Evaluation and comparison report

Tuning λ parameter

We first evaluate the graph-cut (GC) algorithm results obtained for different values of the λ (smoothness) factor (Figure 1). We used the default and params images from the fourth homework.

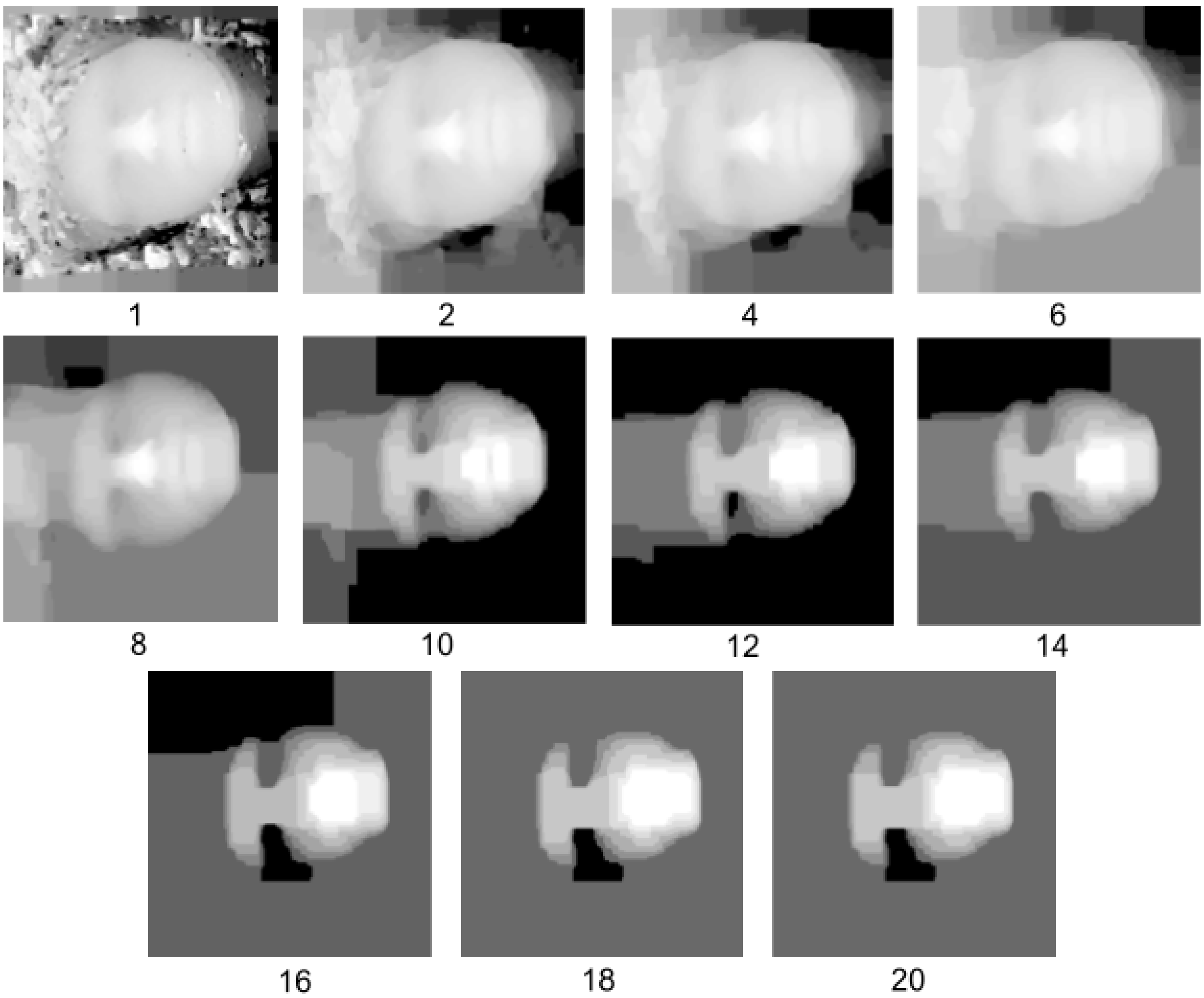

Figure 1: Comparison of disparity maps computed on default images from the 4th exercise for different values of the λ parameter. The number below each image represents the used λ value. Invalid disparities near borders are not shown.

