Gradient-SDF: A Semi-Implicit Surface Representation for 3D Reconstruction — Supplementary Material

Christiane Sommer* Lu Sang* David Schubert Daniel Cremers
Technical University of Munich
Computer Vision Group
{c.sommer, lu.sang, d.schubert, cremers}@tum.de

This Supplement contains information on code and data, as well as some more visualizations of results. To make it easier to parse, all information about a figure or table can be found in the respective caption and no additional text is added.

1. Code and reproducibility

Our code is on `https://github.com/c-sommer/gradient-sdf`. We did not make the link to the repository publicly available (while the repository itself is), and would also ask reviewers not to do so. Installation and compilation instructions can be found in the `README` file.

With the default settings in our C++ code, the results from the paper are largely reproducible, since we explicitly switched off OpenMP to generate the numbers. The code can also be used with OpenMP to get a better feeling for speed using some basic CPU parallelization. This will lead to slightly different results due to random summation order in the Gauss-Newton step. For the gradient analysis experiment, we additionally provide Matlab code to generate the sphere data and to process the tracking results.

2. Used code and data

		code/data	year	link	license
[27]	TUM RGB-D Benchmark	dataset	2012	`https://vision.in.tum.de/data/datasets/rgbd-dataset`	CC BY 4.0
[8]	Redwood Large Dataset of Object Scans	dataset	2016	`http://www.redwood-data.org/3dscan/`	Public Domain
[33]	SSD Surface Reconstruction (version 3.0)	code	2011	`http://mesh.brown.edu/ssd/software.html`	BSD-3
[27]	TUM RGB-D Benchmark	code	2012	`https://vision.in.tum.de/data/datasets/rgbd-dataset/tools`	BSD-2
[7]	SDF Tracker	code	2013	`https://code.google.com/archive/p/oru-ros-pkg/`	BSD-3
[22]	BAD SLAM	code	2019	`https://github.com/ETH3D/badslam`	BSD-3

Table A.1. Used datasets and code in our submission, together with reference, link and license. We did our real-world experiments on two datasets, TUM RGB-D (for which ground truth poses exist), and Redwood LOD (without ground truth). For code, we used the smooth signed distance surface reconstruction (SSD) to generate a mesh from our output surfel cloud, and SDF Tracker to reconstruct geometry following [7], see Figures A.3 and A.4, right column. The TUM RGB-D code was used to compute ATEs of estimated camera poses, and BAD SLAM for qualitative and quantitative results of dense RGB-D bundle adjustment.

