188.513 Stereo Vision (VU 2,0)
2021S

Group 4
Marko Kadic, 12045128
Fabian Pechstein, 00726104

June 2021

The goal of this assignment was to implement four sub-tasks based on a provided paper dealing with stereo matching by implementing a guided filter by Hosni et al. [2].

1 Simple Cost Volume Computation

To find the right disparity between two rectified images a cost volume is computed. Whereas the cost for pixel p at disparity d is defined as:

$$C(p,d) = \sum_{p \in N_p} \sum_{c} |L_c(q) - R_c(q-d)| \tag{1}$$

with a correlation window of size n placed over pixel p, c represents the colour channels. Costs should for a pixel should reach a local minimum at the right disparity, given the case that neither occlusions or texture-less regions play a hand. Disparity labels are thus selected with $min_{d \in D}(C(p,d))$, and our final depth map is smoothed through an average filter of size n_a.

2 Cost Volume With Guided Filter

We can see in figure 1 that our initial solution has lot of issues around disparity borders, texture-less regions, as well as regions with repeating patterns (e.g. the book shelve).
To improve on this, Hosni et. al proposed the use of a guided filter, which should preserve disparity borders. We have made use of Matlab's implementation of *imguidedfilter* [1] with the additional parameters for filter-window size (n_s) as well as smoothing factor. The latter has been found to have little influence during our experiments (see figures 5 and 6), thus we fixed the smoothing factor to 0.5.

[1] `https://de.mathworks.com/help/images/ref/imguidedfilter.html`

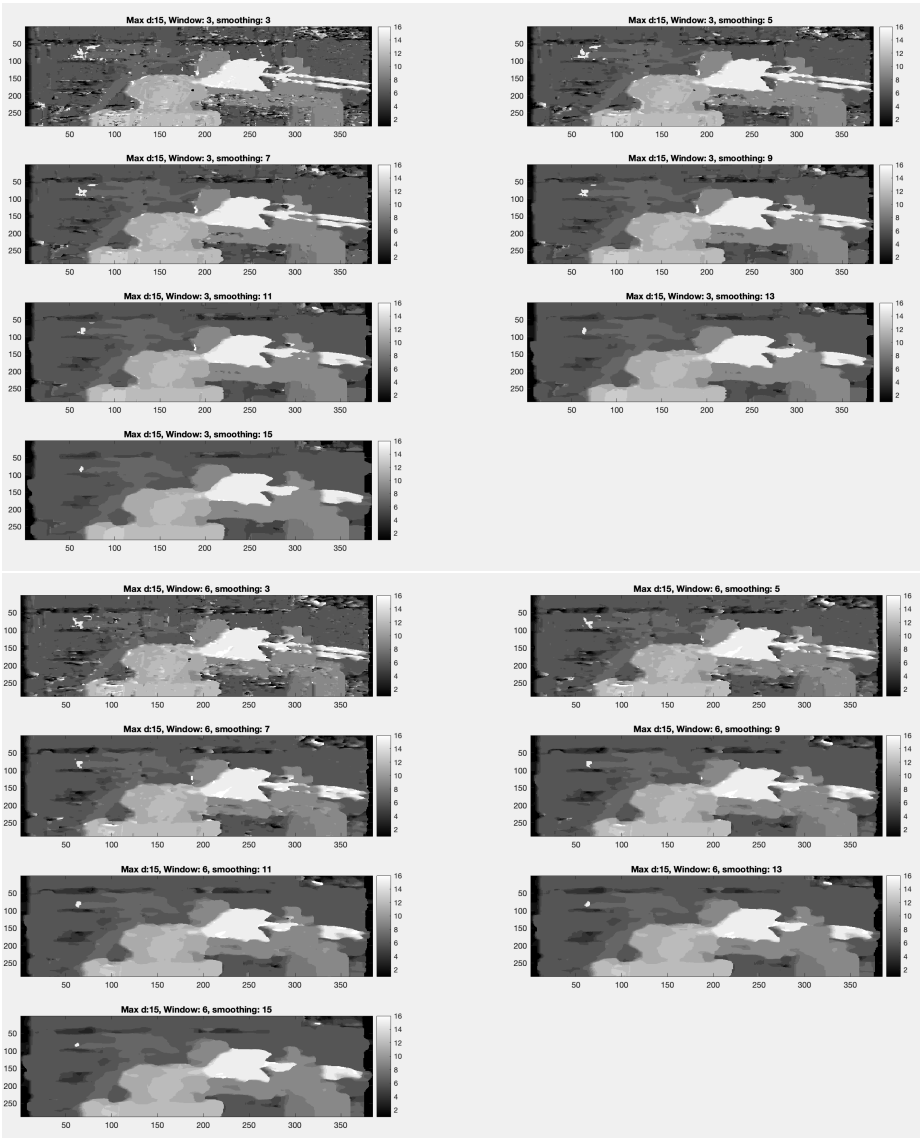

Figure 1: Various parameter combinations to show the effect of average smoothing.

