

# Lab Report 06 - Stereo Matching

Manuel Galliker 14-921-969  
manuelga@student.ethz.ch

November 15, 2019

## 1 1. Disparity Computation

In the first part of this exercise a winner-takes-all stereo matching algorithm was implemented. It uses SSD (Sum of Squared Distances). First the functions *imgRectL* and *imgRectR* are used to project the images onto a single plane. This would correspond with only a translation in horizontal (*x*) direction two cameras at the same height above ground.

Since the orientation of the two cameras is identical the epi-poles lie at infinity and the epi-lines are parallel and horizontal. From this we know that from a point *a* in the first image a corresponding point *a'* in the second image can be obtained by a translation in *x* direction from the coordinate of point *a*.

Then The implemented algorithm uses a range of pixels (by default  $[-40, 40]$ ) to shift the second image in *x* direction for the hole range. For each shift the difference between the two filtered images is calculted and then filtered by an average filter using an aerial *filter\_window\_size* and the functions *fspecial* and *imfilter*. Where then this difference is smallest its assumed to have found a good matching point and the distance the point is shifted is stored in the disparity map.

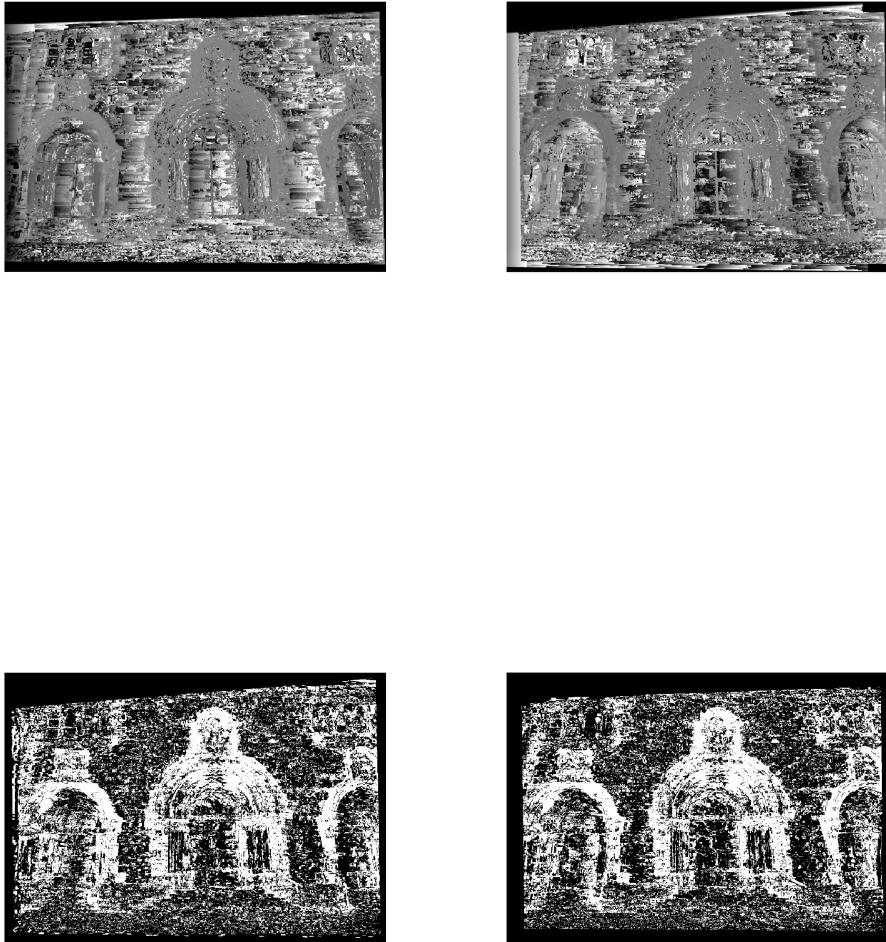


Figure 1: 3x3 Filter Window

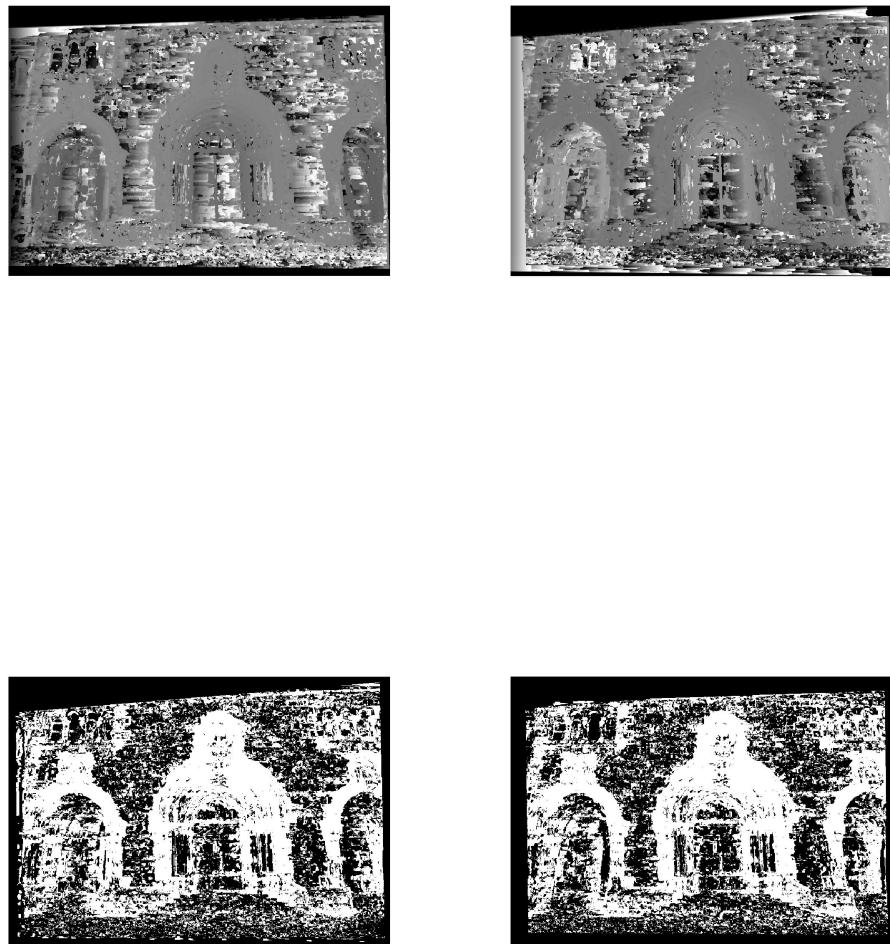


Figure 2: 5x5 Filter Window

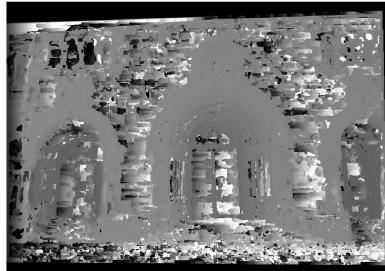


Figure 3: 7x7 Filter Window

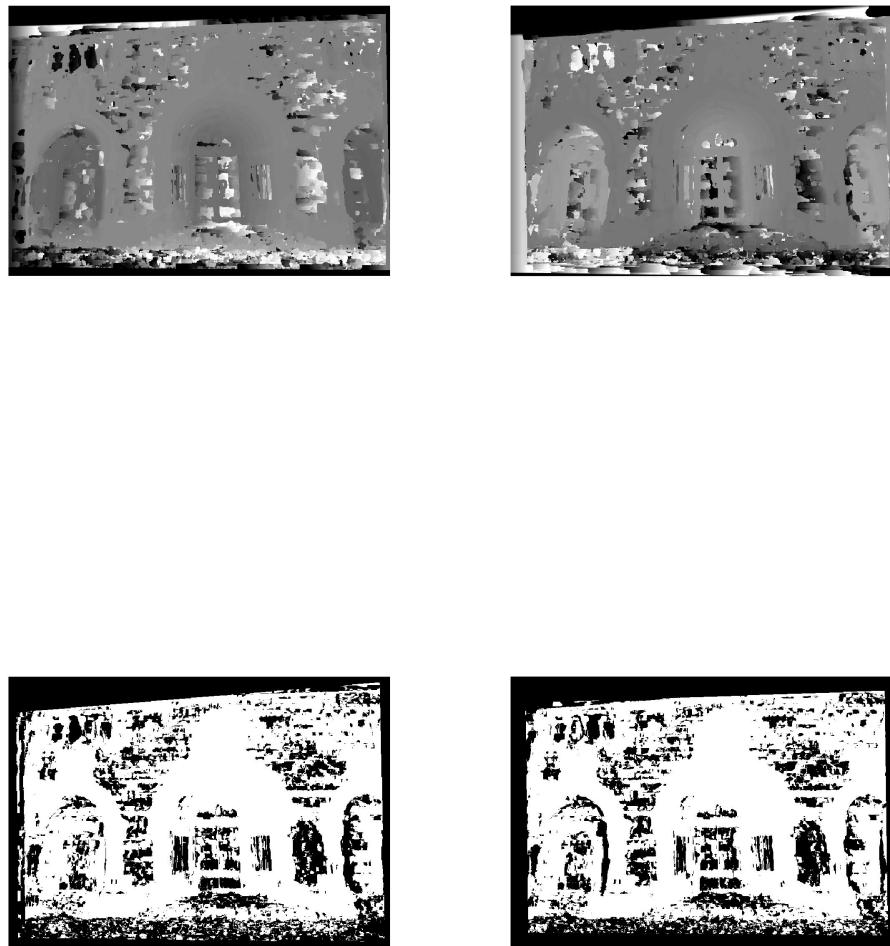


Figure 4: 11x11 Filter Window

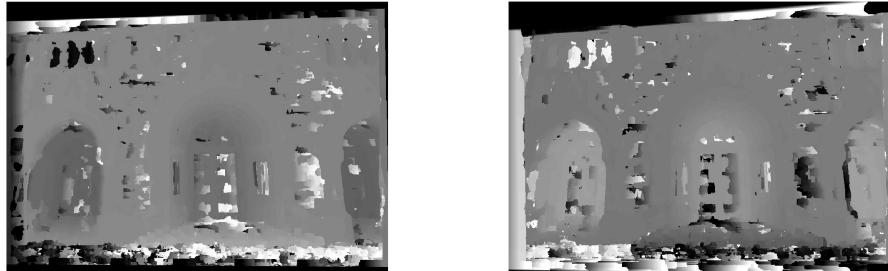


Figure 5: 15x15 Filter Window

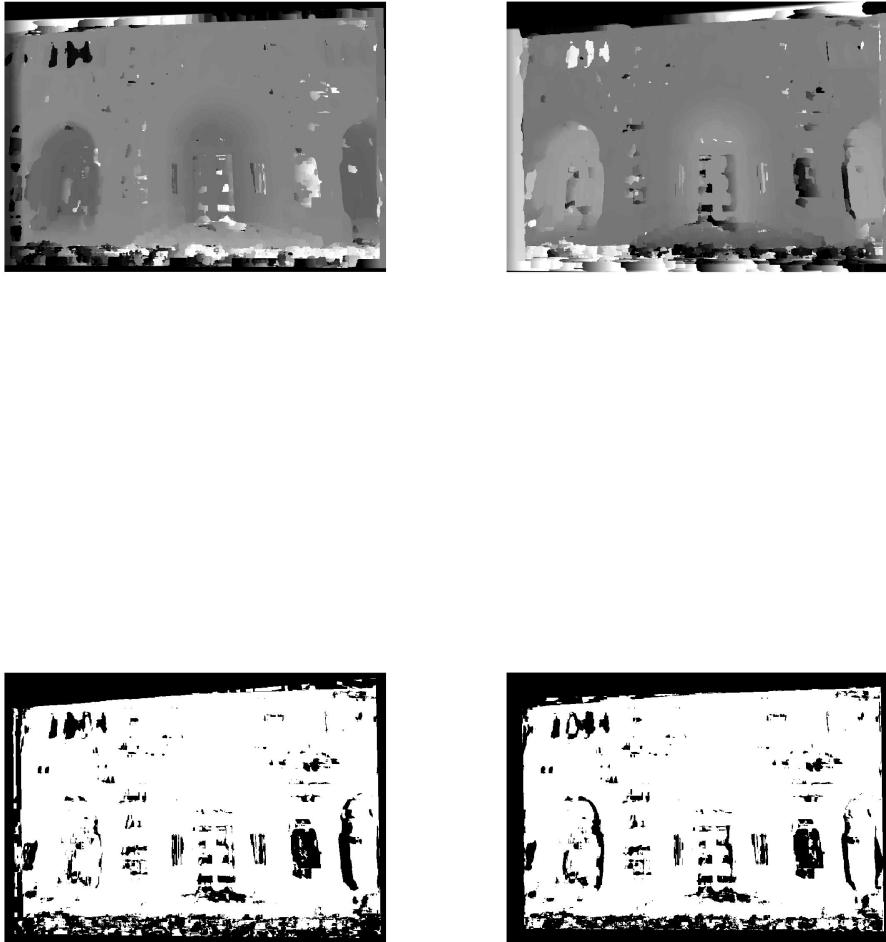


Figure 6: 19x19 Filter Window

By looking at the black and white pictures the points which are correctly mapped (in white) according to the given function can be seen. As the filter window is increased the