3. Gradient quality visualized

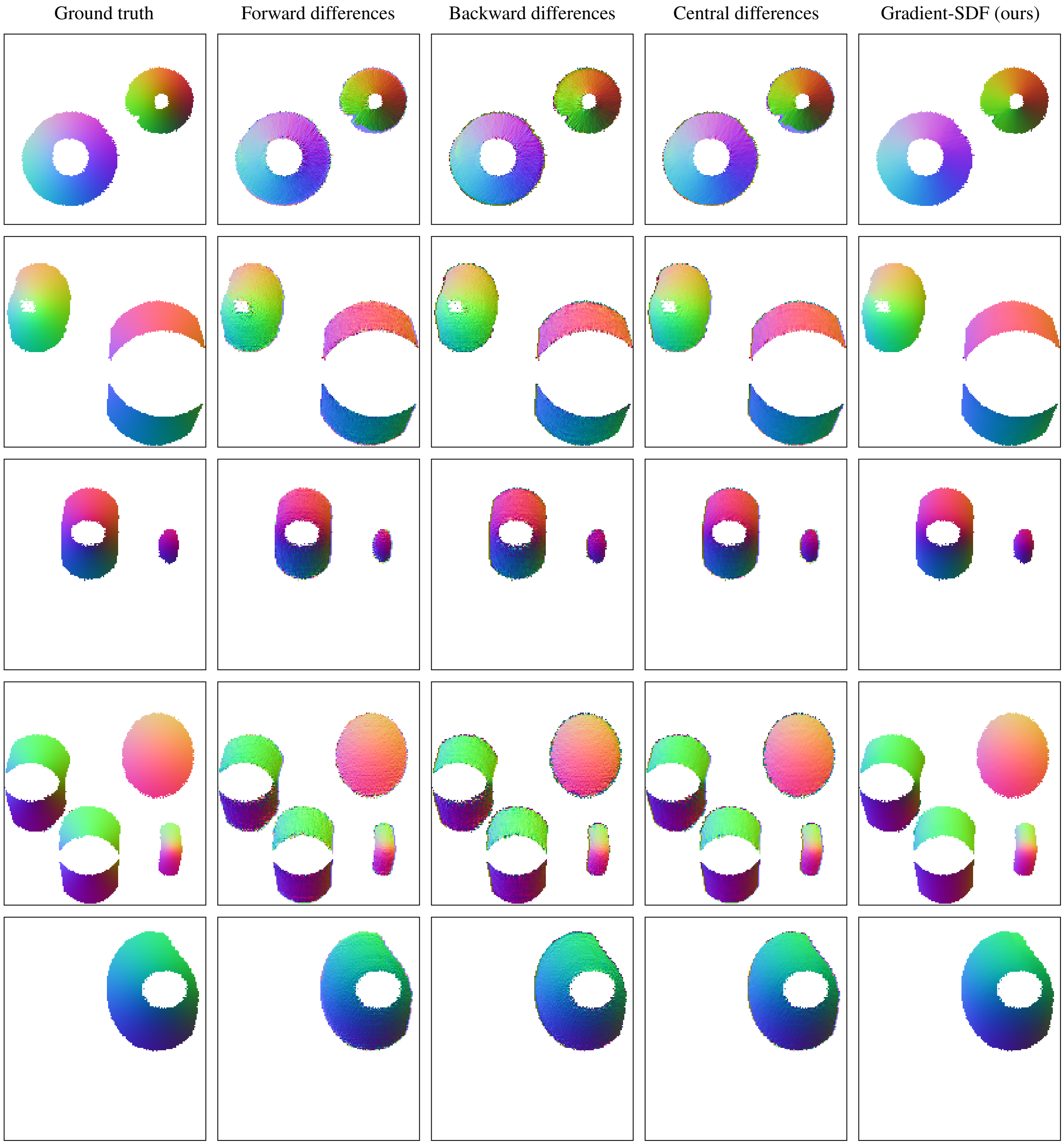

Figure A.1. Qualitative results for the gradient of the signed distance function of the rendered spheres from Figure 3 in the main paper. Shown are five slices through the 3D gradient color-coded as normal map (after normalization) in regions where the computed SDF is non-truncated and has a non-zero weight (truncation is chosen as $10v_s$ for a voxel size v_s of 1 cm). While central finite differences (*column 4*) are clearly better and less noisy than simple forward or backward differences (*columns 2, 3*), our Gradient-SDF vectors (*right*) are still significantly smoother and less noisy. Comparing to the ground truth (*left*), we can visually confirm the quantitative result from Figure 3: our gradients are a lot closer to the ground truth ones than central finite differences.

