Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions

Naveen Kumar Subramanian

M.Sc Computational Science and Engineering - Technische Universität München

Abstract

In this paper we present a novel method for real-time camera tracking and 3D reconstruction of static indoor environments using an RGB-D sensor. We show that by representing the geometry with a signed distance function (SDF), the camera pose can be efficiently estimated by directly minimizing the error of the depth images on the SDF. As the SDF contains the distances to the surface for each voxel, the pose optimization can be carried out extremely fast. By iteratively estimating the camera poses and integrating the RGB-D data in the voxel grid, a detailed reconstruction of an indoor environment can be achieved. We present reconstructions of several rooms using a hand-held sensor and from onboard an autonomous quadrocopter. Our extensive evaluation on publicly available benchmark data shows that our approach is more accurate and robust than the iterated closest point algorithm (ICP) used by KinectFusion, and yields often a comparable accuracy at much higher speed to feature-based bundle adjustment methods such as RGB-D SLAM for up to medium-sized scenes

1 Introduction

3D simultaneous localization and mapping (SLAM) is a highly active research area as it is a pre-requisite for many robotic tasks such as localization, navigation, exploration, and path planning.Structure from motion (SfM) techniques from computer vision typically use images from a moving camera.The contribution of this work is a novel method for estimating the camera motion directly based on the SDF. The key insight behind our approach is that the SDF already encodes the distance of each voxel to the surface.

2 Main Objective

In this paper,

- We describe a direct approach to camera tracking on SDFs.
- We present a thorough evaluation of SDF based tracking and mapping on public benchmarks
- We compare the tracking performance to existing real-time solutions.
- We study the influence of alternative distance metrics, weighting functions and different camera motions on a large number of different scenes.
- We demonstrate that our approach is directly applicable to position control of a quadrocopter and the automatic 3D reconstruction of rooms as shown in figure 1

Figure 1: Reconstruction of livring oom with a handheld sensor using our approach.

3 Approach

We represent the geometry using a signed distance function stored in a voxel grid. We follow an iterative approach where we first estimate the camera pose given the previous SDF, and then update the SDF based on the newly computed camera pose. Note that we optimize the camera pose directly on the SDF, while KinectFusion [18] first generates a synthetic depth images that it subsequently aligns to the current depth image using ICP.

3.1 Camera Tracking

In this part, we present how we estimate the camera motion given an SDF and a depth image.we assume that we already have a representation of the geometry acquired from the previous depth image given by the SDF Function .This function returns for any point x the signed distance from x to the surface. The idea is now to use the SDF to construct an error metric that describes how well a depth image fits to the SDF.This is explained in Figure 2.

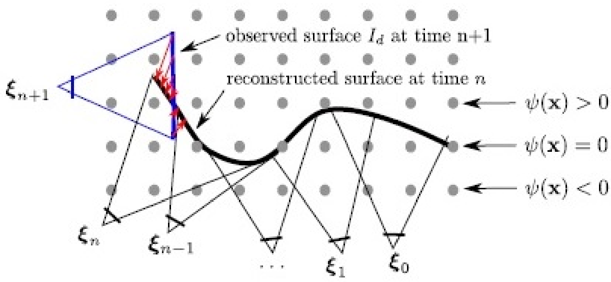


Figure 2: The SDF is constructed from the first n depth images and corresponding camera poses

3.2 Distance and Weighting Functions

These functions are used to determine which voxels have been observed by the depth camera and update their distances accordingly .

3.2.1 Projective Point-to-Point

The projective point-to-point distance as the difference of the depth of the voxel and the observed depth at (i,j) negative values are assigned to voxels in front of the observed surface, and positive values to voxels behind.It is explained in figure 3

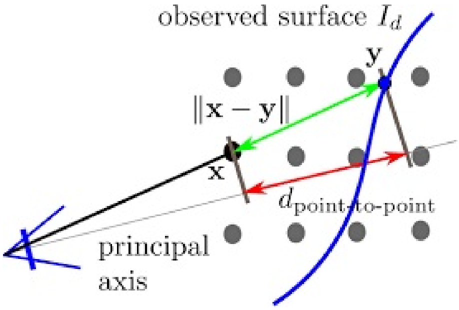


Figure 3: Projective point to point

3.2.2 Projective Point-to-Plane

the point-to-point metric gets increasingly inaccurate the less the viewing angle is orthogonal to the surface . As a first step, we apply a bilateral filter to the depth image and compute the normals for all pixels. Given a voxel x, we compute its corresponding pixel coordinates (i; j) and read out the observed surface normal n(i; j).It is explained in figure 4

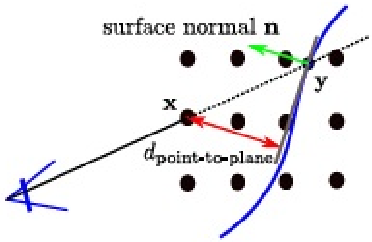
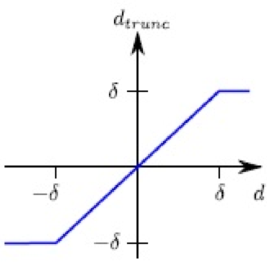


Figure 4: Projective point to plane

3.2.3 Truncation

we truncate the projected distances d and apply a weighting term that blends out large distances, which makes the gradient of our SDF zero in regions that are far away from the estimated surface.For example , consider function shown in figure 5.



$$d_{trunc} = \begin{cases} -\delta & \text{if} \quad d < -\delta \\ d & \text{if} \quad |d| \leq \delta \\ \delta & \text{if} \quad d > \delta \end{cases}$$

Figure 5: We truncate large estimated distances to limit the influence of approximation errors and noise.