disparity map gets less sensitive to noise, which makes intuitive sense. As can be seen, especially around edges of the stairs and the half circle elements above the door some details can be lost and edges get smoothed out. So depending on the application a tradeoff value has to be found here.

Furthermore, it can be seen that the algorithm performs especially poorly on the flat stone brick surface. Since the stonebricks are arranged in a pattern this might lead to some wrong distance matches (especially between stone block edges) with the given range. Without adjusting the range this can only be fixed by extensive filtering.

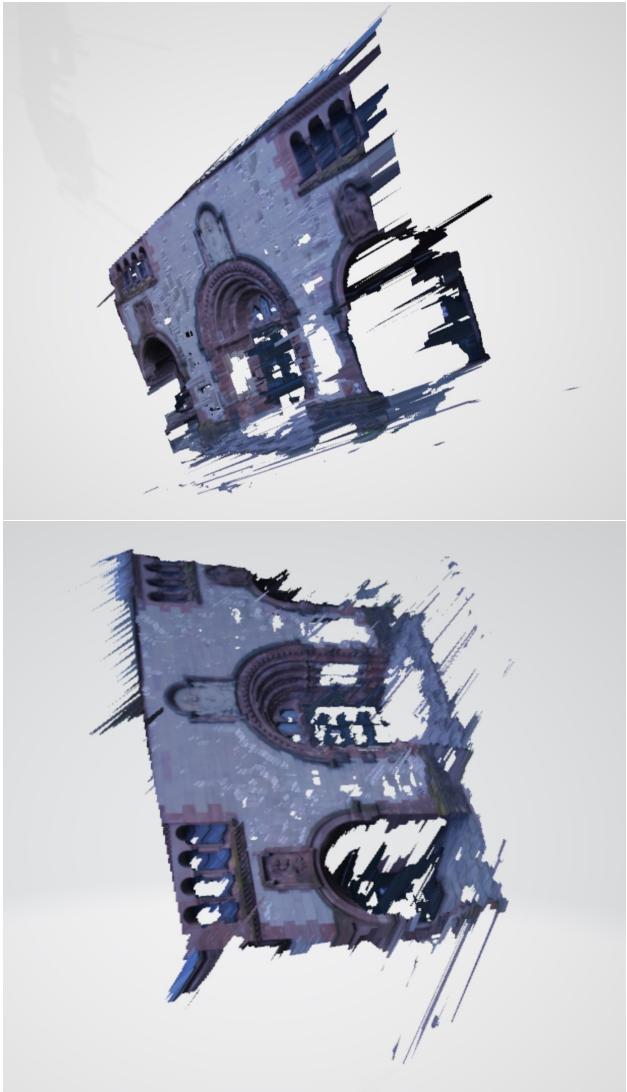


Figure 7: Textured 3D Model from Disparity Computation with range  $[-40, 40]$  and filter Window Size 19

As can be seen in the picture above a somewhat usable 3D reconstruction can already be obtained with the right tuning of this algorithm. Doors, stairs and Windows are already visible.