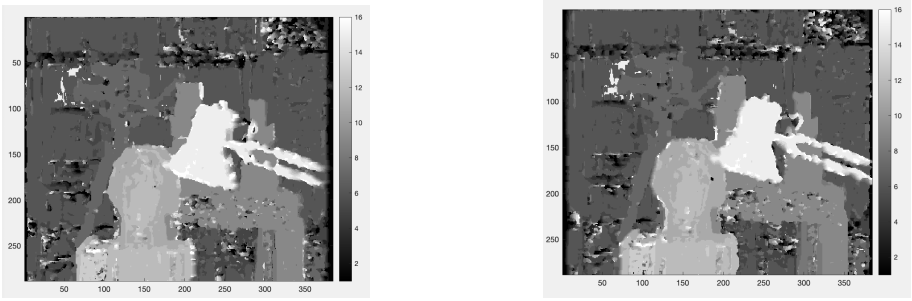

Figure 2: Our results for task 1 with implementing an average filter for the cost volume.

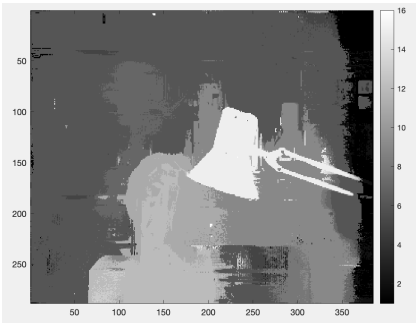

Figure 3: left disparity map for for $n_a = 3$, $n_s = 42$

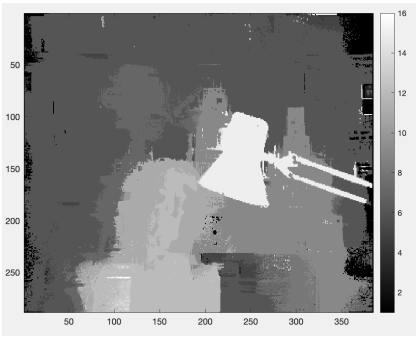

Figure 4: right disparity map for $n_a = 3$, $n_s = 42$

We can see that smaller values for n_a tend to add more noise, whereas larger values for n_s help to preserve disparity borders (e.g. visible at the lamp in figures 5 and 6). We have added respective scripts to test our parameter configuration for tasks 1 and 2 see scripts with the post-fix *_parameter_selection* in our submitted code zip file.

Figures 3 and 4 show our resulting disparity maps with our selected parameters:

$$n_a = 3 \quad n_s = 42 \quad n_s = 0.5 \tag{2}$$

3 Occlusion Detection and Filling

Occlusion detection has been implemented as suggested by Hosni et al. [2] through a left–right consistency check. Pixels that fail said check are marked as invalid and are later filled with their next (left or right) valid neighbouring pixel. Figures 7 and 8 show our results. Noteworthy in this regard is that the leftmost

