Supplementary Material for ECCV 2012 Paper: Extracting 3D Scene-consistent Object Proposals and Depth from Stereo Images

Michael Bleyer[1⋆], Christoph Rhemann[1⋆,2], and Carsten Rother[2]

[1] Vienna University of Technology, Vienna, Austria
[2] Microsoft Research Cambridge, Cambridge, UK

Abstract. This document accompanies the paper: "Extracting 3D Scene-consistent Object Proposals and Depth from Stereo Images". We provide implementation details and results of a competing object segmentation method [1]. We also show our results on the Middlebury Stereo Benchmark [2]. Note that information given in this document is not necessary to understand the content of the main paper.

The Match Measure In the definition of the photo consistency term of sec. 2 (Model), we have described the match measure as being a weighted sum of gradient and color differences. Let us now formulate this match measure. Note that our photo consistency term (which includes the match measure) is identical to that described in the Patch- Match Stereo paper [3] where the following information can be found as well.

The function $\phi(q, q')$ of eq. (3) computes the pixel dissimilarity between a pixel q of the left and a pixel q' of the right image as

$$\phi(q, q') = (1 - \alpha) \cdot \min(||I_q - I_{q'}||, \tau_{col}) + \alpha \cdot (||\Delta I_q - \Delta I_{q'}||, \tau_{grad}). \quad (1)$$

Here, $||I_q - I_{q'}||$ denotes the L1-distance of colors of q and q' in RGB space and $||\Delta I_q - \Delta I_{q'}||$ represents the absolute difference of gray-value gradients. By using the gradient we can handle small radiometric differences that occur in left and right images (e.g., one image is slightly darker than the other). We truncate the pixel dissimilarities using parameters τ_{col} and τ_{grad}. This truncation limits the influence of occluded pixels in the cost aggregation procedure.The parameters are set as described in [3], i.e., $\{\alpha, \tau_{col}, \tau_{grad}\} := \{0.9, 10, 2\}$.

Depth Segmentation Algorithm In the Object Image Computation step of sec. 3 (Optimization), we have described a depth segmentation algorithm. We now explain this method in more detail.

As described in the paper, we start by applying a meanshift color segmentation algorithm [4] on the left input image. We then fit a disparity plane to each color segment using our initial disparity map F'. We now extract groups of segments that can be well modeled using the same disparity plane via applying an energy minimization approach that is explained as follows.

We first record all planes that have been computed in the plane fitting process. Our goal is to assign each color segment to one of these planes such that an

⋆ This work was supported in part by the Vienna Science and Technology Fund (WWTF) under project ICT08-019.

Fig. 1. Object segmentation result of "Blocks World" [1] on one of our images.

energy is minimized. This energy consists of a data term and a smoothness term. For each pixel of the left image, the data term measures the absolute difference between the point's disparity according to its assigned plane and its disparity in the initial disparity map F'. The smoothness term puts a constant penalty on spatial neighboring pixels assigned to different planes (Potts model). Note that there is a parameter λ that balances data and smoothness terms. To approximate the energy minimum, we apply alpha-expansions of all planes present in the original solution, i.e., the one obtained after plane fitting. Note that the computational complexity of this step is relatively low, as the energy can be optimized on a segment level. (Nodes in the graph correspond to whole segments.) After running three iterations of the alpha-expansion algorithm we obtain our depth segments by grouping all color segments that are assigned to the same disparity plane in the optimized solution. λ represents the parameter that we vary to obtain depth segmentations of different granularities such as shown in fig. 4 of the paper.

Results of "Blocks World" [1] As stated in the paper, "Blocks World" [1] can be regarded as a competing object segmentation method, as this algorithm gives a mapping of image pixels to one of seven different classes. We experienced that the publicly available code of [1] gives only very coarse segmentations on our test images, which are clearly inferior to our result. An example result for the "Parade" test set is shown in fig. 1.

Middlebury Results We show the results on all four Middlebury images in fig. 2. Fig. 3 shows our ranking in the Middlebury Online Table [2]. Our method takes rank 13 out of 117 algorithms. It also performs better than our reimplementation of [3] (see fig. 3).

Generality We train object stereo [5], our reimplementation of PatchMatch stereo [3] as well as our algorithm on the Middlebury evaluation set shown in fig. 2. We then apply the parameters that gave the highest Middlebury ranking for computation of the 2005 test set. Quantitative results are shown in tab. 1. Our algorithm achieves the lowest error percentage on 3 of 6 images (bold numbers in tab. 1). Fig. 4 shows the corresponding disparity and error maps.