## 2 Graph Cut

Similar to the exercise before a depth map is obtained by this time using the provided graph cut function. The most important parameters to tune are the filter window size (which was chosen to be 7x7) as well as the scaling of the data cost DC and the smoothness cost SC. Here, Dc is linked to the cost of assigning a label to a pixel of interest and a higher value gives a model with higher accuracy. Sc is linked to assigning labels to the pixels around it and a higher value gives a smoother result.

Here are a few of the tried results with variation of DC:

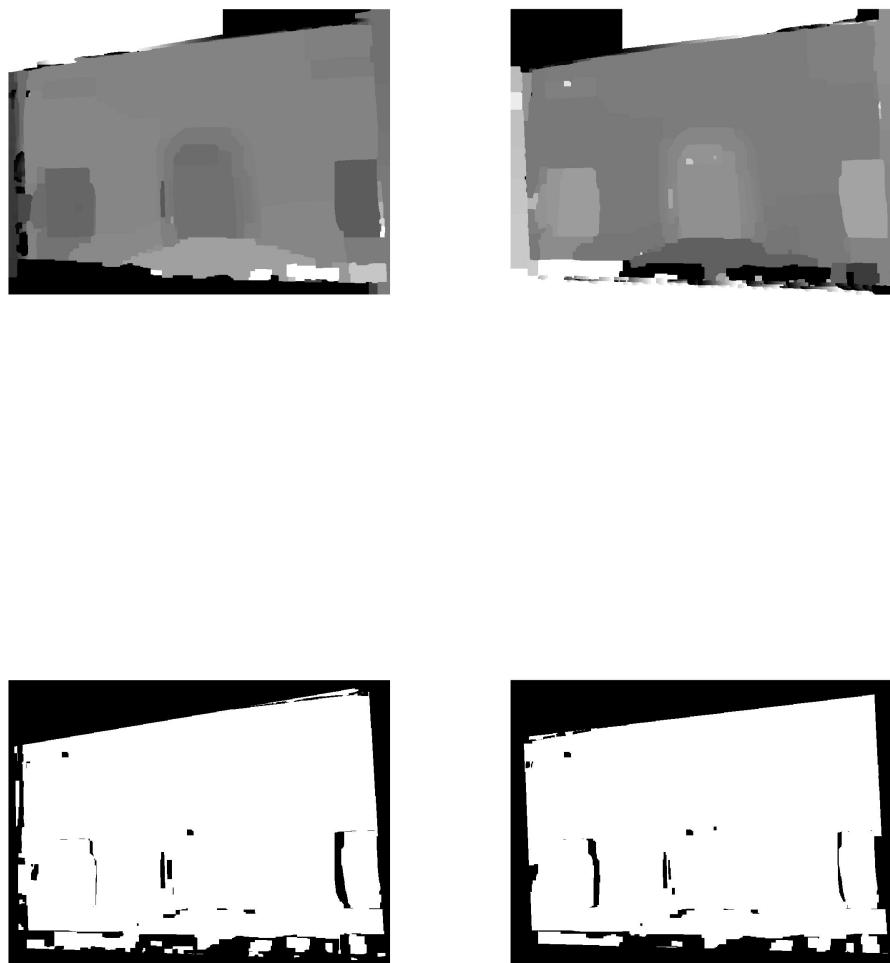


Figure 8:  $DC * 500$ ,  $SC * 5$

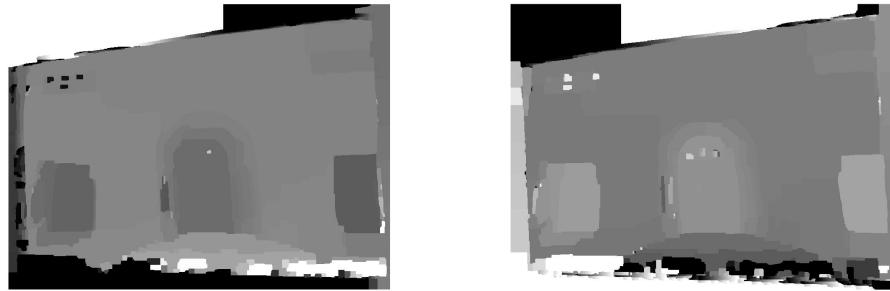


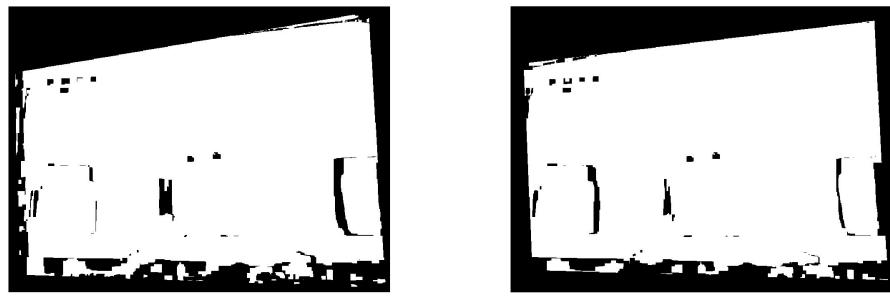
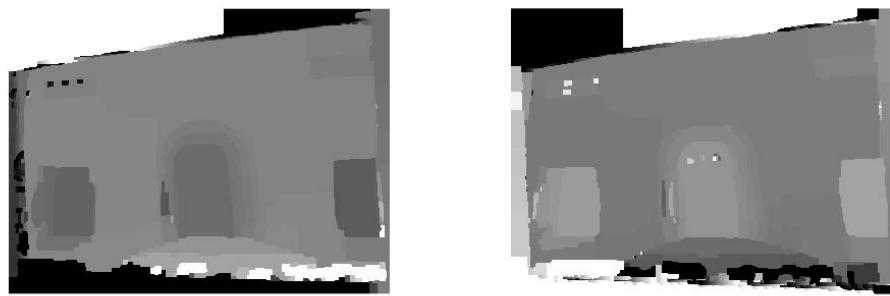
Figure 9:  $DC * 1000$ ,  $SC * 5$