Parameter λ regularizes the variation between neighboring pixels. Based on the provided results, we see that as λ increases, the disparity maps become smoother and neighboring pixels/regions tend to merge into similar and less distinct disparity values. Eventually, this leads to larger homogeneous surfaces with fewer finer details. On the other hand, smaller λ values allow greater variation, resulting in much noisier disparity maps, with more fragmented structures and stronger fluctuations. Both extremely large and low λ values are undesirable, while moderate ones (e.g., 2/4) provide a better balance of smoothness and details. This will be further demonstrated in the following sections.

Next, we show the results for the case where λ=0, using the default images from the previous homework. When λ is set to 0, there is no control over the variation between neighboring pixels. As shown in Figure 2, the resulting disparity map from the GC algorithm is very similar (i.e., almost identical) to that of the region-growing algorithm. Since region growing does not use any smoothness regularization, such similarity is expected.

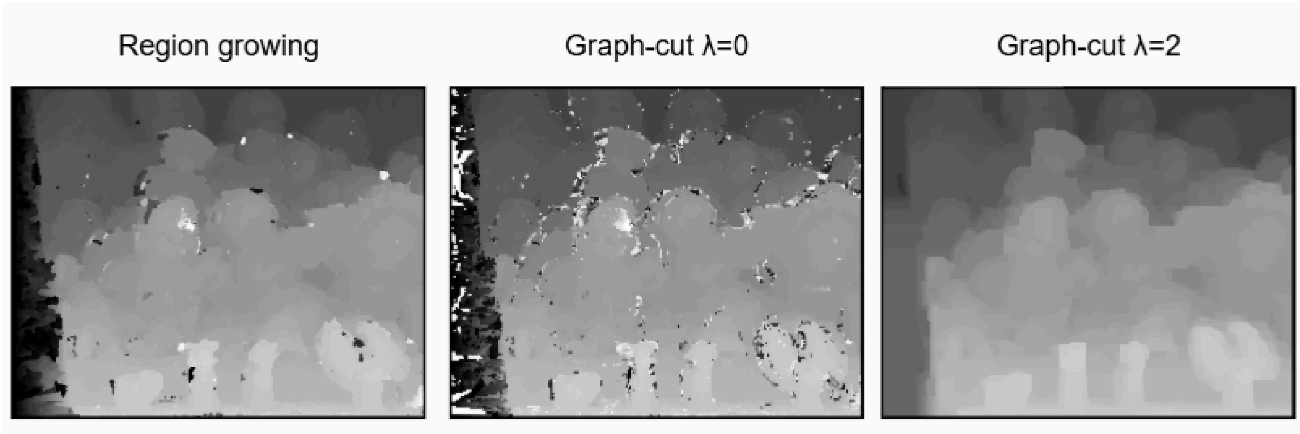

Figure 2: Comparison of disparity maps computed on default images from the third exercise using the region-growing and GC algorithm (λ=0, λ=2).

Tuning patch size

We now evaluate the GC algorithm results obtained for different patch sizes on two datasets:

- The default images from the fourth homework (Figure 3).
- The default images from the third homework (Figure 4).

