3D PANORAMA RECONSTRUCTION BASED ON SUBMAP JOINING

Huixuan Wang[1], Yanwen Guo[2], Minh N. Do[3], Caiming Zhang[1], Changhe Tu[1]

1 School of Computer Science and Technology, Shandong University, Jinan, Shandong 250101, P.R. China
2 National Key Lab for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210023, P.R. China
3 Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.

ABSTRACT

We present a new approach for constructing the 3D panorama for an indoor environment by joining together aligned submaps. For each submap, the trajectory of the moving camera is estimated based on the Kanade-Lucas-Tomasi (KLT) features. Our method can update the feature set status by adding new features and removing expiring ones adaptively to accommodate scene changes. The accuracy of the estimated poses is further improved through sparse bundle adjustment. Furthermore, we utilize a linear optimization framework to align all submaps to obtain a consistently extended 3D panorama and to refine the visual odometry at the same time. We evaluated our approach on publicly available benchmark datasets. The experiments demonstrate that the proposed method achieves low translational drift and is robust even when the camera moves very fast.

Index Terms— SLAM, RGB-D camera, linear optimization, visual odometry

1. INTRODUCTION

Visual odometry refers to estimation of camera motion. Drift-free visual odometry is specifically important to robot localization, path planning, navigation and augmented reality [1-5]. Recently, with the introduction of novel RGB-D sensors such as Microsoft Kinect, the technique of simultaneous localization and mapping (SLAM) is widely explored in these fields.

Most SLAM methods favor sparse feature extraction [6-9]. Endres et al. [9] propose a RGB-D SLAM system which estimates the pairwise transformation to build a pose graph based on several types of features detected. For the cases that few feature matches could be established, Henry et al. [12] use iterative closest point (ICP) to align the dense point clouds. They optimize the system by using sparse bundle adjustment (SBA). In contrast to feature-based methods, dense approaches have begun to emerge in recent years [13-17]. KinectFusion [15] registers each new measurement with the already constructed map via ICP in a frame-to-model manner. However, it does not optimize those previous camera poses, making accumulated drift difficult to correct. It has been shown that instead of the geometric error, accumulated photometric error between consecutive frames can also be minimized [16]. Additionally, visual SLAM systems combining the photometric and geometric errors have been developed [13, 17]. Kerl et al. [13] show that their system outperforms several state-of-the-art sparse feature-based methods. A key issue that hampers the practical application of those dense methods is that such methods always suffer from high computational cost.

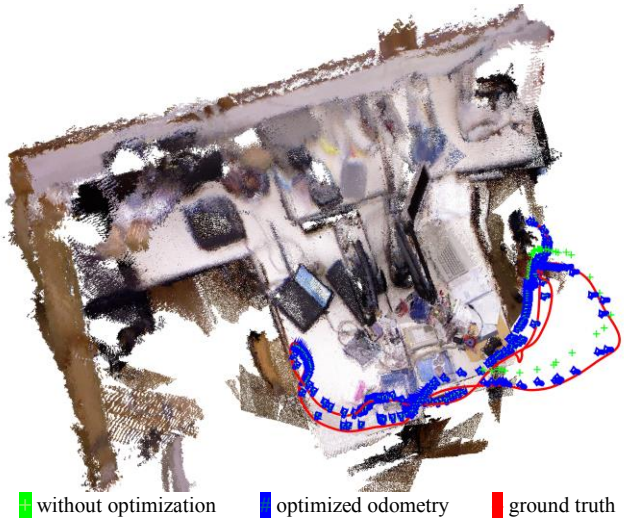
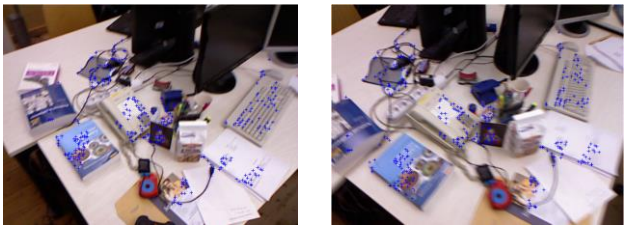


Fig. 1 Comparison between the estimated odometry without and with linear optimization and the ground truth for the fr1/desk dataset.

Filtering-based methods have been used for large-scale SLAM systems [18-21]. Camera poses and feature locations are included in the state vector of a filter, e.g. an extended Kalman filter [20], or sparse Information filter [18, 19, 21], and are refined incrementally. The experimental results show that a large-scale map could benefit from joining together local submaps. Zhao et al. [21] provide a map-joining algorithm through solving a sequence of linear least squares problems. Their experimental results are very close to the best solutions obtained by full nonlinear optimization,

if an accurate initial value is known. They use the submaps already available in the dataset for their experiments, but how to generate a good initial submap is not touched. Generating high-quality submaps is crucial as it lays the foundation for an effective SLAM system.