Figure 10:  $DC * 2000$ ,  $SC * 5$

As can be seen in the images above a  $DC_{factor}$  of 500 performed best for producing a continuous image. No definitive conclusion about the accuracy can be made without the measurements of the building.

Here are a few of the tried results with variation of SC:



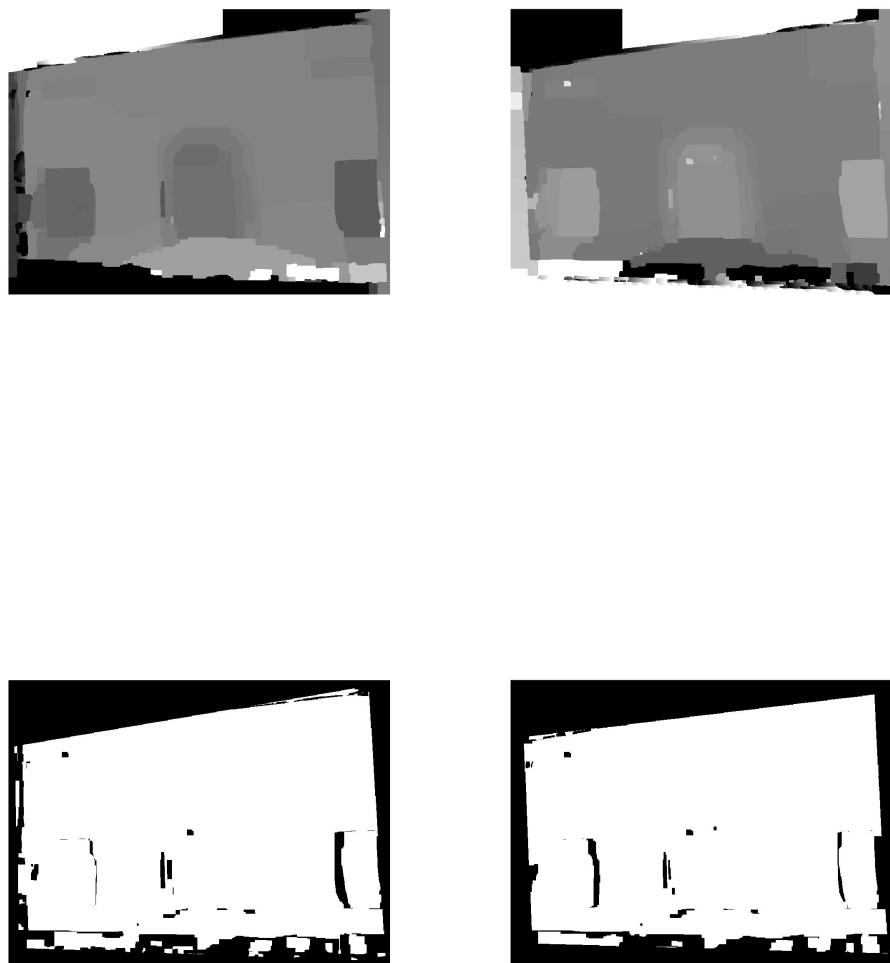


Figure 12:  $DC * 500$ ,  $SC * 5$

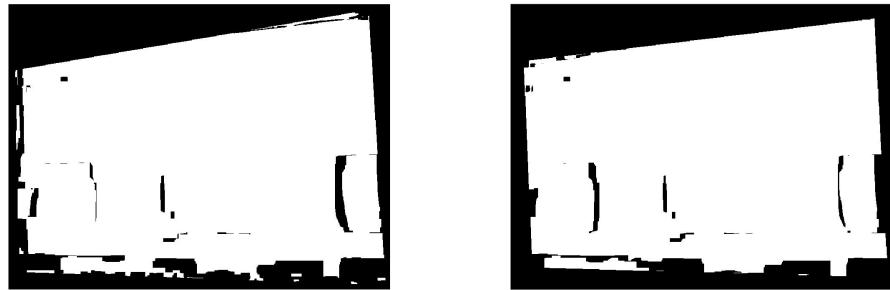
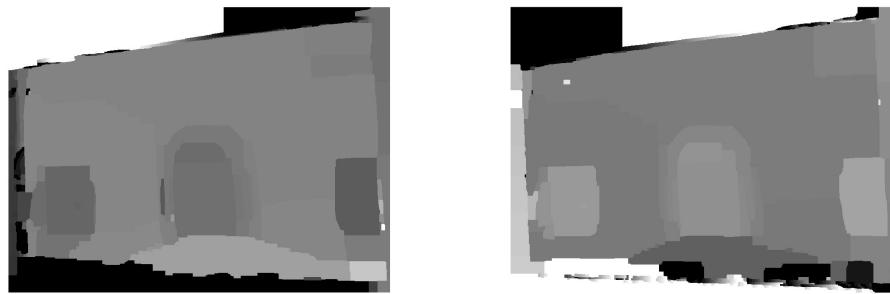


Figure 13:  $DC * 500$ ,  $SC * 7$

Here the SC factor of 7 gives the most correctly mapped points.