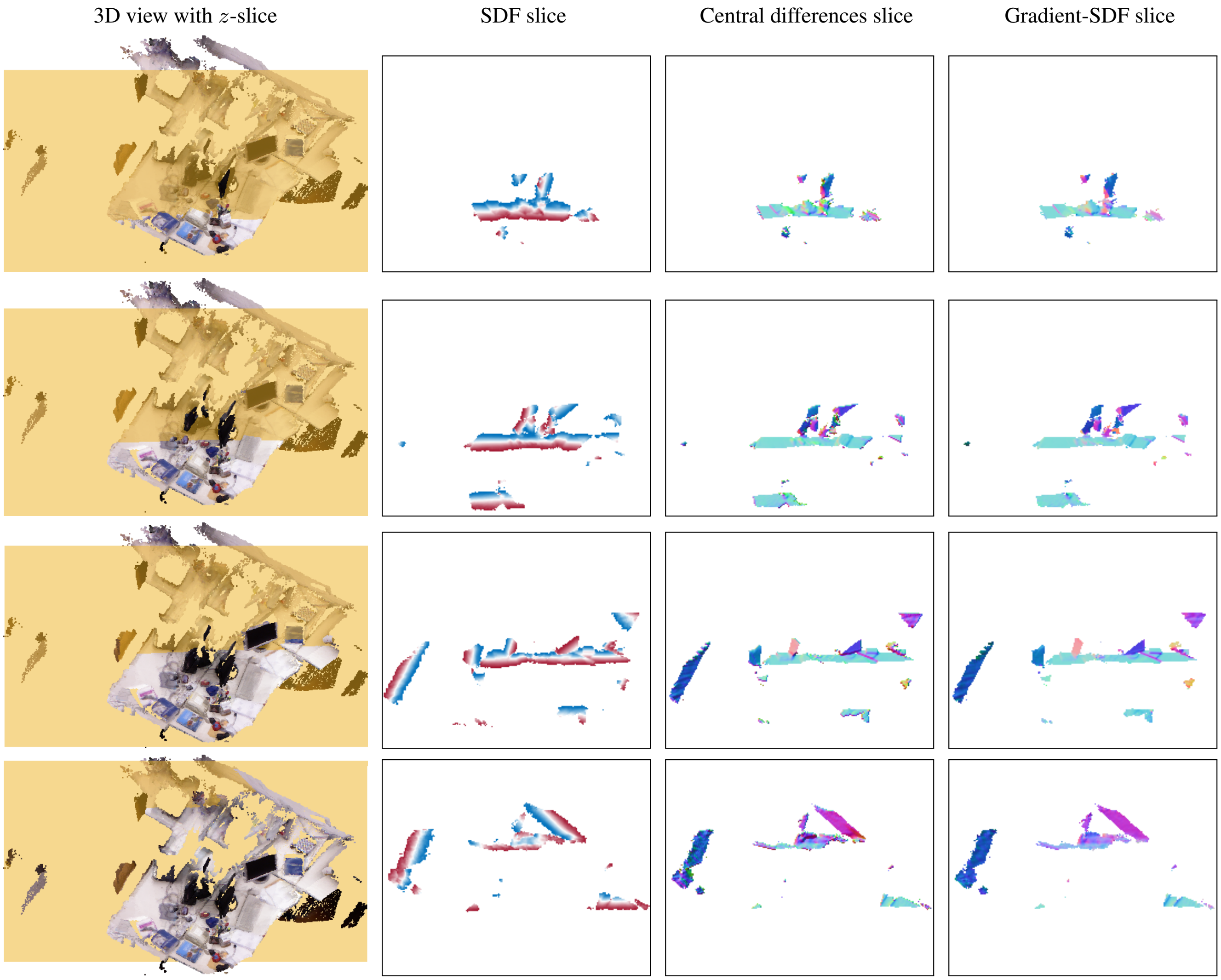

Figure A.2. Qualitative results for the gradient of the signed distance function of the *fr1/desk1* sequence of [27]. Shown are four slices perpendicular to the z-axis through the SDF and the 3D gradient color-coded as normal map (after normalization), at $z = 0.60$, $z = 0.86$, $z = 1.14$, and $z = 1.90$. The *left* column shows the position of the slice in the colored 3D point cloud for better orientation. Just like for the synthetic data in Figure A.1, our Gradient-SDF vectors are smoother than the ones obtained using finite differences.

4. Geometry after depth tracking

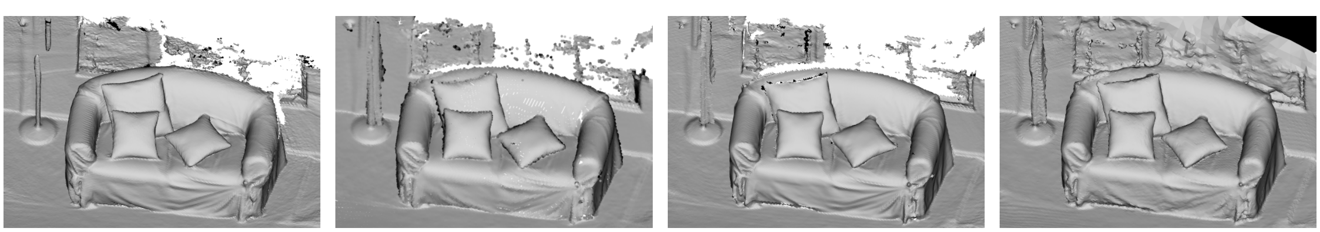

Figure A.3. Grayscale-shaded 3D reconstructions of the *00577_sofa* from [8], see also Figure 2. *Left* shows the reconstruction obtained from running SDF Tracker [7] with a 512^3 volume. The remaining three figures show different visualizations of our reconstruction: an oriented point cloud (surfels), the mesh extracted using marching cubes, and the mesh reconstructed from the point cloud using SSD surface reconstruction [33].