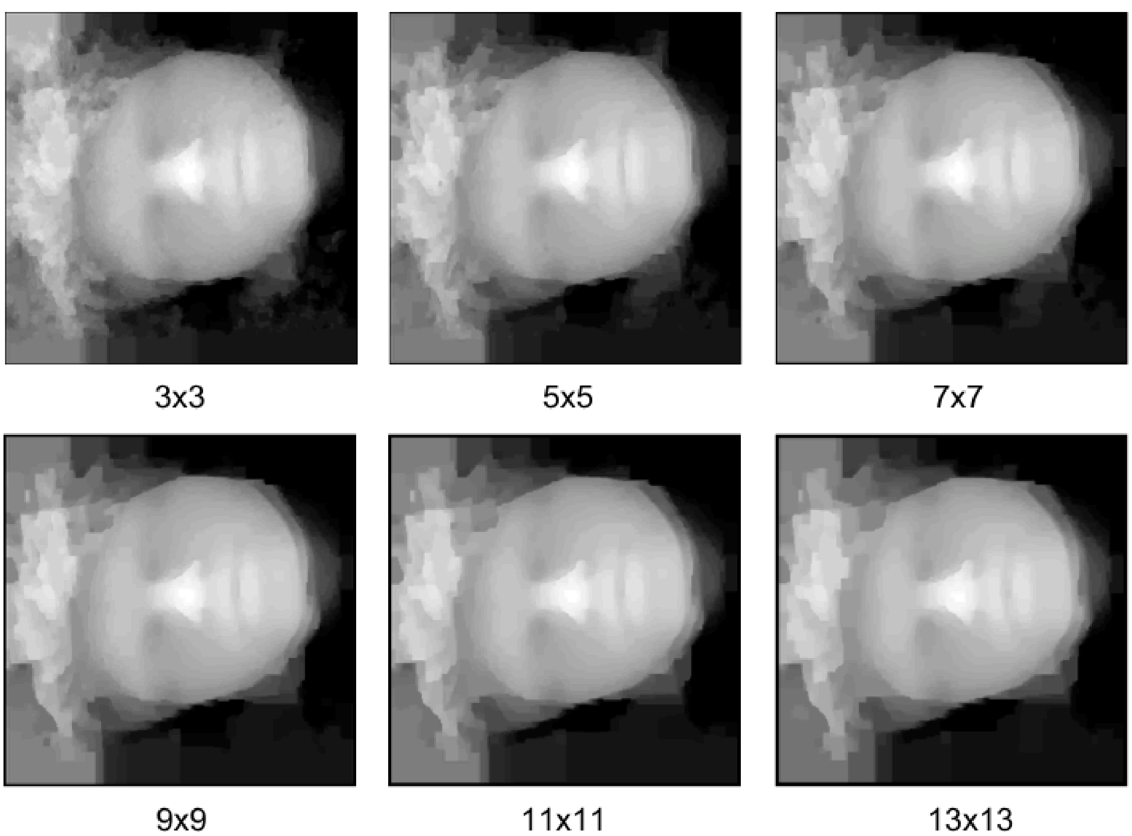

Figure 3: Comparison of disparity maps computed on default images from the fourth exercise for different patch sizes. The number below each image represents the used patch size. Black borders represent invalid disparity regions.