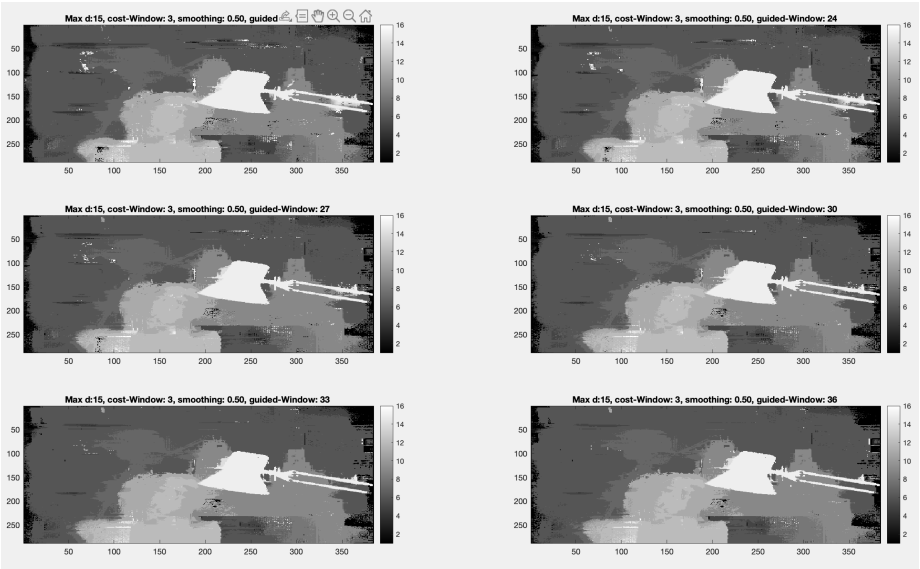

Figure 5: various parameters at $n_a = 3$

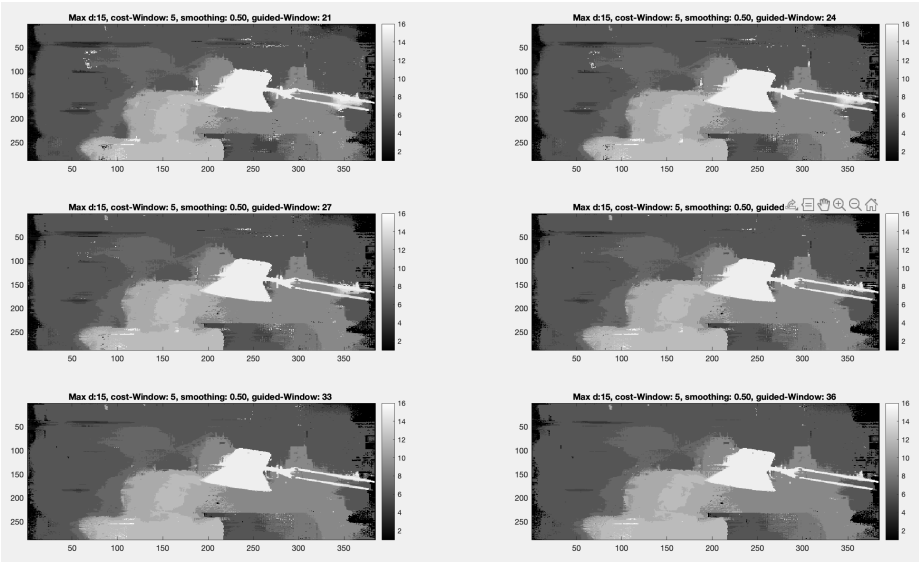

Figure 6: various parameters at $n_a = 5$

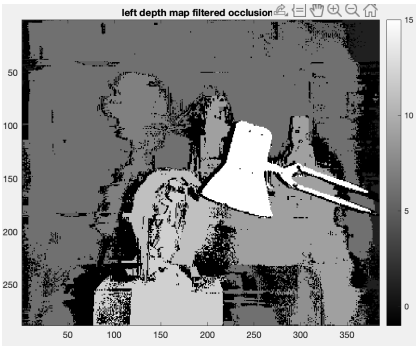

Figure 7: left disparity map with filtered occlusions

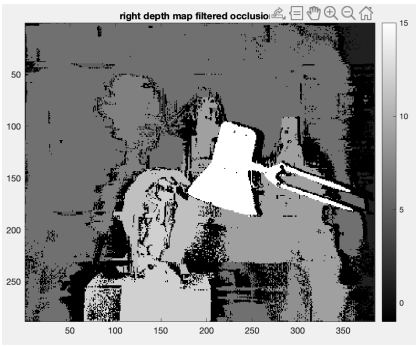

Figure 8: left disparity map with filtered occlusions

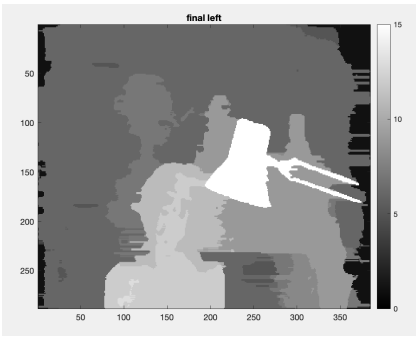

Figure 9: our final left disparity map

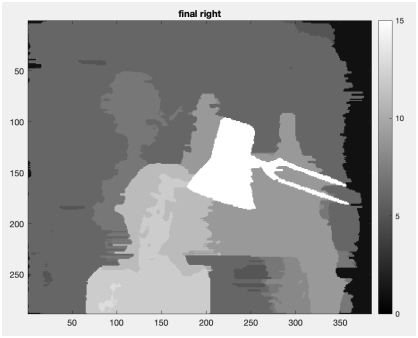

Figure 10: our final right disparity map

and rightmost image border pixels are starting to cause troubles from here on, which will become obvious when taking a look at our filled and smoothed results (figure 9 and 10). Unfortunately we didn't manage to solve the occurrence of filling artefacts in time for the submission.