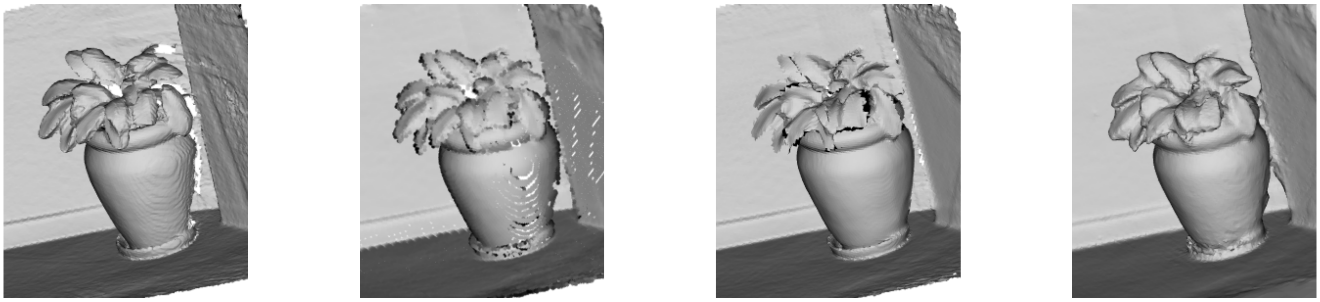

Figure A.4. Grayscale-shaded reconstructions of the *06127_plant* from [8]. Visualized are the same four reconstruction modes as in Figure A.3, from *left* to *right*: SDF Tracker mesh, our oriented point cloud, our mesh using marching cubes, and mesh reconstruction using SSD from our point cloud. As explained in the main paper, our tracker can perform direct SDF tracking using a hash map, achieving comparable results to SDF Tracker while storing significantly less voxels.

5. Photometric Optimization

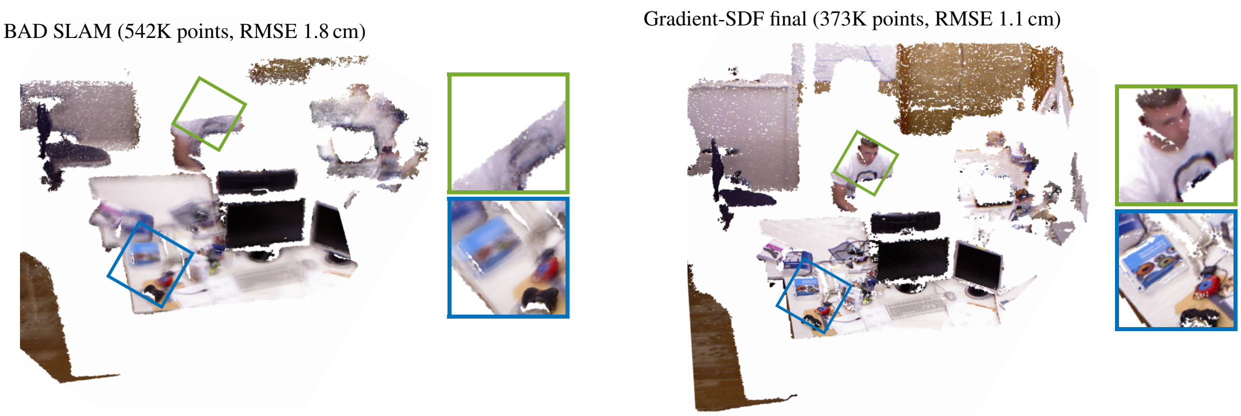

Figure A.5. Colored point cloud reconstruction produced by BAD SLAM (*left*) and after optimization of the BA cost on our Gradient-SDF (*right*) on the *fr1/xyz* sequence of [27].