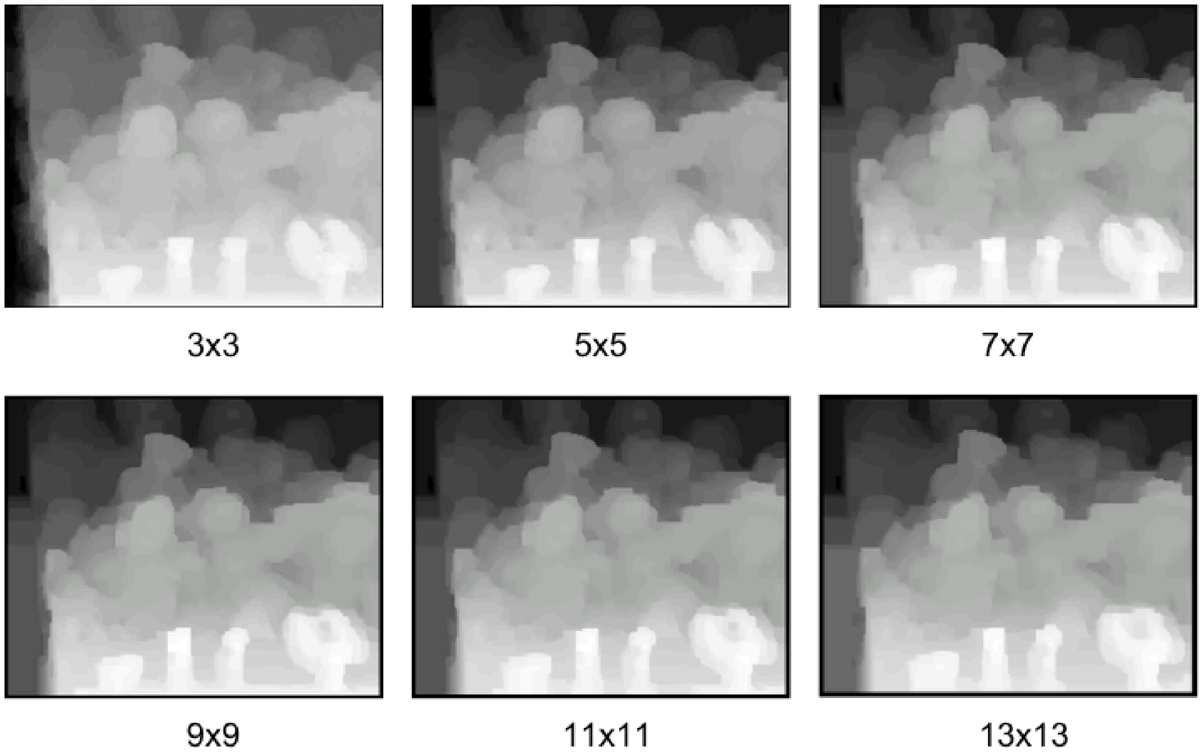

Figure 4: Comparison of disparity maps computed on default images from the third exercise for different patch sizes. The number below each image represents the used patch size. Black borders represent invalid disparity regions.

The patch size determines the area of pixels over which the correlation score between two views is computed. Based on the results, we see that smaller patch sizes result in sharper disparity maps that better capture and preserve details, but also introduce more noise and local inconsistencies. On the other hand, larger patch sizes result in smoother and more homogeneous regions but tend to lose some finer details and structures.

It is also immediately visible that larger patch sizes result in wider black borders (invalid disparity regions), as a greater portion of the image is excluded from computation near the edges.

We must also note that increasing the patch size leads to greater computation time (Table 1). On average, with an image size of 512x512, the parameters $\lambda=10$, zoom=2, and disparity range=[10, 55], the time increases by approximately 1.4x for every two pixel increase in patch size.

Patch size	3x3	5x5	7x7	9x9	11x11	13x13
Time [s]	15.5	20.9	29.9	42.7	59.2	82.7

Table 1: Computation time of the GC algorithm on the default homework’s images for different patch sizes.

In conclusion, the moderate patch sizes (e.g., 7x7/9x9) provide a good balance between detail preservation, smoothness, and computation time.

Additional results that confirm the previous observations are shown in the appendix section.

Region-growing and Graph-cut comparison

We evaluate and compare region-growing and GC algorithms on the [Middlebury 2005 dataset] with the third-size images (Figure 5). Every image pair contains ground-truth (GT) disparity maps to which we will later compare our results.

For both algorithms, we set the disparity range to [0,85] ([ref]) and use NCC/ZNCC scores with 7x7 patches. Specifically, for the GC algorithm, we use a factor zoom=2 and a smoothness factor λ=10.

Figure 5: Six images from the Middlebury 2005 dataset. The left column represents the left image (view 1) and the right column represents the right image (view 5). The name of the specific pair of images (scene) is written below each.

We show the disparity maps from both region-growing and GC algorithms alongside the ground-truth maps for all six images (Figure 4). The invalid pixels on our computed maps are colored in black.

