Report of Anton Maksimov (antonma, 16-952-137), Task 6 "Model fitting"
on ETHZ course "Computer Vision".

1. We implement more efficient for MATLAB matrix version of SSD map calculation, smooth it using averaging filters — boxes of different sizes. Using this map we find the best dispairity for each pixel and save it.
We can see that the more is size of the filter, the more smooth is the picture and more connected areas are present.

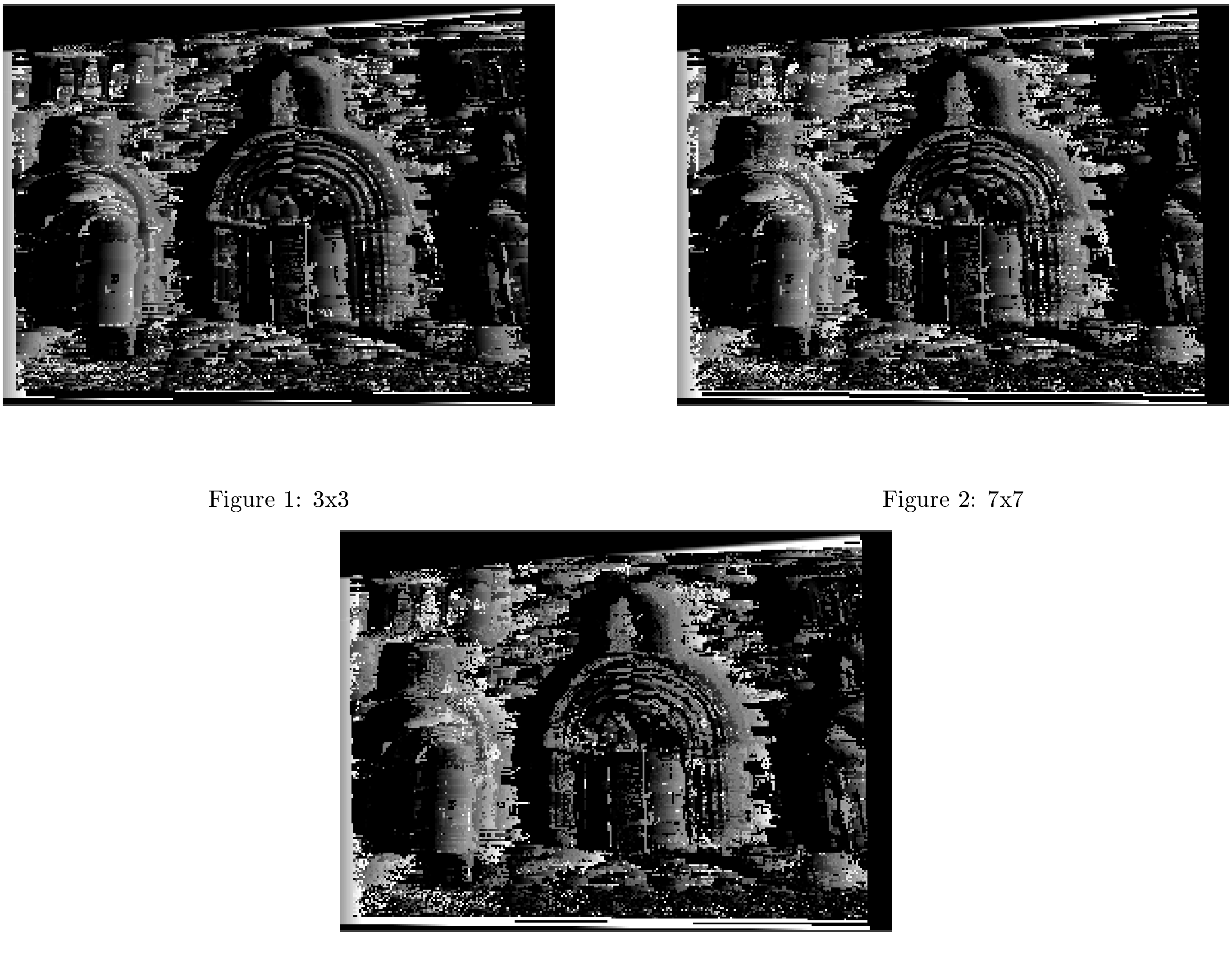

Figure 1: 3x3

Figure 2: 7x7

Figure 3: 12x12

Figure 4: Dispairity maps for «winner–takes–all» and different sizes of box filter.