Individual terms of the energy function Tab. 2 shows the contribution off individual terms to the quality of disparity maps. Here, we test our method with the same parameters as in the previous experiment (= "All Terms On" in tab. 2). "Gravity Off" means that we set $\lambda_{gravity} := 0$, while the other parameters are set to the values of "All Terms On". This disables the gravity constraint.

	Art	Books	Dolls	Laundry	Moebius	Reindeer	Avg. Error
Object Stereo	**6,71**	13,14	11,37	15,50	11,54	7,17	10,90
PM Stereo	9,11	9,31	**5,16**	12,53	9,51	**4,79**	8,40
Ours	8,36	**7,68**	5,17	**11,76**	**9,25**	5,16	**7,90**

Table 1. Generality of our approach. We plot the percentage of pixels having a disparity error > 1 pixel in unoccluded regions (black pixels in fig. 4). Our method performs better than object stereo [5] and PatchMatch stereo [3] on 3 of 6 images and achieves the lowest average error percentage.

	Art	Books	Dolls	Laundry	Moebius	Reindeer	Avg. Error
All Terms On	8,36	7,68	5,17	11,76	9,25	5,16	7,90
Gravity Off	8,30	8,20	5,20	11,84	9,51	5,70	8,30
Intersection Off	7,63	8,08	4,86	12,18	9,35	5,34	7,91
Tightness Off	8,00	8,10	5,26	12,15	9,56	6,19	8,21

Table 2. Influence of physics-based terms of our energy. We plot the error percentage in unoccluded regions (black pixels in fig. 5). Red numbers indicate a lower error percentage in comparison to "All Terms On".

"Intersection Off" means that we set $\lambda_{intersect} := 0$ to disable the intersection constraint. All other parameters are set to the values of "All Terms On". We finally disable the bounding box tightness constraint by setting $\lambda_{tight} := 0$ (= "Tightness Off"). Switching off individual terms leads to higher error percentages, in general. Corresponding disparity and error maps are shown in fig. 5.

References

1. Gupta, A., Efros, A., Hebert, M.: Blocks world revisited: Image understanding using qualitative geometry and mechanics. In: ECCV. (2010)
2. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV **47** (2002. http://vision.middlebury.edu/stereo/) 7–42
3. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo - stereo matching with slanted support windows. In: BMVC. (2011)
4. Christoudias, C., Georgescu, B., Meer, P.: Synergism in low-level vision. In: ICPR. Volume 4. (2002) 150–155
5. Bleyer, M., Rother, C., Kohli, P., Scharstein, D., Sinha, S.: Object stereo - joint stereo matching and object segmentation. (In: CVPR '11)

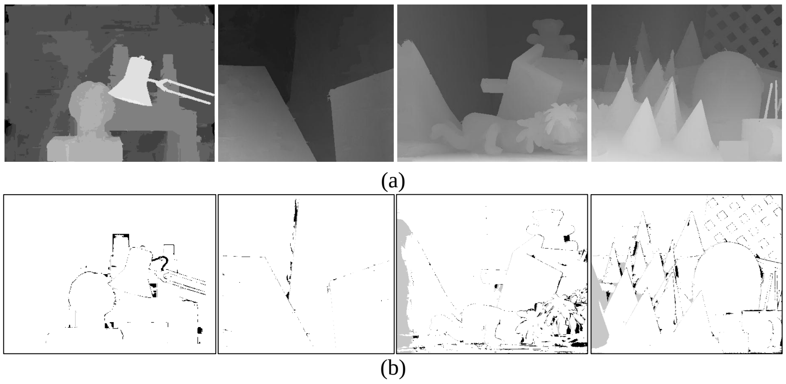

Fig. 2. Results on the Middlebury set. (a) Disparity maps. (b) Disparity errors > 1 pixel.