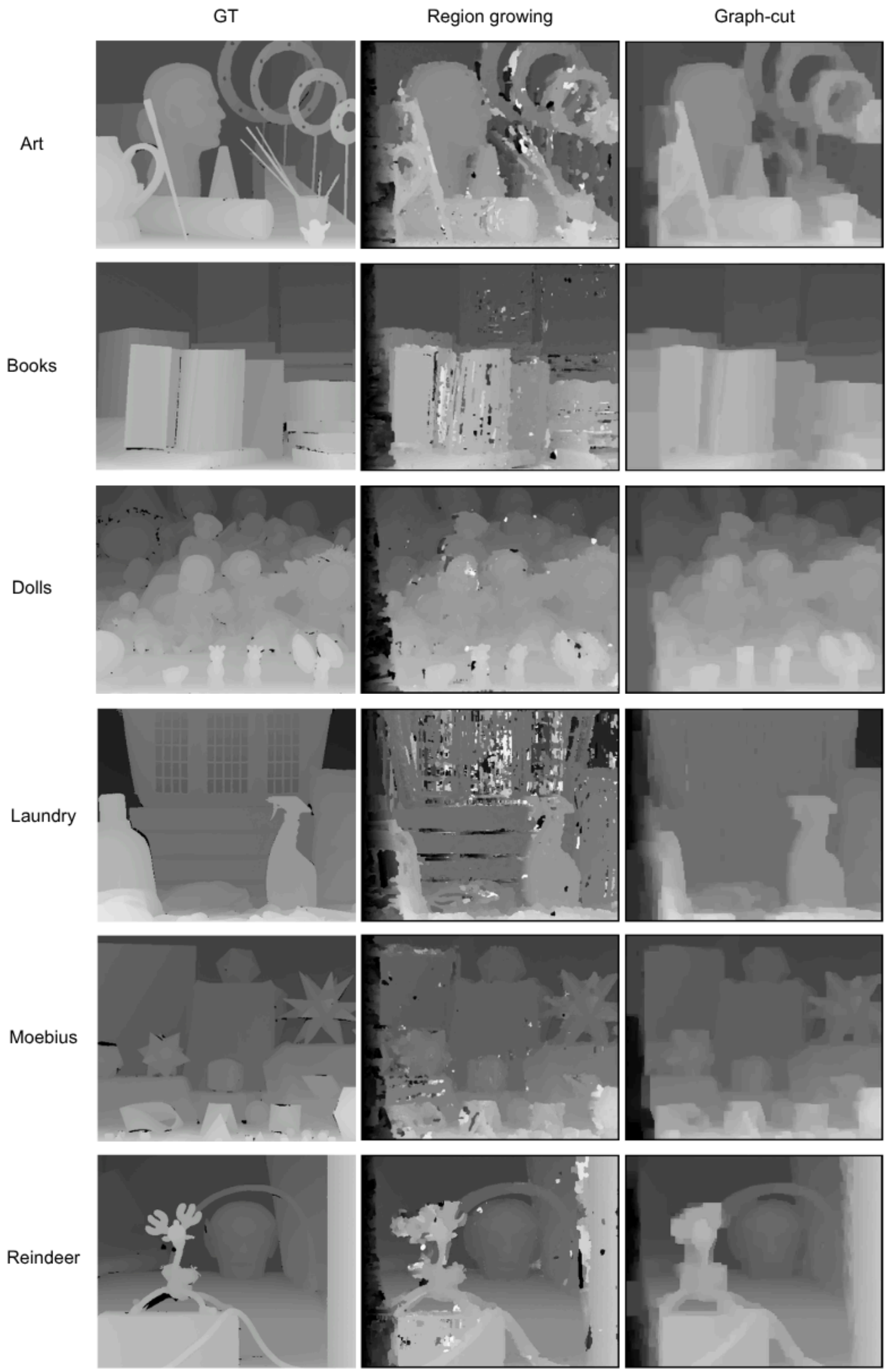

Figure 6: Computed disparity maps of both algorithms, alongside ground-truth maps.

Based on Figure 6, we observe that the GC results are visually much closer to the GT maps compared to those of the region-growing algorithm. The GC maps are noticeably cleaner, with

smoother surfaces and less noisy artifacts. Shapes are more consistent and better preserved. Furthermore, the object boundaries are more coherent.
On the other hand, the region-growing algorithm produces some visible irregularities. More specifically, in the Books scene, which contains sharp edges and fine details, this algorithm struggles to maintain structure. Similarly, in the Laundry scene, the algorithm produces noisier results, especially around the window area where disparity (depth) discontinuities occur at short distances.

We also provide a quantitative comparison between two algorithms (Figure 7). We compute the mean absolute error (MAE), Precision@1, and Smoothness (TV-L1) scores.
Precision@1 represents the percentage of pixels whose difference to the true ground truth is within ±1 range (ref). Smoothness score measures the total variation of neighboring pixels (lower score=>smoother map).

