Direct Camera Pose Tracking and Mapping With Signed Distance Functions

Bylow, Erik; Sturm, Jürgen; Kerl, Christian; Kahl, Fredrik; Cremers, Daniel

2013

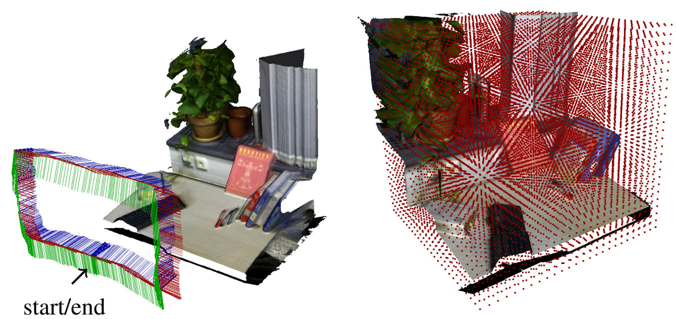
LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Direct Camera Pose Tracking and Mapping With Signed Distance Functions

Erik Bylow*, Jürgen Sturm†, Christian Kerl†, Fredrik Kahl*
and Daniel Cremers†
*Center for Mathematical Sciences, Lund University, Lund, Sweden
Email: erikb@maths.lth.se, fredrik@maths.lth.se
† Department of Computer Science, Technical University of Munich, Garching, Germany
Email: juergen.sturm@in.tum.de, christian.kerl@in.tum.de, cremers@in.tum.de

***Abstract*—In many areas, the ability to create accurate 3D models is of great interest, for example, in computer vision, robotics, architecture, and augmented reality. In this paper we show how a textured indoor environment can be reconstructed in 3D using an RGB-D camera. Real-time performance can be achieved using a GPU. We show how the camera pose can be estimated directly using the geometry that we represent as a signed distance function (SDF). Since the SDF contains information about the distance to the surface, it defines an error-metric which is minimized to estimate the pose of the camera. By iteratively estimating the camera pose and integrating the new depth images into the model, the 3D reconstruction is computed on the fly. We present several examples of 3D reconstructions made from a handheld and robot-mounted depth sensor, including detailed reconstructions from medium-sized rooms with almost drift-free pose estimation. Furthermore, we demonstrate that our algorithm is robust enough for 3D reconstruction using data recorded from a quadrocopter, making it potentially useful for navigation applications.**



(a) (b)

Fig. 1: On small work spaces, our method is nearly drift-free. (a) 3D reconstruction and the estimated camera trajectory of a small office scene. (b) Visualization of the (downsampled) voxel grid underlying the reconstruction volume ($m = 256$).

I. 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. To be truly useful, such systems require the fast and accurate estimation of the robot pose and the scene geometry.

This extended abstract is based upon our recent work [2], of which we plan to give a live demonstration during the RSS RGB-D workshop. An example of a 3D model acquired with our approach are shown in Figure 1. Our scanning equipment consists of a handheld Microsoft Kinect sensor and a laptop with a GPU from Nvidia. The laptop provides a live view on the reconstructed model. As can be seen in the figure, the resulting models are highly detailed and provide absolute metric information about the scene which is useful for a large variety of subsequent tasks.

The contribution of this work is to use the signed distance function (SDF) directly to estimate the camera pose. Using this approach and in contrast to KinectFusion [10], we do not need to generate a depth image from the SDF or to run the iteratively closest point (ICP) algorithm. As a result, we obtain an increased accuracy and robustness [2].

This work has partially been supported by the DFG under contract number FO 180/17-1 in the Mapping on Demand (MOD) project.

II. Related Work

Simultaneous localization and mapping refers to both the estimation of the camera pose and mapping of the environment.

Laser-based localization and mapping approaches often use scan matching or the ICP [1] to estimate the motion between frames. Graph SLAM methods use these motion estimates as input to construct and optimize a pose graph [8]. The resulting maps are often represented as occupancy grid maps or octrees [12] and are therefore well suited for robot localization or path planning. [6] were the first to apply the Graph SLAM approach to RGB-D data using a combination of visual features and ICP. A similar system was recently presented by [5] and extensively evaluated on a public benchmark [11].

Newcombe et. al. [10] recently demonstrated with their well-known KinectFusion approach that dense reconstruction is possible in real-time by using a Microsoft Kinect sensor.

Midway through this work we got know about the master thesis of [3] who developed an approach for camera tracking similar to ours. However, his focus lies more on object detection and recognition in an SDF, and no thorough evaluation of the accuracy was performed.