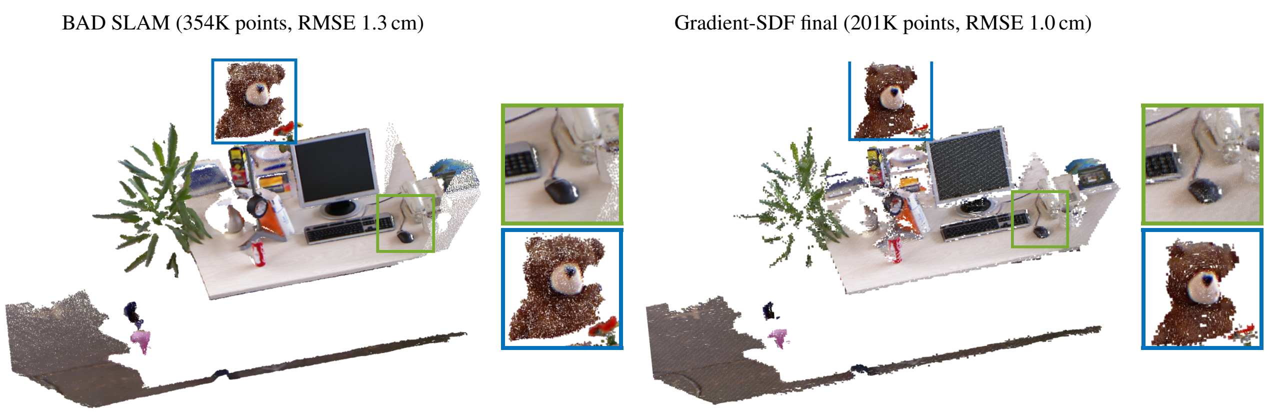

Figure A.6. Colored point cloud reconstruction produced by BAD SLAM (*left*) and after optimization of the BA cost on our Gradient-SDF (*right*) on the *fr2/xyz* sequence of [27].

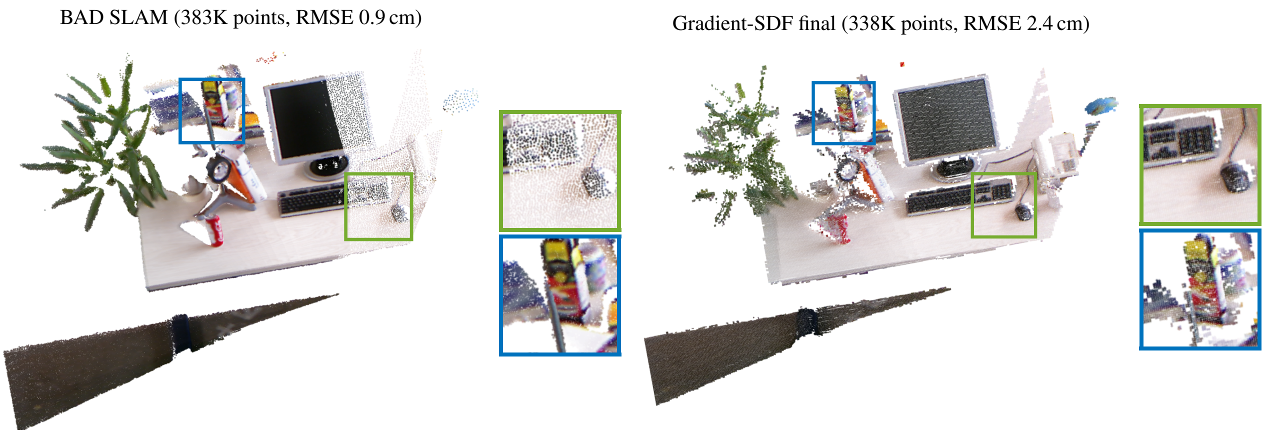

Figure A.7. Colored point cloud reconstruction produced by BAD SLAM (*left*) and after optimization of the BA cost on our Gradient-SDF (*right*) on the *fr2/rpy* sequence of [27].