Figure 14: Textured 3D Model from Graph Cut with range  $[-40, 40]$  and filter Window Size 7

As can be seen this implementation works considerably better than the winner-takes-all stereo matching algorithm in smoothness. This time the flat surface of the wall is represented quite well and features such as the stair and the door are very well represented. Due to this algorithm being computationally more expensive finding the correct range is even more important.

### 3 Disparity Range Automation

To calculate an automatic disparity range an approach using 8-point RANSAC was chosen. This algorithm, as was seen in previous exercises, returns a large set of point correspondences of the two images. This dataset was used to calculate the average distance in  $x$  direction between matching points. From this we get a good measurement of how far the detected points, which are assumed to be points of interest, are apart. In manner of good engineering this average point distance is then multiplied by a safety factor (in this case chosen to be 2 in this case) and set as the upper and lower boundaries of the range. As can be seen in the pictures below, this performs really well. For the provided image pair this algorithm resulted in an average range of  $[-6, 6]$

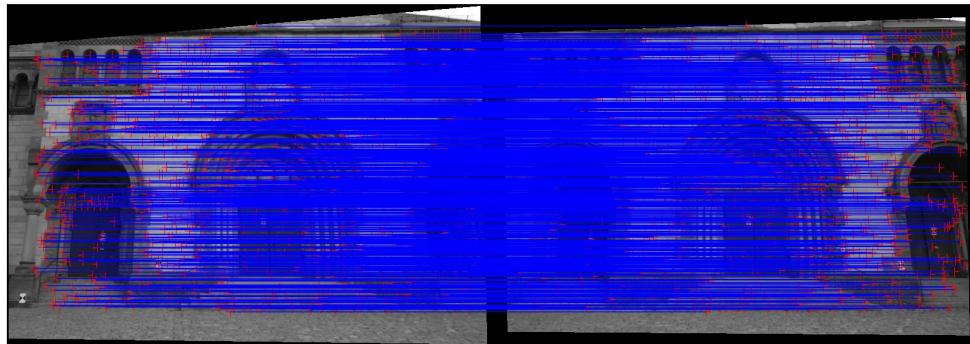


Figure 15: 8-Point RANSAC Matching of the two images

This results in a much lower computation time for both algorithms. It can be imagined that such a method would not perform well if the distance between corresponding points varies a lot. This would be the case if it was tried to compute a disparity map of an object that varies in orders of magnitude more in distance from the camera.

#### 3.1 Disparity Computation

For the disparity computation this range results in a way better performance in the amount of correctly mapped points, especially on the flat stone wall.

The smaller range also decreases the possible space for mismatches.