Error Threshold = 1		Sort by nonocc						Sort by all				Sort by disc	
Algorithm	Avg. Rank	Tsukuba nonocc	Tsukuba all	Tsukuba disc	Venus nonocc	Venus all	Venus disc	Teddy nonocc	Teddy all	Teddy disc	Cones nonocc	Cones all	Cones disc	Average Percent Bad Pixels
ADCensus [94]	7.3	1.07 14	1.48 11	5.73 16	0.09 2	0.25 7	1.15 3	4.10 6	6.22 3	10.9 6	2.42 6	7.25 6	6.95 7	3.97
CoopRegion [41]	8.7	0.87 3	**1.16** 1	4.61 2	0.11 4	0.21 3	1.54 7	5.16 16	8.31 12	13.0 13	2.79 17	7.18 5	8.01 21	4.41
AdaptingBP [17]	8.8	1.11 17	1.37 6	5.79 17	0.10 3	0.21 4	1.44 5	4.22 8	7.06 6	11.8 9	2.48 8	7.92 12	7.32 11	4.23
RVbased [116]	11.3	0.95 8	1.42 9	4.98 7	0.11 6	0.29 11	**1.07** 1	5.98 21	11.6 30	15.4 25	2.35 4	7.61 7	6.81 6	4.88
DoubleBP [35]	11.8	0.88 5	1.29 3	4.76 5	0.13 8	0.45 20	1.87 12	3.53 5	8.30 11	9.63 4	2.90 21	8.78 29	7.79 18	4.19
RDP [102]	12.3	0.97 9	1.39 7	5.00 8	0.21 23	0.38 17	1.89 13	4.84 10	9.94 19	12.6 11	2.53 9	7.69 9	7.38 12	4.57
OutlierConf [42]	12.7	0.88 4	1.43 10	4.74 4	0.18 16	0.26 9	2.40 22	5.01 12	9.12 16	12.8 12	2.78 16	8.57 23	6.99 8	4.60
SubPixDoubleBP [30]	17.1	1.24 25	1.76 28	5.98 21	0.12 7	0.46 22	1.74 10	3.45 4	8.38 13	10.0 5	2.93 23	8.73 27	7.91 20	4.39
SurfaceStereo [79]	17.3	1.28 30	1.65 19	6.78 35	0.19 18	0.28 10	2.61 30	3.12 3	**5.10** 1	**8.65** 1	2.89 20	7.95 14	8.26 27	4.06
WarpMat [55]	19.7	1.16 18	1.35 5	6.04 22	0.18 17	0.24 6	2.44 25	5.02 13	9.30 17	13.0 15	3.49 35	8.47 22	9.01 41	4.98
ObjectStereo [98]	20.3	1.22 24	1.62 15	6.36 27	0.59 55	0.69 39	4.61 56	4.13 7	7.59 7	11.2 8	**2.20** 1	6.99 4	**6.36** 1	4.46
PatchMatch [112]	24.3	2.09 66	2.33 52	9.31 64	0.21 22	0.39 18	2.62 31	2.99 2	8.16 8	9.62 3	2.47 7	7.80 10	7.11 9	4.59
Undr+OvrSeg [48]	25.8	1.89 58	2.22 51	7.22 44	0.11 5	0.22 5	1.34 4	6.51 31	9.98 20	16.4 36	2.92 22	8.00 15	7.90 19	5.39
GC+SegmBorder [57]	25.8	1.47 43	1.82 30	7.86 55	0.19 19	0.31 12	2.44 25	4.25 9	5.55 2	10.9 7	4.99 73	**5.78** 1	8.66 34	4.52
Ours	25.8	1.60 48	1.84 34	8.17 57	0.50 50	0.76 41	5.28 60	2.77 1	8.17 9	8.75 2	2.30 2	6.89 3	6.68 3	4.48
InfoPermeable [109]	26.7	1.06 13	1.53 12	5.64 13	0.32 35	0.88 50	4.15 50	5.60 18	13.0 45	14.5 17	2.65 13	9.16 37	7.69 17	5.51
PatchMatch Reimpl.	27.3	1.66 49	1.92 39	8.30 58	0.50 50	0.84 45	4.76 58	**2.87** 1	8.05 8	**8.43** 1	2.44 6	7.05 4	7.08 8	4.49
CostFilter [95]	27.5	1.51 46	1.85 36	7.61 50	0.20 21	0.39 19	2.42 23	6.16 25	11.8 33	16.0 29	2.71 14	8.24 18	7.66 16	5.55
GlobalGCP [104]	27.8	0.87 2	2.54 55	4.69 3	0.16 14	0.53 26	2.22 20	6.44 28	11.5 29	16.2 32	3.59 38	9.49 46	8.95 40	5.60

Fig. 3. The ranking of our method in the Middlebury table. Our algorithm takes ranks 1 and 2 on the complex Teddy and Cones images according to the error in non-occluded pixels.