4 Evaluation

We have chosen to test our implementation with the remaining test images from the Middlebury 2001 Stereo Dataset [1] [2]. Thus we need to evaluate our calculated depth maps against the provided ground truth data. As an error metric we have chosen the average error for left and right disparity maps ($|groundtruth - result|$).

Given the fact that our implementation has issues with the border pixels as well as can't really deal with slanted surface in the images, our results are

[2] `https://vision.middlebury.edu/stereo/data/scenes2001/`

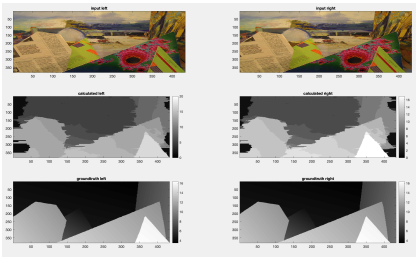

Figure 11: **Barn1:** interpreted using guided filter

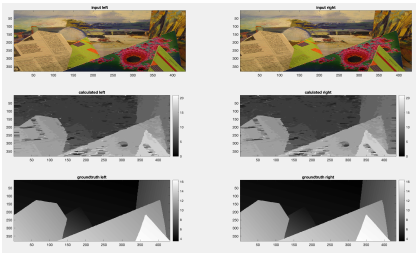

Figure 12: **Barn1:** interpreted using average filter

naturally bad in comparison to current state-of-the-art approaches. Even in comparison with the simpler approach described in section 1, we can see that the despite being more noisy and vulnerable to texture-less regions the average absolute error is still smaller for the test images (see table 1). One explanation that would come to mind, would perhaps be that the parameters that we have deducted from task 2 would not be suitable for images other than **tsukuba**, as well as our issues with left and right most border pixels that are influenced by the size of the guided filter window. Additionally, we we are dealing with varying image sizes, making the choice of our fixed pixel-parameters questionable in this case and would explain the bad performance.

One positive aspect that can be observed in figures 13 and 14 is that the guided filter is less vulnerable to texture-less regions.

image set	error with guided filter	error with average filter
barn1	248381.69	**201879.69**
barn2	286777.25	**241704.62**
bull	256979.50	**176322.50**
sawtooth	336673.31	**195218.06**
venus	336774.50	**263334.12**

Table 1: Absolute average error comparison between the approaches described in sections 1 and 2 respectively.

5 Middlebury Stereo Evaluation - Version 3

We have executed the provided Matlab SDK with our implementation using the *training* set in quarter resolution, see table 2 for the preliminary results. Figure 15 shows our terrible ranking. There would be definitely room for improvement.

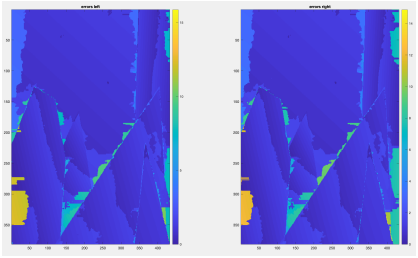

Figure 13: **Barn1:** errors with guided filter

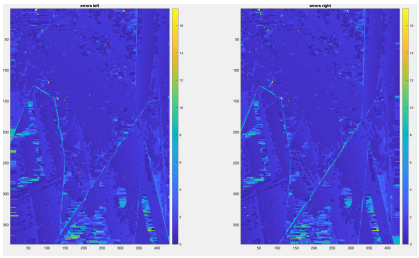

Figure 14: **Barn1:** errors with average filter

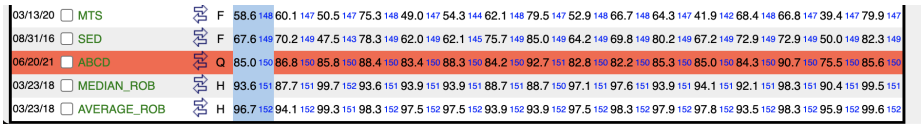

Figure 15: the benchmark results are, as expected with the perceived errors, quite bad.

Adirondack	0.688189
ArtL	0.736085
Jadeplant	0.739812
Motorcycle	0.616852
MotorcycleE	0.780856
Piano	0.676062
PianoL	0.861887
Pipes	0.597745
Playroom	0.649329
Playtable	0.713052
PlaytableP	0.705214
Recycle	0.665096
Shelves	0.767101
Teddy	0.489470
Vintage	0.695076
Overall	0.683088

Table 2: Results of Middlebury Benchmark v3

References

[1] Daniel Scharstein and Richard Szeliski. "High-accuracy stereo depth maps using structured light". In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 1. IEEE. 2003, pp. I–I.

[2] Asmaa Hosni et al. "Real-time local stereo matching using guided image filtering". In: *2011 IEEE International Conference on Multimedia and Expo.* IEEE. 2011, pp. 1–6.