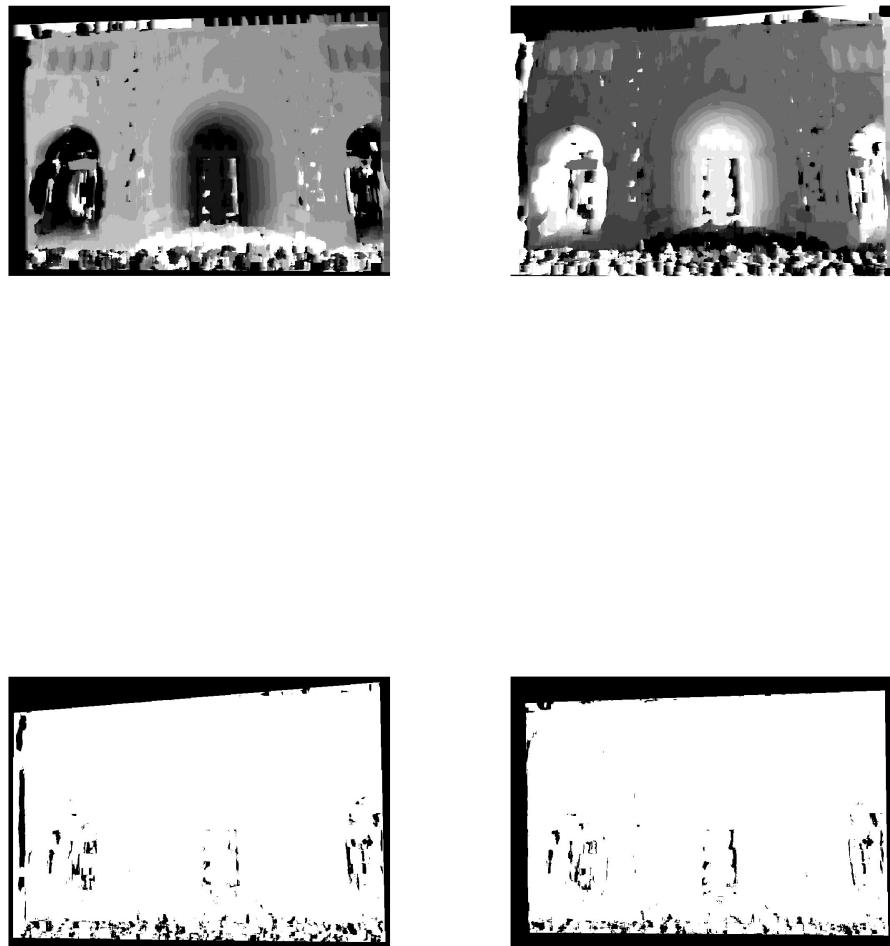


Figure 16: 19x19 Filter Window and range  $[-6, 6]$

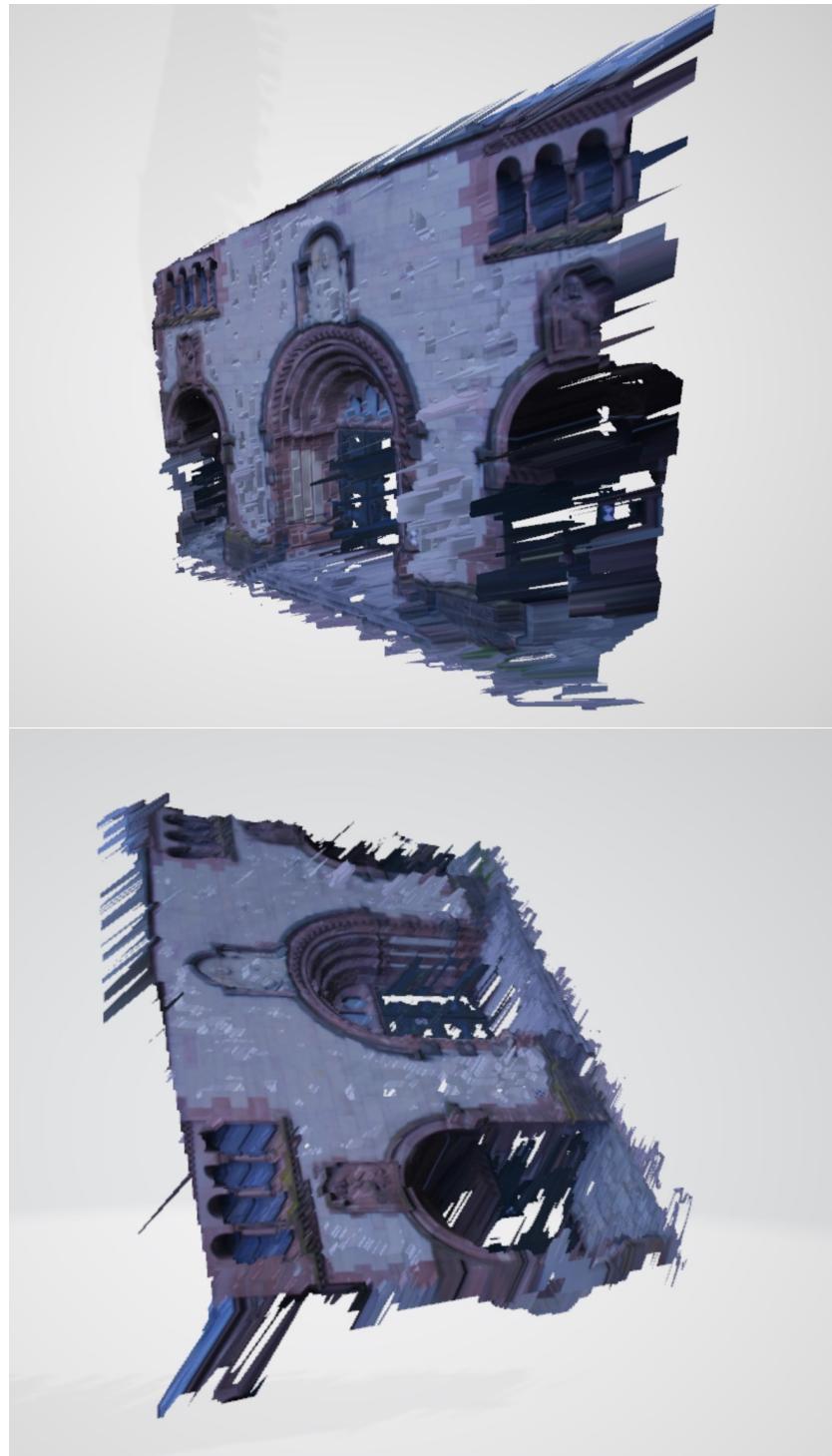


Figure 17: Corresponding 3D Model

### 3.2 Graph Cut

This model is looking great! Also on the graph cut algorithm the reduced range performs very well for the reasons already named.