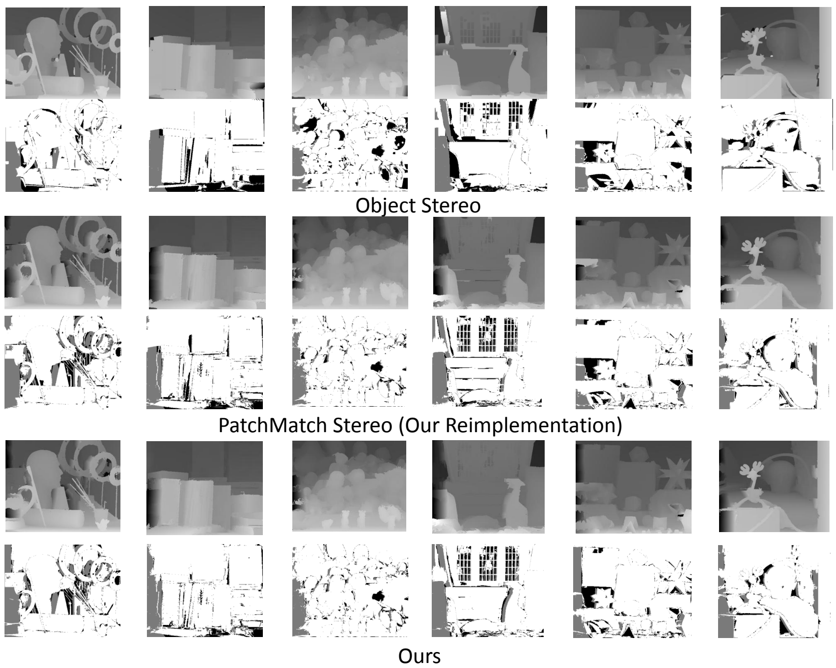

Fig. 4. Corresponding disparity and error maps for the experiment of tab. 1.

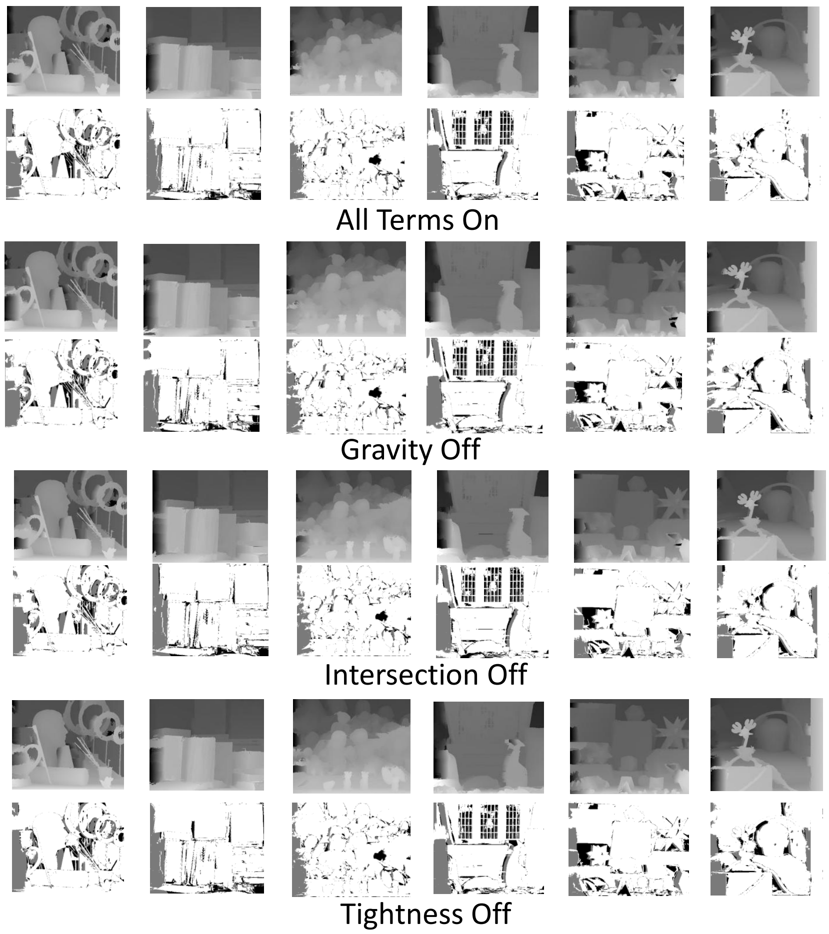

Fig. 5. Corresponding disparity and error maps for the experiment of tab. 2.