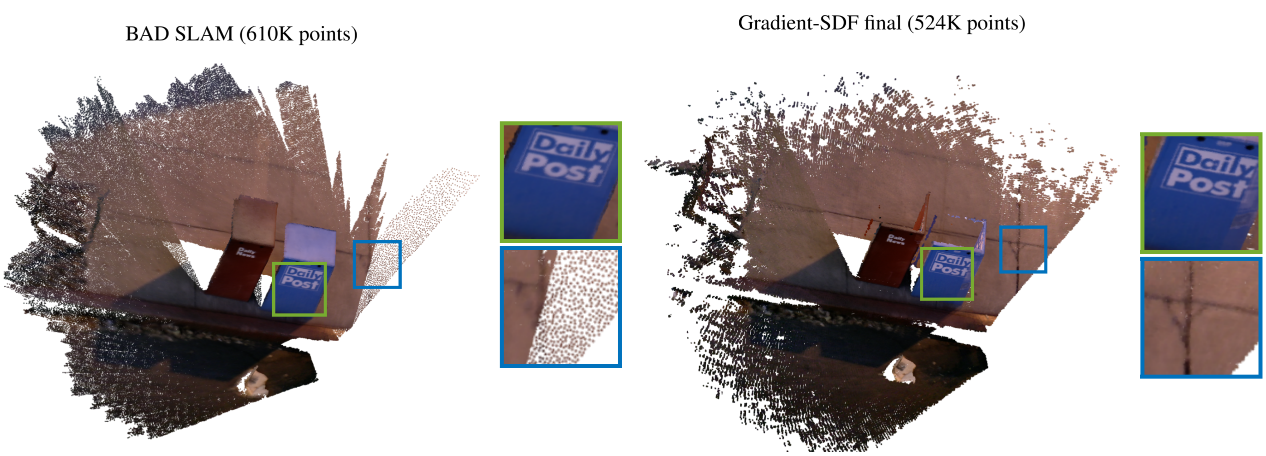

Figure A.8. Colored point cloud reconstruction produced by BAD SLAM [22] (*left*) and after optimization of the BA cost on our Gradient-SDF (*right*) on the *01847_kiosk* sequence of [8]. For this dataset, no ground truth poses are available, thus no ATE is stated.

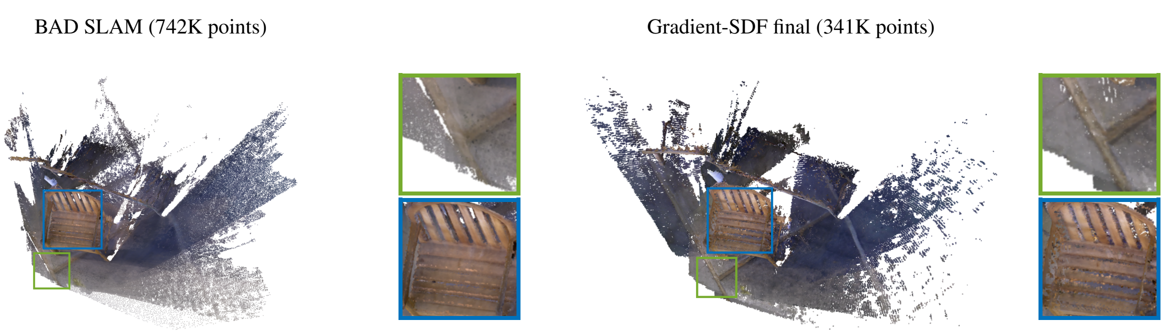

Figure A.9. Colored point cloud reconstruction produced by BAD SLAM [22] (*left*) and after optimization of the BA cost on our Gradient-SDF (*right*) on the *08309_woodenchair* sequence of [8].

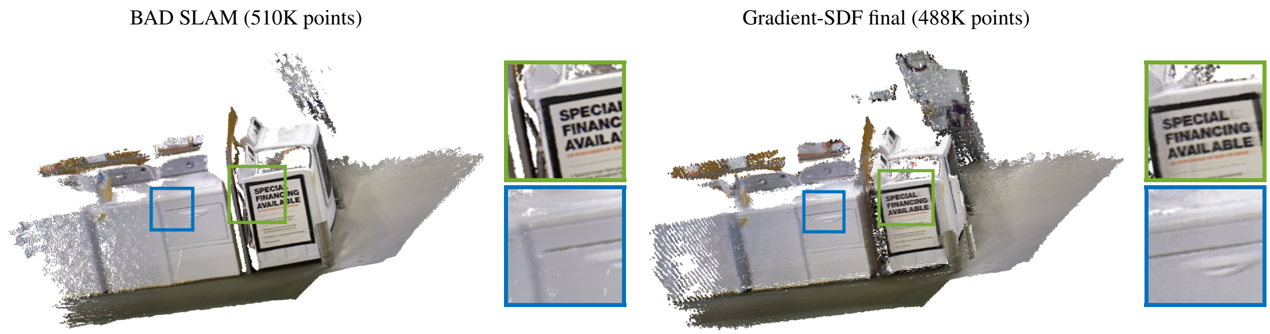

Figure A.10. Colored point cloud reconstruction produced by BAD SLAM [22] (*left*) and after optimization of the BA cost on our Gradient-SDF (*right*) on the *05635_washmachine* sequence of [8].

References

[33] Fatih Calakli and Gabriel Taubin. SSD: Smooth signed distance surface reconstruction. In *Computer Graphics Forum*, volume 30, pages 1993–2002. Wiley Online Library, 2011.