In this paper, we present a new SLAM approach for constructing 3D panorama for indoor environments by joining together local submaps. A submap is defined by a vector of camera poses and feature locations and is built along with updates of feature sets. We extract KLT features from the color streams of a Kinect camera and remove invalid features from the feature set. The trajectory of the camera is estimated using the features and is further refined through SBA. In the next step, we treat each submap as an observation and utilize a linear optimization approach to obtain a global map. We explain how to build submaps from publicly available datasets and test the proposed approach using them. Furthermore, we reconstruct the scene to intuitively demonstrate the accuracy of the estimated odometry. The comparison between different SLAM methods for the fr1/desk dataset is shown in Fig. 1.



(a) (b)

Fig. 2 A submap begins from frame 86 and ends at frame 103 of the fr1/desk dataset. (a) The extracted keypoints on frame 86. (b) The tracked keypoints on frame 103. We mark the keypoints with blue crosses.

Our main contribution is that we develop a new visual SLAM framework that builds the global map efficiently by joining together a sequence of submaps with high-quality visual odometry estimated.

The rest of this paper is organized as follows. Section 2 explains the process of building submaps. Joining all submaps together by solving a sequence of linear least squares problems is described in Section 3. In Section 4 some experimental results are shown to evaluate the performance of the proposed method on publicly available datasets. Section 5 concludes the whole paper and highlights future work.

2. FEATURE BASED SUBMAP BUILDING

2.1. Establishing 3D Correspondences

We extract the KLT features [22] on the first color frame, and track them on the following frames. A new submap starts from a frame on which the number of the tracked points is lower than a threshold λ (we use 500 in this experiment) while the previous submap ends. The first frame of each submap is marked as a keyframe. Obviously, every two adjacent submaps share one common frame.

More features need to be extracted from a keyframe for keeping the number of the observations of features in balance. The tracked points appearing on the end frame seem strong and stable because they are visible on any other frame of a submap. Given the depth map, the 3D coordinates of each feature can be computed by means of the intrinsic parameters of the camera. We can thus establish feature correspondences easily.

2.2. Optimizing Pairwise Transformation

Suppose that the camera is placed at the world origin without any rotation at the beginning of a submap. Registering dense point clouds generated by the end frame with those points of the keyframe using ICP [23] derives a rigid transformation T. Using this transformation, we can transform the matched 3D keypoints in the coordinate frame of the end pose to the start pose. A correspondence is regarded as an outlier if the Euclidean distance between a point and the corresponding transformed point exceeds a threshold. Through extensive experiments, we set this threshold to 0.025 meter in our experiments. Consequently, we estimate the pairwise transformation using those inliers in a frame-to-keyframe manner. Furthermore, the location of the observations of features is optimized by minimizing the accumulated reprojection error between the reprojected 3D feature points and the corresponding points on the imaging plane. For more details on SBA, please refer to [24].

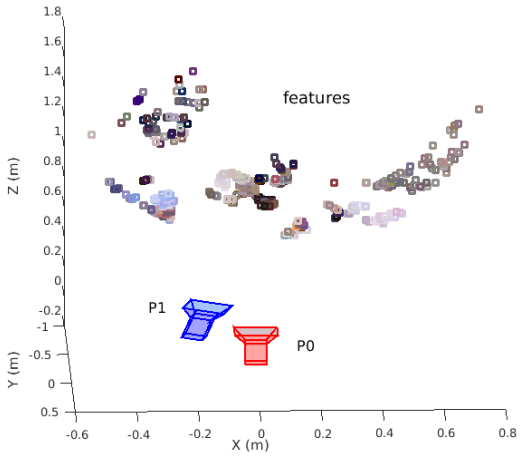


Fig. 3 A submap: P_0 is the start pose, P_1 is the end pose, and features are in the coordinate frame of P_0.

We use SIFT to describe features and compare all valid features of the current local map with the previous ones. New features which have not seen before are assigned a number as identity and are added to the feature set.

In this way, we build a submap which consists of camera poses and a number of features in the world coordinate system. Fig.3 shows the submap built from frame 86 to frame 103 of the fr1/desk dataset.

3. LINEAR SOLUTION TO SUBMAP JOINING

Let M^{L_1} and M^{L_2} be two consecutive submaps, respectively. We discuss how to align them efficiently in this part.