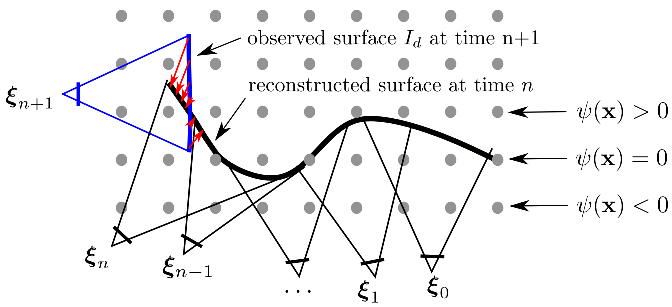


Fig. 2: Our goal is to find the camera pose $\boldsymbol{\xi}_{n+1}$ such that the SDF values between the reprojected 3D points is minimized. The SDF is constructed from the first n depth images and corresponding camera poses $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n$.

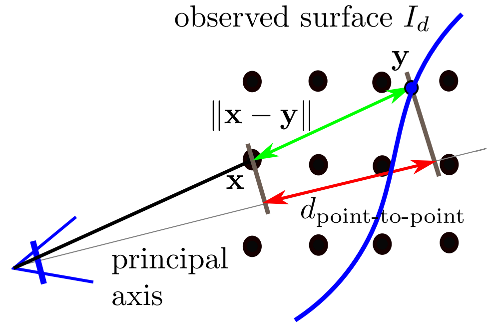


Fig. 3: Visualization of the projective point-to-point distance. Note that computing the true distance is computationally involved.

III. Approach

The geometry is represented using a signed distance function stored in a voxel grid, based on the work by [4]. We follow an iterative approach where first the camera pose given the SDF is estimated, and then the SDF is updated when the camera pose is found. In Section III-A the tracking problem on a given SDF is solved. In Section III-C a method to update the SDF efficiently given a new depth image is presented.

A. Camera Tracking

Here we show how the pose of the camera is estimated and we assume for now that we have an estimation of the SDF, $\psi : \mathbb{R}^3 \to \mathbb{R}$, available, which represents the 3D model seen from the n first images.

For each pixel (i, j), we have its depth $z = I_d(i, j)$. Given this, we can reconstruct the corresponding 3D point $\mathbf{x}_{ij}$ in the local coordinate system. By transforming this point to the global coordinate frame, $\mathbf{x}_{ij}^G = R\mathbf{x}_{ij} + \mathbf{t}$, the distance to the surface can be read in the SDF. Given that the SDF and the camera pose is correct, the reported value should then be zero.

The optimal rotation R and translation $\mathbf{t}$ is the one that re-projects as many 3D points as close to the surface as possible. This idea is illustrated in Figure 2.

To find the rotation and translation the SDF is used to define an error-function

$$E(R, \mathbf{t}) = \sum_{i,j} \psi(R\mathbf{x}_{ij} + \mathbf{t})^2, \tag{1}$$

where i, j iterate over all pixels in the depth image. Remember that in an SDF, all points on the surface have a distance of zero. In the noise free case the error function would give an optimal error of zero. In practice, due to noise, the error function will never be exactly zero.

To minimize this error function we use the Lie algebra representation of rigid-body motion as the twist coordinates $\boldsymbol{\xi} = (r_x, r_y, r_z, t_x, t_y, t_z)$, as described in [9]. Using this notation, we can short write $\psi(R\mathbf{x}_{ij} + \mathbf{t})$ as $\psi_{ij}(\boldsymbol{\xi})$ and rewrite (1) as

$$E(\boldsymbol{\xi}) = \sum_{i,j} \psi_{ij}(\boldsymbol{\xi})^2, \tag{2}$$

To minimize this we start by linearizing ψ around our initial pose estimate $\boldsymbol{\xi}^{(0)}$ that we set to the estimated previous camera pose $\boldsymbol{\xi}_n$ of time step n and plugging this into (2) which gives us a quadratic form that approximates the original error function, i.e.,

$$E_{\text{approx}}(\boldsymbol{\xi}) = \sum_{i,j} (\psi_{ij}(\boldsymbol{\xi}^{(k)}) + \nabla\psi_{ij}^\top(\boldsymbol{\xi} - \boldsymbol{\xi}^{(k)}))^2. \tag{3}$$

Putting the derivative of (3) to zero results in a system of linear equations

$$\mathbf{b} + A\boldsymbol{\xi} - A\boldsymbol{\xi}^{(k)} = 0. \tag{4}$$

From this, we can compute the camera pose that minimizes the linearized error as

$$\boldsymbol{\xi}^{(k+1)} = \boldsymbol{\xi}^{(k)} - A^{-1}\mathbf{b}. \tag{5}$$

Based on this new estimate, we re-linearize the original error function (2) and solve iteratively (5) until convergence.