Image name	**Method**	**MAE**	**Precision@1 (%)**	**Smoothness (TV)**
Art	Seed	7.132	62.13	0.94
Art	GC	5.794	64.62	0.24
Books	Seed	4.47	72.04	0.912
Books	GC	1.337	81.06	0.105
Dolls	Seed	3.616	78.26	0.504
Dolls	GC	2.13	79.2	0.14
Laundry	Seed	8.142	56.64	1.837
Laundry	GC	4.639	67.98	0.158
Moebius	Seed	3.525	74.14	0.575
Moebius	GC	2.978	76.45	0.155
Reindeer	Seed	5.157	74.15	0.628
Reindeer	GC	4.07	75.59	0.203

Figure 7: Quantitative results comparison.

As shown, the GC outperforms the region-growing algorithm across all metrics and all image pairs. The difference is especially significant in the smoothness metric, which is expected given the neighbor regularization term that GC uses. These results confirm our previous qualitative observations.

We also observe that GC is more than twice as fast as the region-growing algorithm. The mean computation time across our dataset images was 37 seconds for GC and 89 seconds for the region-growing algorithm.

Appendix

We show additional results on the GC algorithm for different patch sizes, this time using the Books scene from the Middlebury dataset (Figure 8).

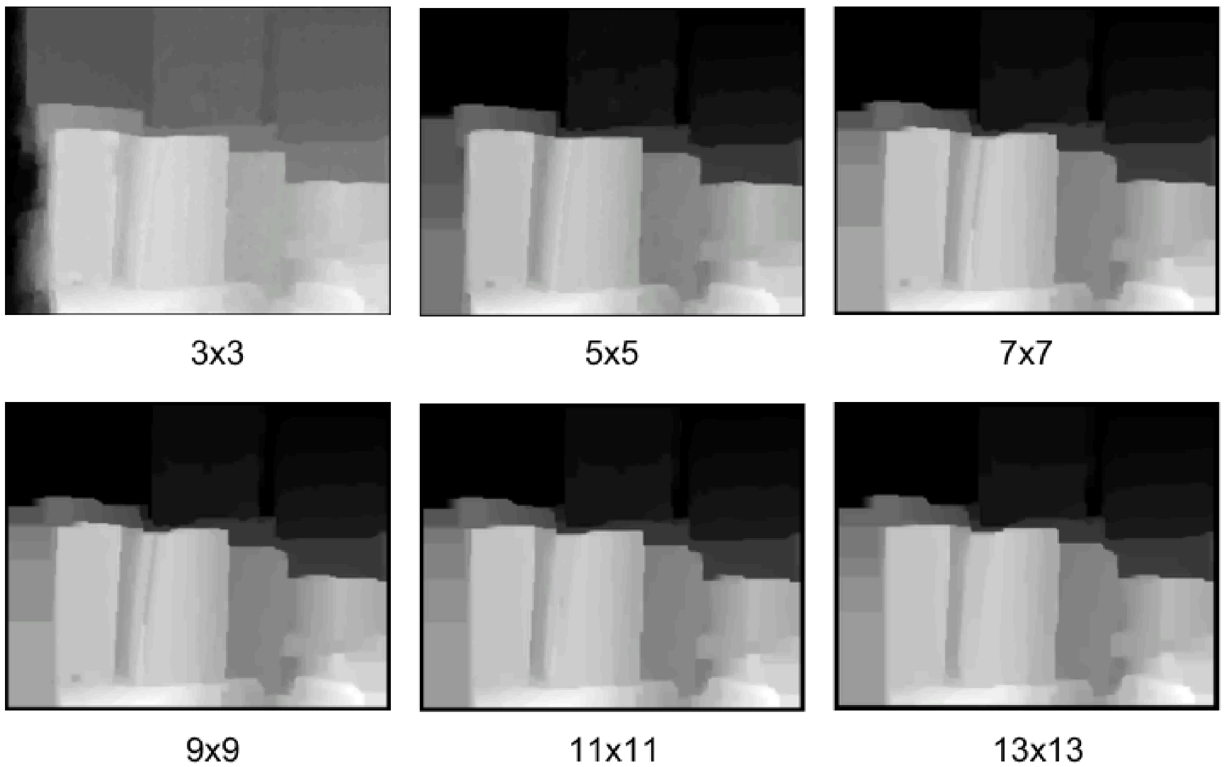

Figure 8: Comparison of disparity maps computed on the Books scene, Middlebury dataset, for different patch sizes. The number below each image represents the used patch size. Black borders represent invalid disparity regions.