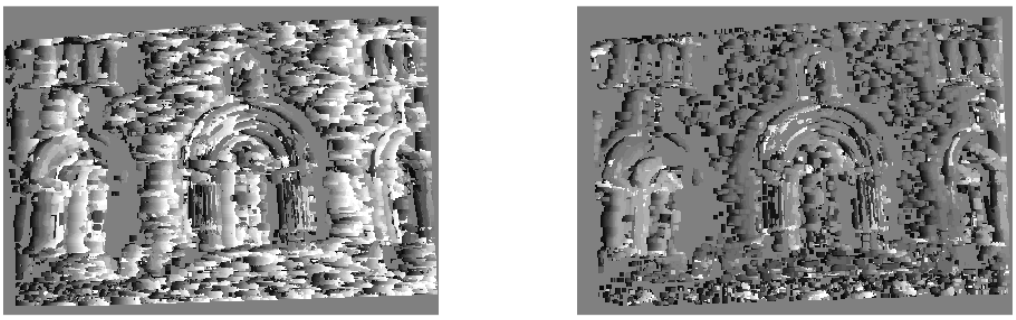

Figure 7. Disparity Map for Filter size 5x5

Figure 8. Image mask for Filter size 5x5

2.3 Filter size 7x7

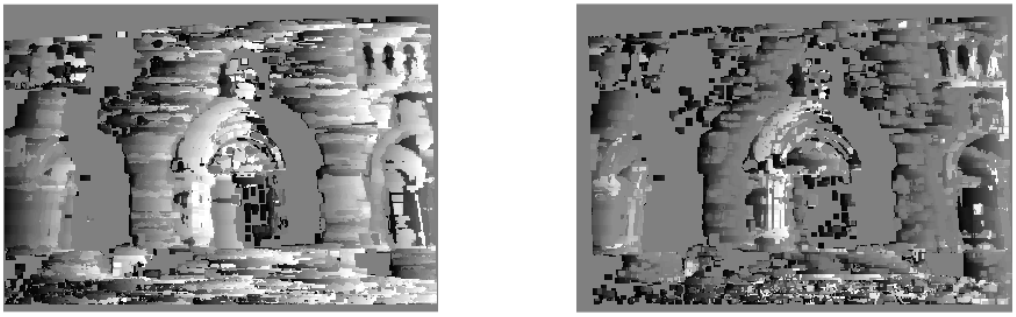

Figure 9. Disparity Map for Filter size 7x7

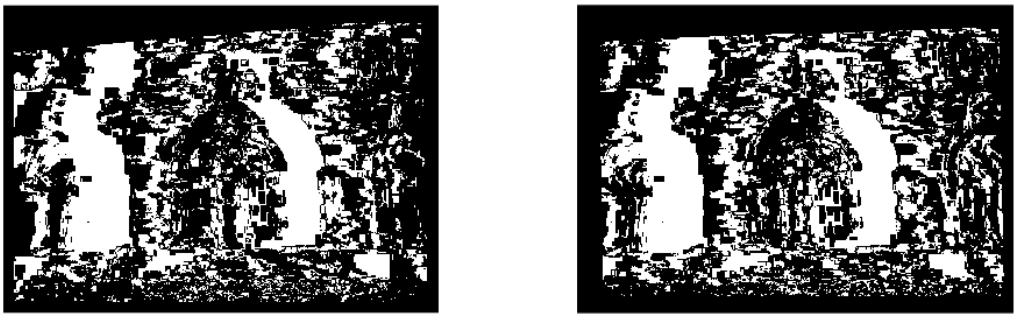

Figure 10. Image mask for Filter size 7x7

3 Graph-cut Stereo Matching

For the Graph Cut stereo matching algorithm, we need to formulate the computation of disparity as a graph labeling problem. Each pixel corresponds to a graph node and each disparity to a label. The goal is to find a labeling $f : P \rightarrow L$ which minimizes:

$$E(f) = E_{\text{data}}(f) + \lambda E_{\text{smooth}}(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{p,q \in N} S(f_p, f_q)$$

where, P represents the set of pixels and L represents a discrete set of labels corresponding to different disparities. $D_p(f_p)$ is the cost of assigning label f_p to pixel p and it is given by the SSD value calculated for the disparity corresponding to label f_p. N is the set of neighboring pixels and $S(f_p, f_q)$ is the cost of assigning labels f_p and f_q to neighboring pixels p and q. The idea is to penalize neighboring pixels having different labels. Hence, for finding the data costs we use the exact same algorithm as in the previous section, but we instead of finding the smallest SSD values we stored all the values and let the graph-cut algorithm find the best labelling for us.