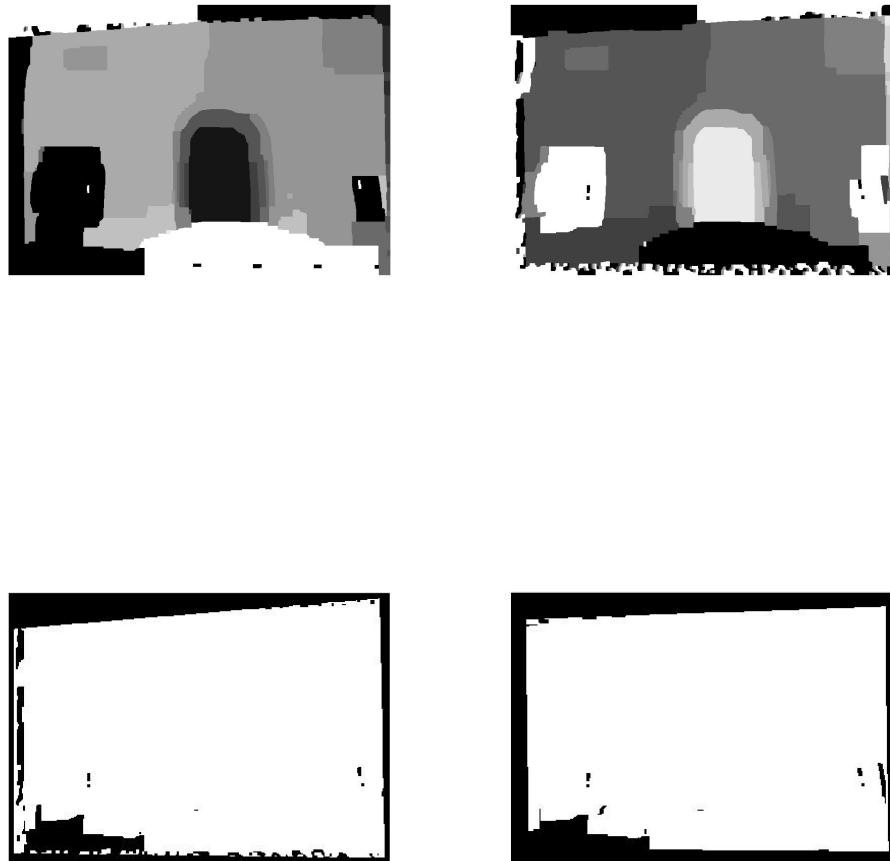


Figure 18:  $DC * 500$ ,  $SC * 7$  and range  $[-6, 6]$

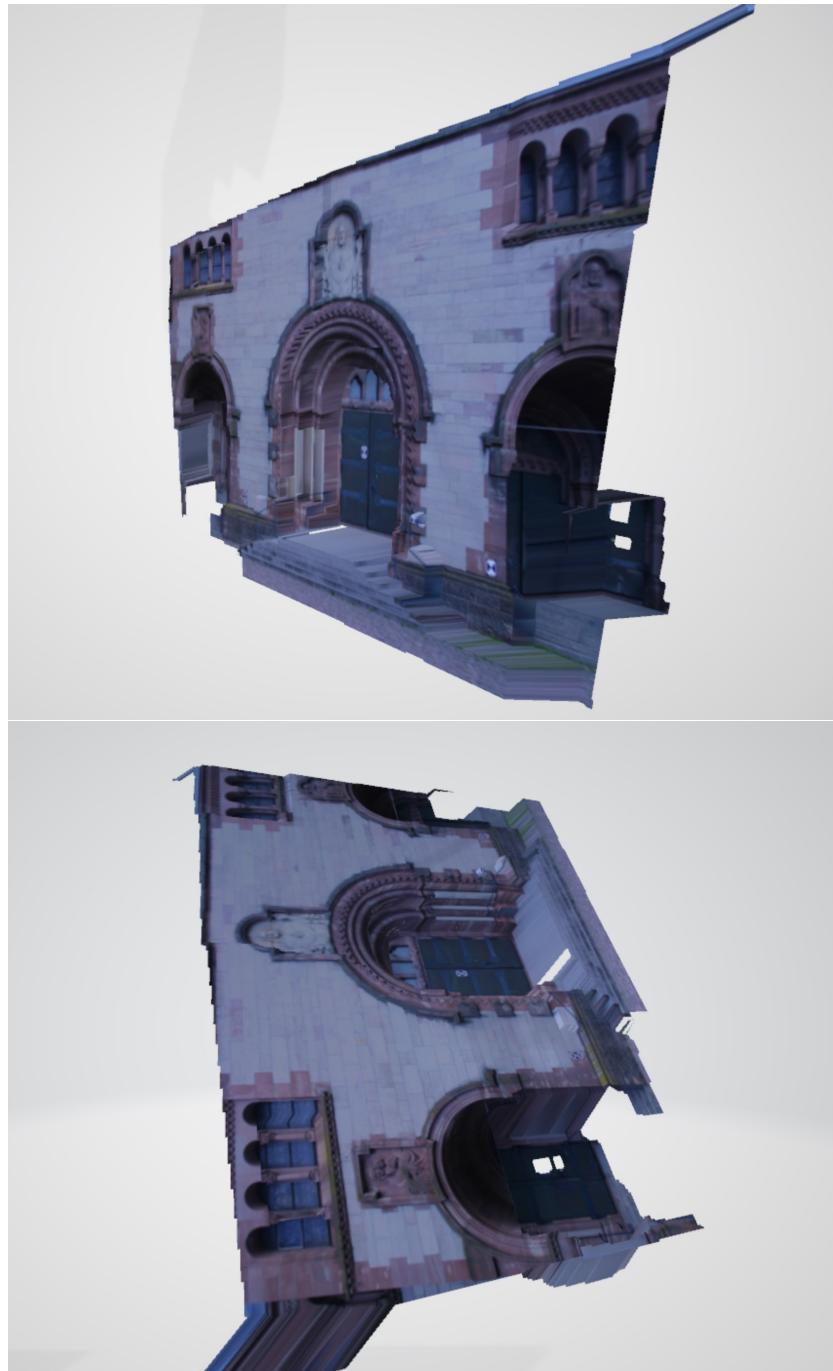


Figure 19: Corresponding 3D Model