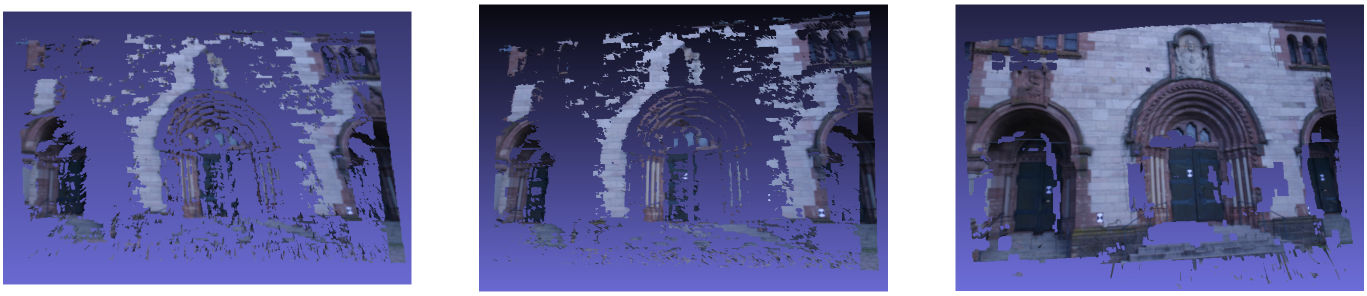

Figure 5: 3x3

Figure 6: 10x10

Figure 7: 20x20

Figure 8: 3D-reconstructions for «winner–takes–all» and different sizes of box filter..

2. We apply graph-cut algorithm for classification of pixels (labeling them with the best value of dispairity), where cost of labeling consists of price to label pixel, which is taken from correspondent SSD, and price of changing label relatively to neighboring pixels, which comes from gaussian filtering (i.e. considers local gradient of intensity). Graph-cut algorithm gives much better results than the previous algorithm, labeling is smoother and uniformer, it has problems only in places of fast change of deepness.
Size of averaging filter is connected with parameters of GraphCut (there are parameters which control smoothness), so we just use 3x3 filter here. In order to search for the best parameters, we change coefficients DataCost (second parameter) and SmoothnessCost (third parameter) of function Graph, choose the best. As the further improvement wee could after that change size and covariance of gaussian filter and choose the best from them.
The best looks with DataCost 100 and SmoothnessCost 2 coefficients.

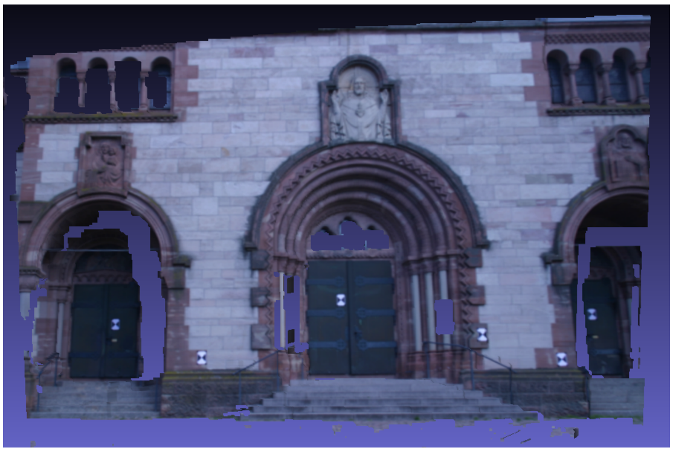

Figure 9: GraphCut. One of the results with coeff near DataCost 1000, near SmoothnessCost 5, gaussian filter size 3 and covariance 3.

The best looks with 100 and 2 coefficients.

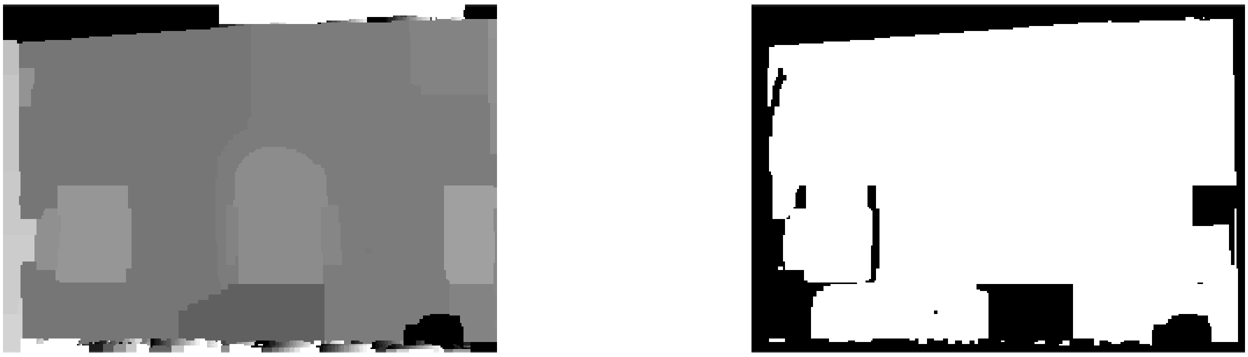

Figure 10: GraphCut. DataCost 100, SmoothnessCost 2, Gaussian size 3, covariance 3

3. In order to estimate range of disparities of the image we get disparities of points of interest from RANSAC on extracted by SIFT matches. We approximate range of disparities as maximal shift among this matching points along horizontal axis multiplied by 2 in order to make it more robust against existence of areas with no points of interest but bigger dispairity value.