3.1. Traditional Ways of Submap Joining

We use P_0, P_1, and P_2 to represent the start pose of M^{L_1}, the end pose of M^{L_1} (i.e. the start pose of M^{L_2}), and the end pose of M^{L_2}, separately. $M^{G_{12}}$ denotes the extended map. The traditional way to build $M^{G_{12}}$ is based on the common features that are present on both M^{L_1} and M^{L_2}. If few or none common features exist, M^{L_2} can be directly transformed into the coordinate frame of P_0. This process is easy-to-implement, but inherently prone to drift. The goal of joining together every two adjacent submaps can be achieved by solving a sequence of nonlinear optimization problems [18]. For pose-based submap joining problem, the g^{2}o framework [10] can be further used to fuse together those local SLAM results [9, 13]. Although these methods seem to work well on most benchmark datasets, a good initialization with an accurate value cannot be guaranteed. This severely limits their practical application.

3.2. A New Way of Submap Joining

The linear SLAM [21] inspires us to develop a novel way of submap joining. It provides a linear solution which performs well by fusing together two local maps in the same coordinate system. Both M^{L_1} and M^{L_2} are defined in the coordinate system of P_1,

$$M^{L_1} = \left(\hat{\mathrm{X}}^{L_1}, I^{L_1}\right), \quad M^{L_2} = \left(\hat{\mathrm{X}}^{L_2}, I^{L_2}\right) \tag{2}$$

where $\hat{\mathrm{X}}^{L_1}$ and $\hat{\mathrm{X}}^{L_2}$ are the estimates of the state vectors X^{L_1} and X^{L_2}, and I^{L_1} and I^{L_2} are the associated information matrices.

$$\begin{aligned} \mathrm{X}^{L_1} &= \left[\mathrm{P}_0^{L_1}, \mathrm{X}_{F_1}^{L_1}, \mathrm{X}_{F_{12}}^{L_1}\right] \\ &= [t_0^{L_1}, r_0^{L_1}, \mathrm{X}_{F_1}^{L_1}, \mathrm{X}_{F_{12}}^{L_1}] \end{aligned} \tag{3}$$

$$\begin{aligned} \mathrm{X}^{L_2} &= \left[\mathrm{P}_2^{L_2}, \mathrm{X}_{F_2}^{L_2}, \mathrm{X}_{F_{12}}^{L_2}\right] \\ &= [t_2^{L_2}, r_2^{L_2}, \mathrm{X}_{F_2}^{L_2}, \mathrm{X}_{F_{12}}^{L_2}] \end{aligned} \tag{4}$$

$\mathrm{X}_{F_1}^{L_1}$ and $\mathrm{X}_{F_2}^{L_2}$ denote the features that appear in M^{L_1} or M^{L_2}, respectively, while $\mathrm{X}_{F_{12}}^{L_1}$ and $\mathrm{X}_{F_{12}}^{L_2}$ represent the common features that are visible in the two submaps. $t_0^{L_1}$ and $t_2^{L_2}$ denote the translation vector, and $r_0^{L_1}$ and $r_2^{L_2}$ denote the rotation angles of P_0 and P_2, respectively.

Our goal is to obtain the state vector of $M^{G_{12}}$ which is defined as

$$\mathrm{X}^{G_{12}} = \left[t_0^{G_{12}}, r_0^{G_{12}}, t_1^{G_{12}}, r_1^{G_{12}}, \mathrm{X}_{F_1}^{G_{12}}, \mathrm{X}_{F_2}^{G_{12}}, \mathrm{X}_{F_{12}}^{G_{12}}\right] \tag{5}$$

Note that $\mathrm{X}^{G_{12}}$ is presented in the coordinate frame of P_2.

If we transform poses and features in $\mathrm{X}^{G_{12}}$ into the coordinate frame of P_1 through a coordinate transformation function g, the new state vector is denoted as

$$\begin{aligned} \bar{\mathrm{X}}^{G_{12}} &= g\left(\mathrm{X}^{G_{12}}\right) \\ &= \left[\bar{t}_0^{G_{12}}, \bar{r}_0^{G_{12}}, \bar{t}_2^{G_{12}}, \bar{r}_2^{G_{12}}, \bar{\mathrm{X}}_{F_1}^{G_{12}}, \bar{\mathrm{X}}_{F_2}^{G_{12}}, \bar{\mathrm{X}}_{F_{12}}^{G_{12}}\right] \end{aligned} \tag{6}$$