Figure 11. Disparity Map for Filter size 3x3

Figure 12. Image mask for Filter size 3x3

4 Generating a textured 3D model

Using the result of our dense stereo algorithms and the camera parameters, we triangulate the corresponding 3D point for each pixel and generate a textured 3D model. The generated textured *.obj-files* from 3D coordinates and images are viewed using **MeshLab**.

4.1 Winner-takes-all

The 3D model for filter size 5x5 was very sparse. This motivated me to try larger window sizes.

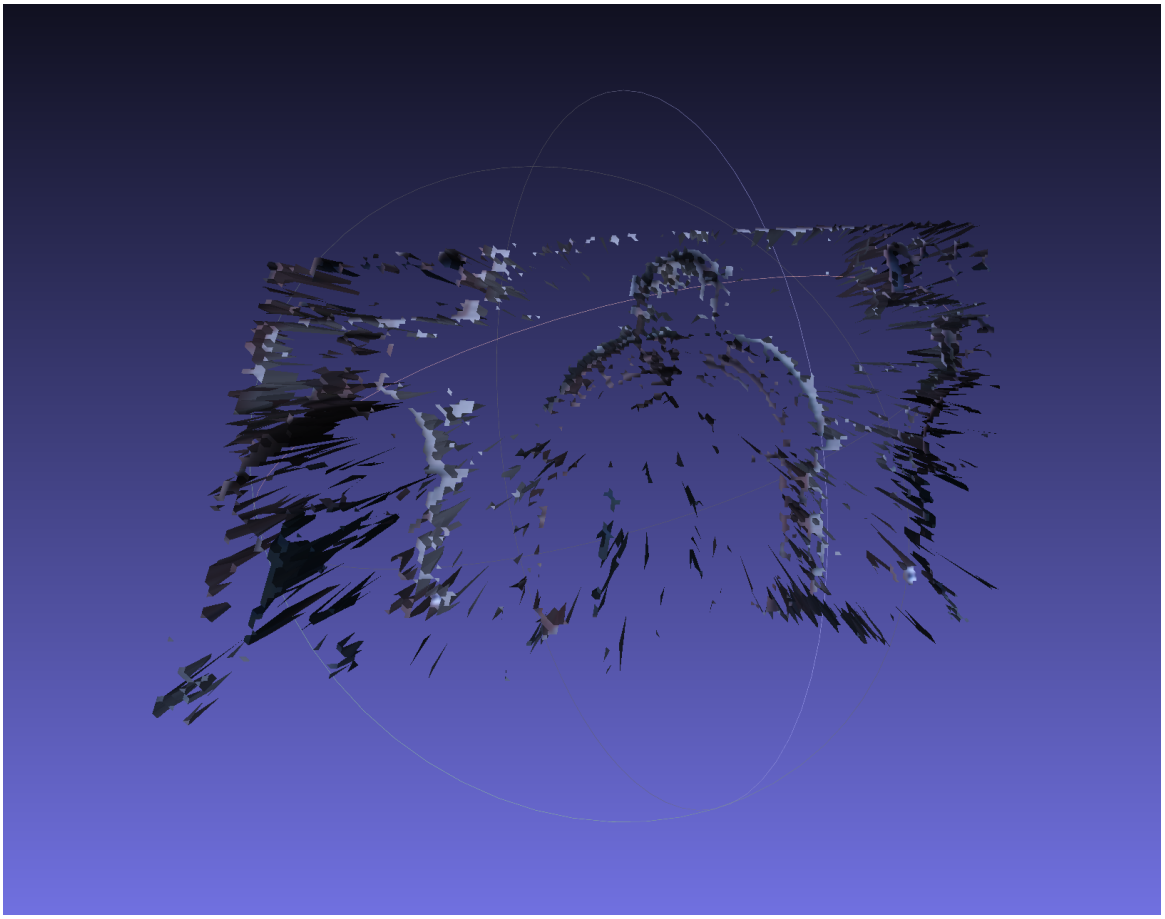

Figure 13. 3D model for Filter size 5x5

The following images correspond to the filter size of 25x25:

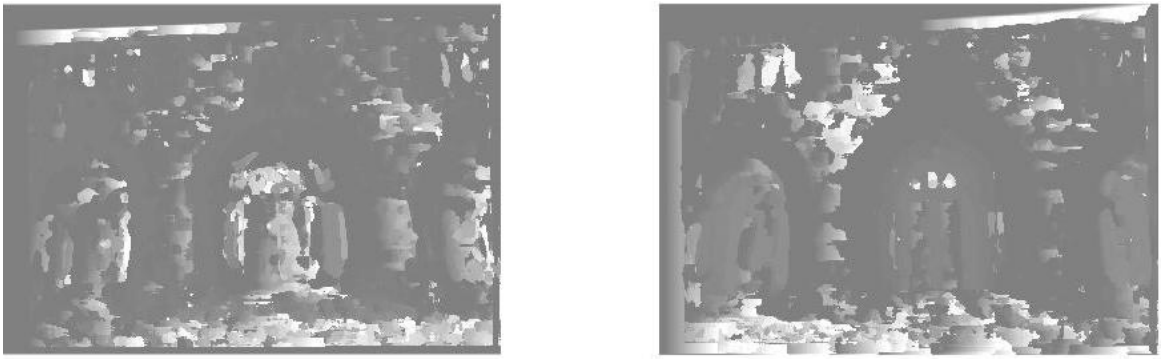

Figure 14. Disparity Map for Filter size 25x25

Figure 15. Image mask for Filter size 25x25

Figure 16. 3D model for Filter size 25x25

The following images correspond to the filter size of 50x50:

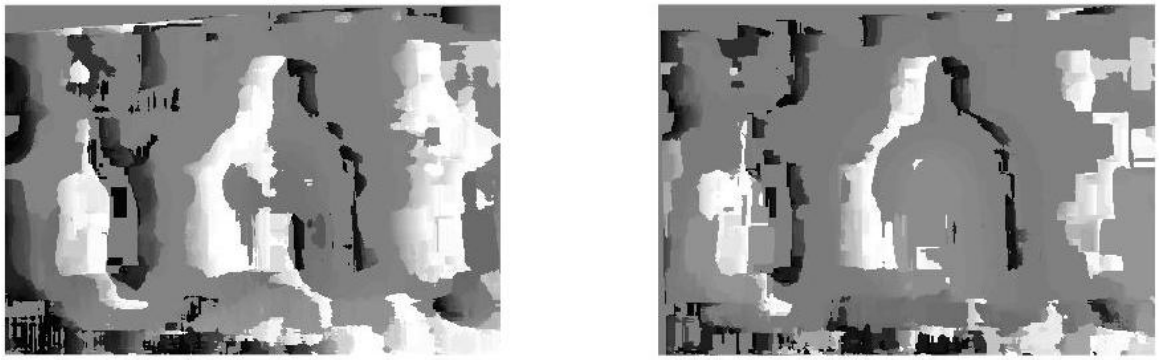

Figure 17. Disparity Map for Filter size 50x50

Figure 18. Image mask for Filter size 50x50

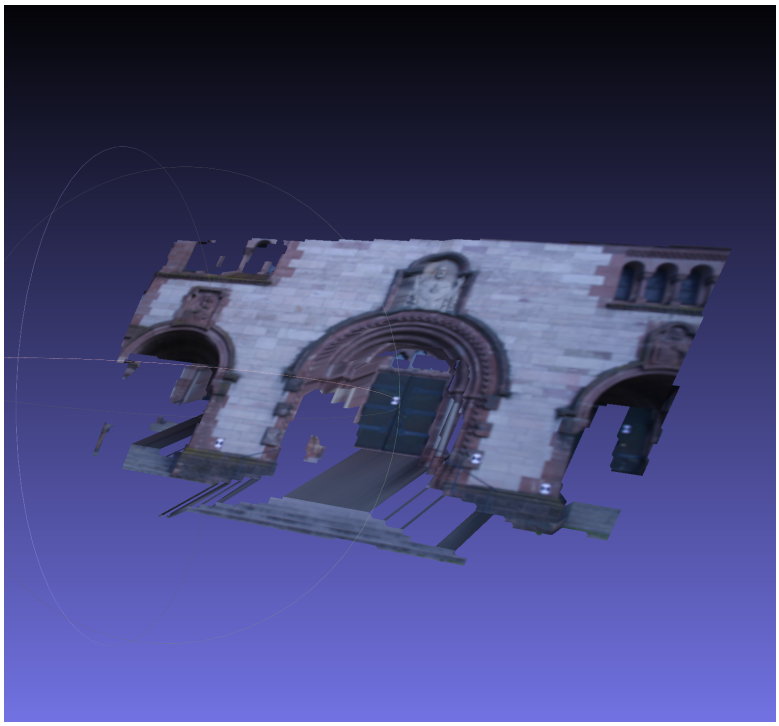

Figure 19. 3D model for Filter size 50x50

I observed that increasing the filter size also increases the resolution of the textured 3D model.

4.2 Graph Cut

However for Graph-Cut algorithm, filter size of 3x3 was perfect!

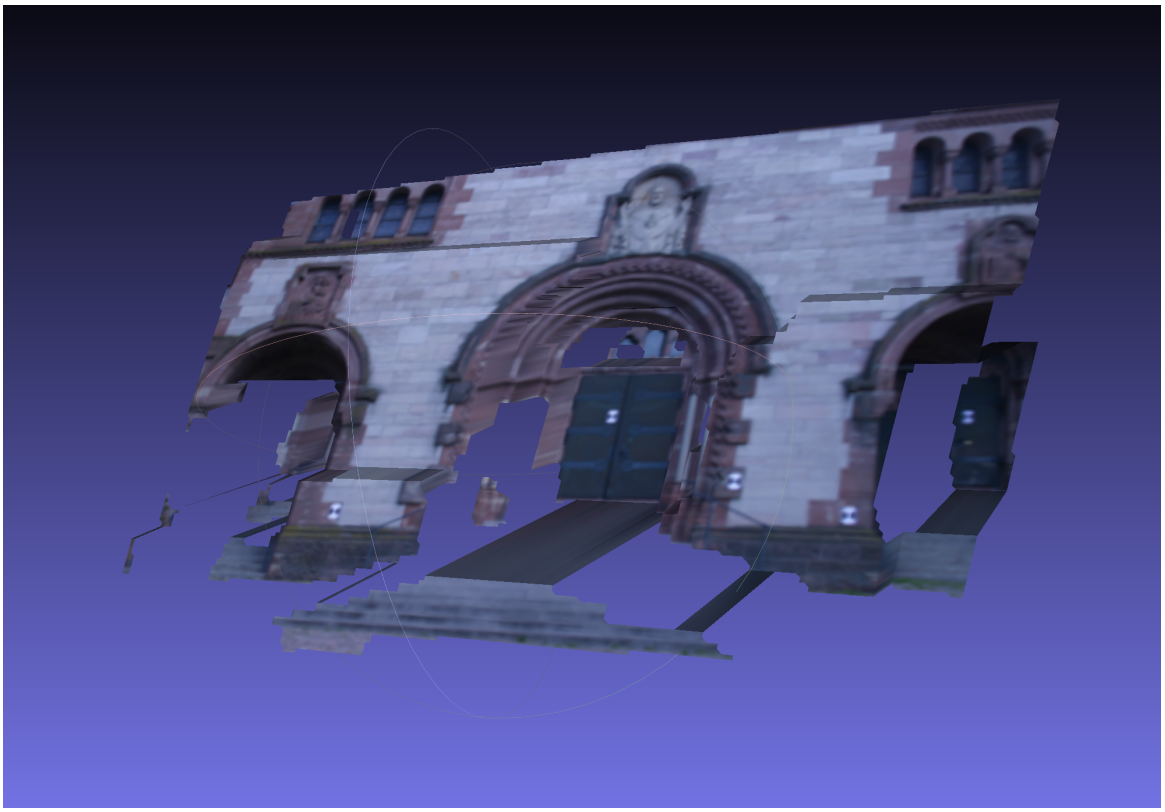

Figure 20. 3D model for Filter size 3x3

5 Automatic Disparity Range Computation

For computing the disparity range automatically, we need to find the point correspondences. I used **vl_sift** and **vl_ubcmatch** from the VLFeat library and then use RANSAC to filter outliers. I select the largest disparity from the inliers. Sometimes disparities can be very large, hence it is safer to limit the maximum disparity. In our example, I have capped the disparity to **40**, this value would change according to the image.