In order to get enough matches we start from small threshold for Sampson distance of inliers in RANSAC and multiply it by 10 every step while we don't get 50 matches. So, this number of inliers is essentially the only parameter needed to be chosen in order to get range of dispairities.

We used function `fundamentalMatrixRANSAC`, but introduced `threshold` as a parameter, so we can iteratively change accuracy of search.

For given images we estimated maximal dispairity (multiplied by 2) as 25–30 during several launches. It workes faster as needs less iterations over dispairity range, quality is the same more or less.

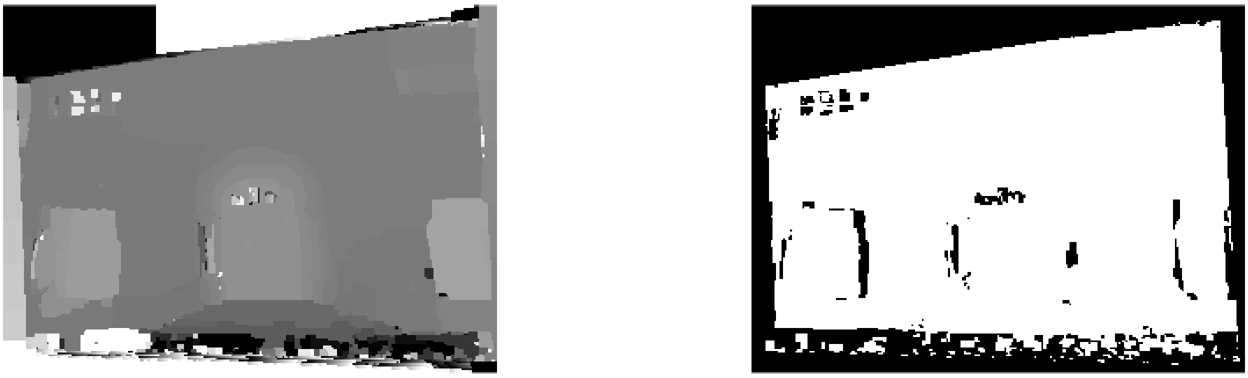

Figure 11: GraphCut. DataCost 500, SmoothnessCost 2, Gaussian size 3, covariance 3

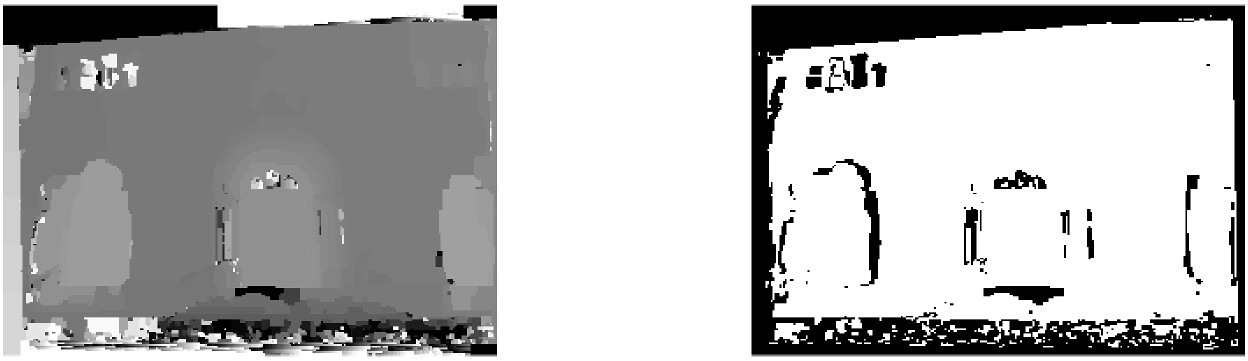

Figure 12: GraphCut. DataCost 1000, SmoothnessCost 2, Gaussian size 3, covariance 3