where the differences between $\bar{\mathrm{X}}^{G_{12}}$ and $\hat{\mathrm{X}}^{L_1}$, $\hat{\mathrm{X}}^{L_2}$ could be measured. The task of aligning M^{L_1} and M^{L_2} is therefore transformed to a linear least squares problem which can be solved efficiently by minimizing the following objective function

$$f\left(\bar{\mathrm{X}}^{G_{12}}\right) = \left\| \begin{matrix} \bar{t}_0^{G_{12}} - \hat{t}_0^{L_1} \\ \bar{r}_0^{G_{12}} - \hat{r}_0^{L_1} \\ \bar{\mathrm{X}}_{F_1}^{G_{12}} - \hat{\mathrm{X}}_{F_1}^{L_1} \\ \bar{\mathrm{X}}_{F_{12}}^{G_{12}} - \hat{\mathrm{X}}_{F_{12}}^{L_1} \end{matrix} \right\|_{I^{L_1}}^2 + \left\| \begin{matrix} \bar{t}_2^{G_{12}} - \hat{t}_2^{L_2} \\ \bar{r}_2^{G_{12}} - \hat{r}_2^{L_2} \\ \bar{\mathrm{X}}_{F_2}^{G_{12}} - \hat{\mathrm{X}}_{F_2}^{L_2} \\ \bar{\mathrm{X}}_{F_{12}}^{G_{12}} - \hat{\mathrm{X}}_{F_{12}}^{L_2} \end{matrix} \right\|_{I^{L_2}}^2 \tag{7}$$

Then the optimal solution can be obtained by

$$\hat{\mathrm{X}}^{G_{12}} = g^{-1}\left(\hat{\bar{\mathrm{X}}}^{G_{12}}\right) \tag{8}$$

It should be noted that the submaps we build in Section 3.1 are defined in the coordinate frame of the start pose. A coordinate transformation as pre-processing is thus needed for some submaps. Then we obtain the state vector of the global map after joining all the submaps together.

4. EXPERIMENTS AND EVALUATION

We use the RGB-D benchmark datasets [11] to evaluate our approach. The datasets provide ground truth and an evaluation tool to compute the root mean square error (RMSE) of drift.

4.1. Drift Evaluation

We test our approach using nine datasets. We extract the camera poses of keyframes from the state vector of the global map of each dataset and compute the Relative Pose Error (RPE) by using ground truth as the benchmark. The results are shown in Table I.

The experiments show that our method yields an average drift of 0.054 m/s and can deal with camera velocities of up to 50 deg/s and 0.43m/s in common indoor scenarios. Note that, the relatively high drift value for fr1/360 and fr1/floor

is due to the mistakes of assigning identity numbers to features. The ID of features plays a crucial role in correcting drift. It facilitates the detection of a loop. That is to say, if a lot of features which already exist in the feature set reappear, a loop closure is formed.

TABLE I: RMSE of translational drift in m/s for odometry achieved by the proposed method on all fr1 datasets.

Dataset Name	Submap Number	Avg. Angular Velocity	Avg. Transl. Velocity	Transl. RMSE
fr1 rpy	153	50.15 deg/s	0.06 m/s	0.020
fr1 xyz	119	8.92 deg/s	0.24 m/s	0.013
fr1 360	268	41.60 deg/s	0.21 m/s	0.090
fr1 desk	137	23.33 deg/s	0.41 m/s	0.029
fr1 desk2	152	29.31 deg/s	0.43 m/s	0.056
fr1 room	314	29.88 deg/s	0.33 m/s	0.079
fr1 floor	549	15.07 deg/s	0.26 m/s	0.089
fr1 plant	473	27.89 deg/s	0.37 m/s	0.066
fr1 teddy	502	21.32 deg/s	0.32 m/s	0.047

4.2. Scene Reconstruction by different methods

To further evaluate the accuracy of the estimated odometry, we reconstruct the scene for the fr1/desk dataset. Fig. 4 shows the reconstruction results using different methods for a scene in this dataset.

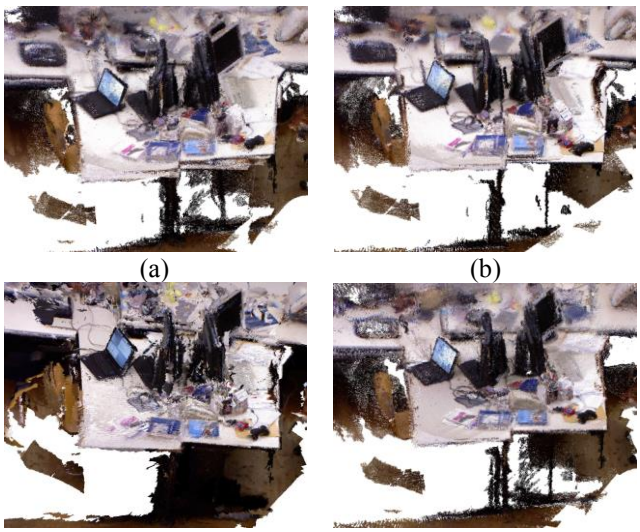


Fig. 4 Reconstruction results. (a) The result by joining together submaps directly without optimization. (b) Our result. (c) Using the SLAM method by Endres et al. (d) Ground truth.