3.2.4 Weighting

we employ a weighting function to give higher weights to voxels in front of the observed surface and lower weights to voxels behind. Depending on the observation model of the depth sensor, different weighting functions can be used.The linear weight, as proposed by Curless and Levoy [[2]] and used in KinectFusion, assigns a constant weight to all voxels up to a certain penetration depth Epsilon.

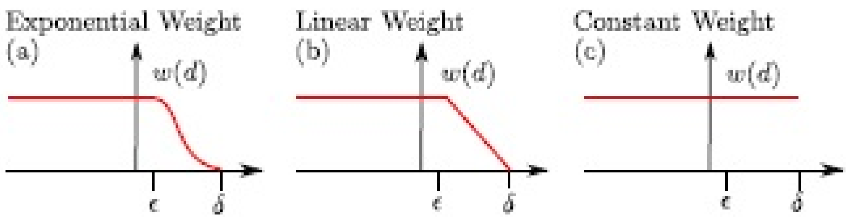


Figure 6: Different Weight functions

4 Experimentl and Results

This approach for 3D reconstruction from an autonomous quadrocopter (see Figure) equipped with an RGB-D camera. Here tracking and reconstruction were carried out in real-time on an external ground station. This demonstrates that our technique is applicable for the navigation of quadrocopters and other robots.Note that in theory, the complexity of pose optimization solely depends on the size of the input images, while the complexity of data fusion depends cubically on the resolution of the volume.

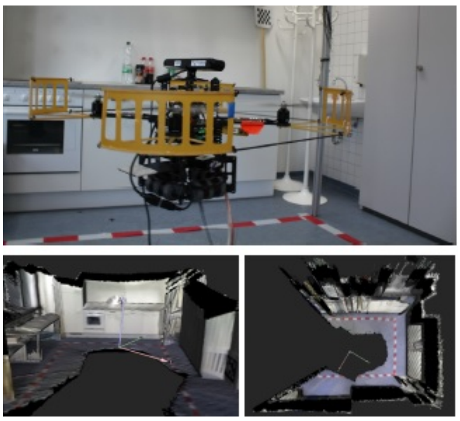


Figure 7: Resulting 3D reconstruction of the room computed in real-time on the ground station

4.1 Benchmark Evaluation

We also evaluated our approach on the TUM RGB-D benchmark :[[4]]. As comparison we used the KinFu implementation [[1]] and RGB-D SLAM [[3]].For our algorithm, we found that the point-to-point metric provides better results on most sequences. In comparison to RGB-D SLAM, we achieve often a similar performance in terms of accuracy but require six to eight times less computation time. The Results are show below.

Method	Res.	Teddy	F1 Desk	F1 Desk2	F3 Household	F1 Floor	F1 360	F1 Room	F1 Plant	F1 RPY	F1 XYZ
KinFu	256	0.156 m	0.057m	0.420 m	0.064 m	Failed	0.913 m	Failed	0.598 m	0.133 m	0.026 m
KinFu	512	0.337 m	0.068 m	0.635 m	0.061 m	Failed	0.591 m	0.304 m	0.281 m	0.081 m	0.025 m
Point-To-Plane	256	**0.072 m**	0.087 m	0.078 m	0.053 m	0.811 m	0.533 m	0.163 m	0.047 m	0.047 m	0.029 m
Point-To-Plane	512	0.101 m	0.059 m	0.623 m	0.053 m	0.640 m	0.206 m	0.105 m	**0.041 m**	**0.042 m**	0.026 m
Point-To-Point	256	0.086 m	0.038 m	**0.061 m**	**0.039 m**	0.641 m	0.420 m	0.121 m	0.047 m	0.047 m	**0.021 m**
Point-To-Point	512	0.080 m	**0.035 m**	0.062 m	0.040 m	**0.567 m**	**0.119 m**	**0.078 m**	0.043 m	**0.042 m**	0.023 m
RGB-D SLAM		0.111 m	0.026 m	0.043 m	0.059 m	0.035 m	0.071 m	0.101 m	0.061 m	0.029 m	0.013 m

Figure 8: BenchMark Evaluation.

5 Conclusion

In this paper we presented a novel approach to directly estimate the camera movement using a signed distance function. Our method allows the quick acquisition of textured 3D models that can be used for real-time robot navigation. By evaluating our method on a public RGB-D benchmark, we found that it outperforms ICP-based methods such as KinFu and, at least on medium-sized scenes, often obtains a comparable performance with bundle adjustment methods such as RGB-D SLAM at a significantly reduced computational effort.

References

[1] Kinectfusion implementation in the point cloud library. http://svn.pointclouds.org/pcl/trunk/.

[2] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In SIGGRAPH, 1996.

[3] F.Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the rgb-d slam system. In ICRA, May 2012.

[4] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam system. In IROS, 2012.