B. Estimating the Distance Function

With known rotation and translation of the camera, the SDF can be updated with the new depth image. Here we present how the point-to-point metric can be used for estimating the SDF.

For each vertex the global (center) coordinates $\mathbf{x}^G$ are known. Given the pose of the current camera $R, \mathbf{t}$, the local coordinates are found by $\mathbf{x} = (x, y, z)^\top = R^\top(\mathbf{x}^G - \mathbf{t})$.

Using the pinhole camera model we can project $\mathbf{x}$ to the pixel $(i, j)^\top$ in the image. We define then the projective point-to-point distance as the difference of the depth of the voxel and the observed depth at $(i, j)^\top$, i.e., $d(\mathbf{x}) := z - I_d(i, j)$.

To decrease the impact of uncertain measurements the estimated distances are truncated and weighted, as proposed by [4].

C. Data Fusion and 3D Reconstruction

To integrate the depth images into the voxel grid we follow the procedure proposed by [4]. To find the SDF which takes all measurements into account the energy function

$$L(\psi) = \sum_{i=1}^{n} \frac{1}{2} w_i (\psi - \psi_i)^2 \tag{6}$$

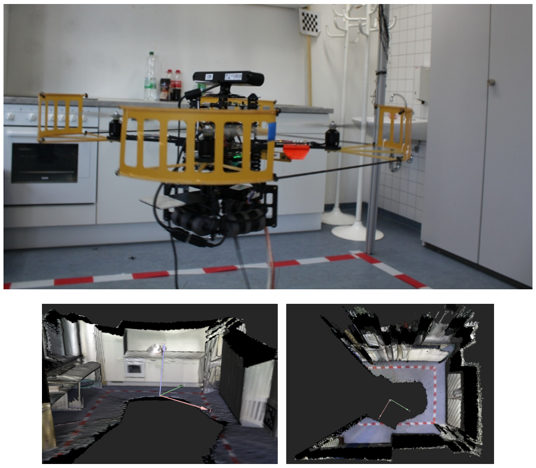


Fig. 4: 3D reconstruction using an autonomous quadrocopter. Top: AscTec Pelican platform used. Bottom: Reconstructed 3D model of a room computed on the ground station. it is only a single model, not several models!

is minimized. The result is the weighted average of all measurements, which can be computed as a running weighted average for each voxel by computing

$$D \leftarrow \frac{WD + w_{n+1} d_{n+1}^{trunc}}{W + w_{n+1}} \quad (7)$$

$$W \leftarrow W + w_{n+1}. \quad (8)$$

Here D is the averaged and weighted distance for the n first images and W is the accumulated weight for the n first images, w_{n+1} and d_{n+1}^{trunc} is the weight and truncated distance for image $n+1$.

IV. Results

In this section we present qualitative results of 3D reconstructions from live-data. For a more comprehensive evaluation we refer to [2].

Figure 1 show a desk scene using our algorithm at a grid resolution of $m = 256$. The resulting reconstruction is highly detailed and metrically accurate, so that it could for example be used by architects and interior designers for planning and visualization tasks.

The method is almost drift-free for small scenes, as can be seen in Figure 1a, where we started and ended a rectangular camera motion at the same spot. Fine details such as the cover appear sharply.

Our approach was also used for 3D reconstruction from an autonomous quadrocopter (see Figure 4) equipped with an RGB-D camera. Note that tracking and reconstruction were carried out in real-time on an external ground station with GPU support. The estimated pose was directly used for position control. This demonstrates that our technique is applicable for robot navigation.

V. 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 obtains a comparable performance with bundle adjustment methods such as RGB-D SLAM at a significantly reduced computational effort. In the future, we plan to include color information in camera tracking and investigate more efficient representation of the 3D geometry. For larger geometries, the combination of our method with a SLAM solver like [8, 7] would be interesting.

References

[1] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14 (2):239–256, 1992.

[2] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers. Real-time camera tracking and 3d reconstruction using signed distance functions. In *RSS*, 2013.

[3] D. Canelhas. Scene representation, registration and object detection in a truncated signed distance function representation of 3d space. Master's thesis, 2012.

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

[5] 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.

[6] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. 2010.

[7] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *ICRA*, 2011.

[8] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *ICRA*, 2011.

[9] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

[10] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, and A.W. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011.

[11] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012.

[12] K.M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *ICRA*, 2010.