As can be seen from Fig. 4 (a), joining together submaps directly without any optimization leads to severe mis-alignment. The reconstructed scene looks messy. Our result (Fig. 4 (b)) is better than the result (Fig. 4 (c)) generated by using the SLAM method [9]. It is obvious that our approach yields a result that looks similar to the ground truth.

5. CONCLUSIONS AND FUTURE WORK

We have presented a novel approach for building feature-based local submaps and joining together these maps to obtain the 3D panorama. Our experiments on publicly available benchmark datasets demonstrate that the proposed approach is capable of dealing with camera velocities of up to 50 deg/s and 0.43m/s in common indoor scenarios, and obtains even more stable visual odometry with an average drift lower than 0.054 m/s. The 3D panorama we generate is comparable to the ground truth.

In the future, we plan to further improve the quality of the 3D panorama by fully leveraging each captured raw RGB-D frame. In addition, we intend to convert the point cloud representation of the final 3D panorama into the high-quality surfel representation.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of the US National Science Foundation (Grant Number CCF-1218682), the National Natural Science Foundation of China (Key Project Number 61332015, Project Number 61373059 and 61321491) and the Natural Science Foundation of Jiangsu (Grant Number BK20150016).

7. REFERENCES

[1] F. Dellaert, "Square Root SAM," In *Proc. of Robotics: Science and Systems (RSS)*, pages 177-184, 2005.

[2] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, "Efficient estimation of accurate maximum likelihood maps in 3d," In *Proc. of the Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[3] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2262-2269, 2006.

[4] A. Hornung, K. M. Wurm, and M. Bennewitz, "Humanoid robot localization in complex indoor environments," In *Proc. of the IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[5] G. Klein, and D. Murra, "Parallel tracking and mapping for small AR workspaces," In *IEEE and ACM Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, 2007.

[6] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "3-D motion and structure from 2-D motion causally integrated over time:

Implementation," In *European Conf. on Computer Vision (ECCV)*, 2000.

[7] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," In *Intl. Symp. of Robotics and Research (ISRR)*, 2011.

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," In *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 6, pages 1052-1067, 2007.

[9] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D system," In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.

[10] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.

[11] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, and R. Siegwart, "Towards a benchmark for RGB-D SLAM evaluation," In *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf. (RSS)*, 2011.

[12] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," In *Proc. of the Intl. Symp. on Experimental Robotics (ISER)*, 2010.

[13] C. Kerl, J. Sturm, and D. Cremers, "Dense Visual SLAM for RGB-D Cameras," In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2013.

[14] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," In *Proc. of Robotics: Science and Systems (RSS)*, vol.25, pages 26-27, 2009.

[15] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," In *IEEE and ACM Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, 2011.

[16] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," In *Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV)*, 2011.

[17] L. P. Morency, and T. Darrell, "Stereo tracking using icp and normal flow constraint," In *IEEE Intl. Conf. on Pattern Recognition*, 2002.

[18] S. Huang, Z. Wang, G. Dissanayake, and U. Frese, "Iterated SLSJF: A sparse local submap joining algorithm with improved consistency," In *Australiasan Conf. on Robotics and Automation*, 2008.

[19] S. Huang, Z. Wang, and G. Dissanayake, "Sparse local submap joining filter for building large-scale maps," In *IEEE Trans. on Robotics*, 2008.

[20] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-Point RANSAC for EKF-based structure from motion," In *IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[21] L. Zhao, S. Huang, and G. Dissanayake, "Linear SLAM: A Linear Solution to the Feature-based, Pose Graph and D-SLAM based on Submap Joining," In *IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[22] J. Shi, and C. Tomasi, "Good features to track," In *IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1994.

[23] P. J. Besl, and N. D. McKay, "A method for registration of 3-D shapes," In *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 14, no. 2, pages 239-256, 1992.

[24] M. I. A. Lourakis, and A. A. Argyros, "SBA: a software package for generic sparse bundle adjustment," In *ACM Trans. on Mathematical Software*, 2009.