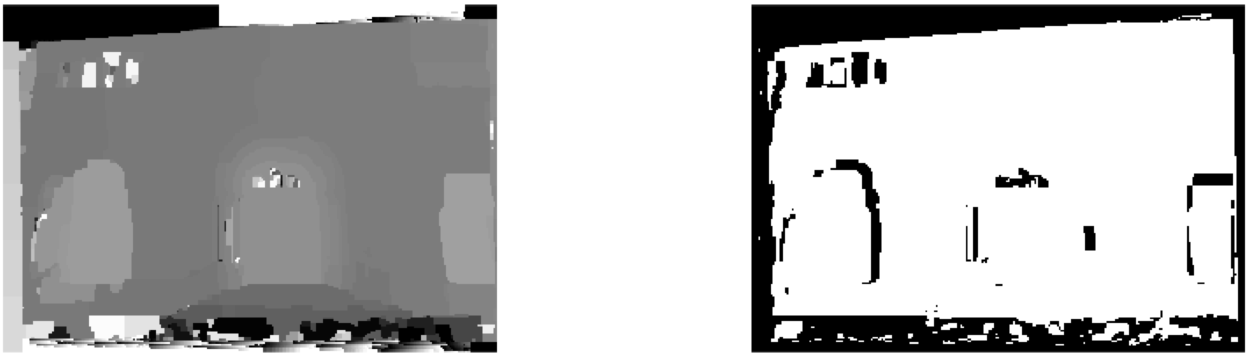

Figure 13: GraphCut. DataCost 1000, SmoothnessCost 5, Gaussian size 3, covariance 3

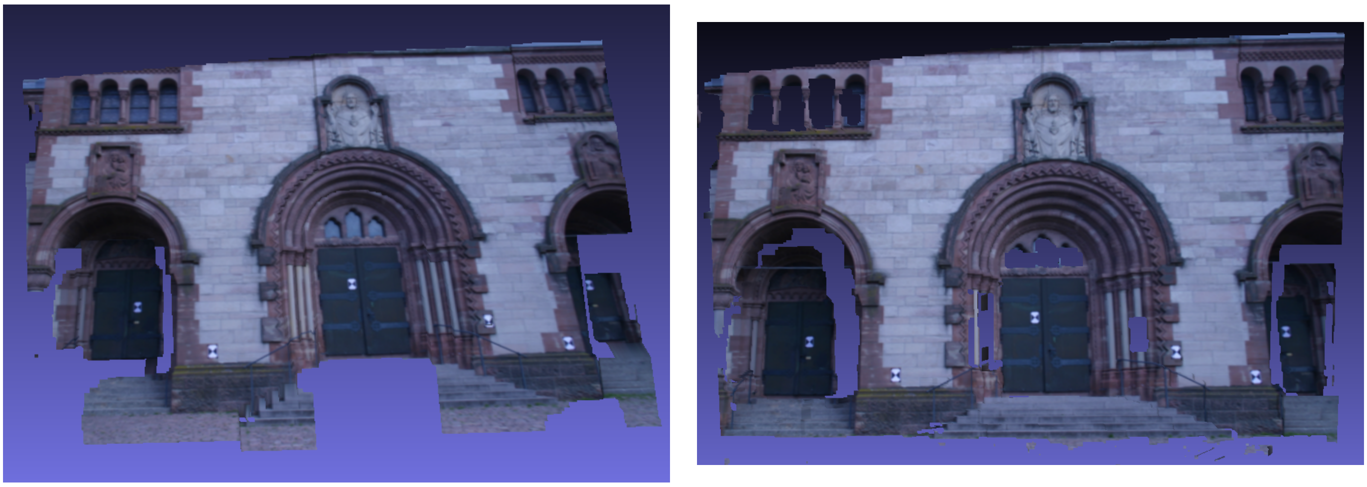

Figure 14: GraphCut. The best found combination of parameters improves quality (left — DataCost 100 and SmoothnessCost 2, right – 1000 and 5).

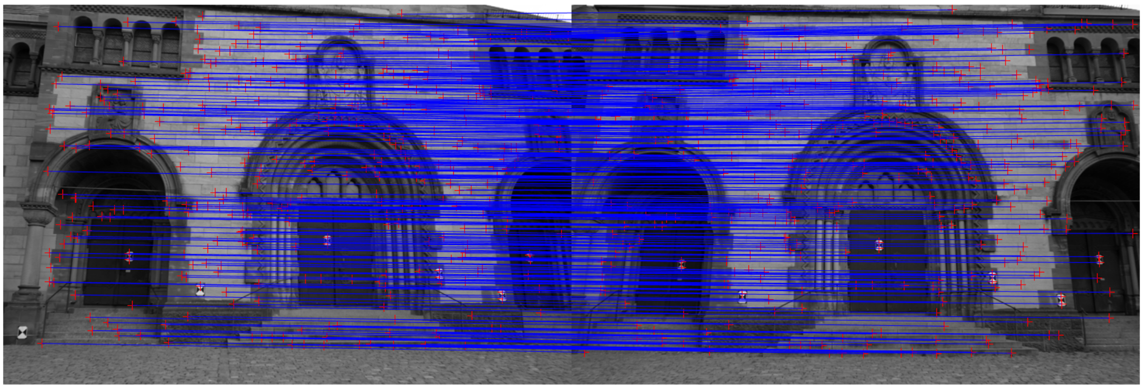

Figure 15: Matches obtained by RANSAC from which we estimate the range of search of dispairities.