Submap-based Bundle Adjustment for 3D Reconstruction from RGB-D Data

Robert Maier, Jürgen Sturm, Daniel Cremers

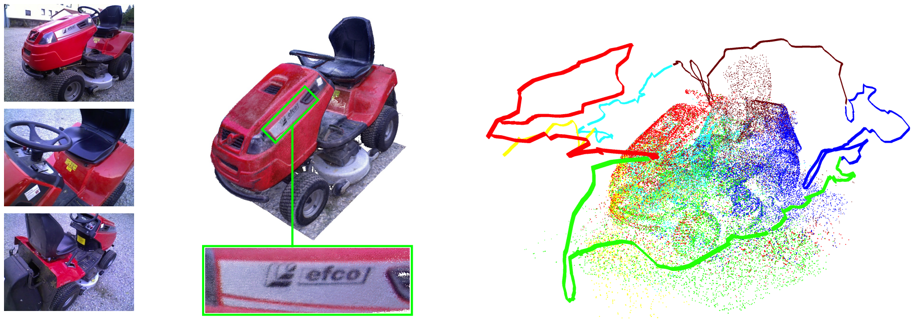
TU Munich, Germany
{maierr,sturmju,cremers}@in.tum.de

Abstract. The key contribution of this paper is a novel submapping technique for RGB-D-based bundle adjustment. Our approach significantly speeds up 3D object reconstruction with respect to full bundle adjustment while generating visually compelling 3D models of high metric accuracy. While submapping has been explored previously for mono and stereo cameras, we are the first to transfer and adapt this concept to RGB-D sensors and to provide a detailed analysis of the resulting gain. In our approach, we partition the input data uniformly into submaps to optimize them individually by minimizing the 3D alignment error. Subsequently, we fix the interior variables and optimize only over the separator variables between the submaps. As we demonstrate in this paper, our method reduces the runtime of full bundle adjustment by 32% on average while still being able to deal with real-world noise of cheap commodity sensors. We evaluated our method on a large number of benchmark datasets, and found that we outperform several state-of-the-art approaches both in terms of speed and accuracy. Furthermore, we present highly accurate 3D reconstructions of various objects to demonstrate the validity of our approach.

1 Introduction

Low-cost sensors such as the Microsoft Kinect have boosted research in 3D reconstruction and enabled novel applications such as product digitalization, remote inspection and assessment, documentation and reverse-engineering. For example, Figure 1 shows a colored 3D model of a lawn-mower generated from hand-held RGB-D sensor data with the approach proposed in this paper. The resulting 3D model is highly accurate. It preserves small details such as the plate in the close-up, it allows for metric measurements and can be used to detect dents and deformations.

In all of these applications, it is important that the 3D model is highly accurate, yet the reconstruction process must be fast enough to be truly useful in practice. While volumetric methods such as KinectFusion [4, 17] are tailored for real-time use, they are inherently prone to drift because they do not impose any long-range consistency. This may lead to inconsistencies such as the formation of double surfaces and blurry textures. Global optimization techniques such as bundle adjustment [8, 10] achieve higher accuracies. Yet, they drastically



(a) Input images

(b) Resulting 3D model and close-up

(c) Submap visualization with colorized camera poses and landmarks

Fig. 1: We increase the efficiency of 3D bundle adjustment significantly with a novel submapping approach for RGB-D data.

increase the computation time, scaling poorly with the number of images and 3D points. Submapping techniques offer a remedy to this problem. They divide the large optimization problem into smaller, independent subproblems through marginalization [20, 21]. Unfortunately, this strategy only works well when the graph is sparsely connected. For the reconstruction of an object in the center, the challenge considered in this work, many overlaps exist and thus the resulting graph is densely connected. This renders full bundle adjustment slow and existing submapping techniques inefficient.

To cope with these challenges, we developed a novel submapping technique for feature-based 3D reconstruction that significantly speeds up map optimization. Moreover, we provide an experimental study on the trade-off between reconstruction accuracy and computation speed on public benchmark sequences. We finally show several examples of 3D scans demonstrating the accuracy that can be achieved using our approach. To the best of our knowledge, we are the first to introduce submapping into an RGB-D based reconstruction method and to demonstrate its efficiency in the context of 3D object scanning.

2 Related Work

Real-time 3D reconstruction from images has a long history in computer vision [2, 5, 7, 10, 13, 17]. We focus our review of previous work on approaches using a single, hand-held sensor. In this scenario, called Visual SLAM, both the pose of the camera and the 3D model need to be estimated. Typically, visual features such as SIFT are extracted and matched across the frames. From this, a SLAM graph can be constructed whose vertices correspond to 6-DOF camera poses and 3D landmark positions and whose edges correspond to the observations. Bundle adjustment (BA) [26] optimizes over camera poses and landmarks using non-linear minimization. While impressive reconstructions can be obtained using BA [2], bad initialization and the scale ambiguity can lead to slow convergence, and the computational effort typically grows cubically with the number of cameras and landmarks. Sparse Bundle Adjustment [16] exploits the fact that 3D landmarks are usually not visible in all of the cameras. When instead stereo cameras or RGB-D sensors are used, the scale ambiguity can be resolved and the initialization is simplified [10, 11]. Moreover, the absolute camera motion between two frames can be estimated directly. By marginalization over all landmarks, the problem can then be reduced to pose graph SLAM [8, 10, 14], which is however still cubic in the number of cameras.

Submap-based BA methods [12, 15, 20, 21] aim at partitioning the full BA problem into several smaller submaps, which are connected among each other and optimized independently in an efficient manner. By expressing the camera poses and landmarks relative to a base node for every submap [23], the error per submap is bounded which generally improves convergence. Advanced techniques for graph partitioning such as nested dissection have been proposed [19]. After the individual submaps have been optimized independently, a global optimization over the submap separator variables is carried out and the whole process is iterated. In this work, we employ submapping to pair the high accuracy of full BA with the improved efficiency.

After map optimization, a 3D model can be generated from the data. Common representations for 3D models include point clouds, surfels [10], triangle meshes [27], and signed distance functions [6]. Octrees allow for efficient data storage and fusion at multiple scales [9, 24, 28]. While KinectFusion [17] demonstrated that impressive 3D models can be acquired by tracking and fusing the depth images directly into a signed distance volume, drift will accumulate in the 3D model and inevitably lead to inconsistencies. Several approaches have been proposed to mitigate these effects in post-processing [27, 29, 30], however at the cost of having different cost functions at different optimization stages, which is not desirable from a theoretical point of view. Therefore, we propose to use BA with a single cost function to achieve global consistency.

3 3D Object Reconstruction Pipeline

We developed a feature-based 3D reconstruction system similar to [11]. Figure 2 depicts a schematic overview.

We acquire RGB-D data of the object with an off-the-shelf ASUS Xtion Pro Live sensor. To account for increasing sensor noise with distance, we pre-process the acquired depth map with a bilateral filter and cut off large depth values.

Our approach uses frame-to-frame camera tracking to estimate the absolute camera poses for the acquired RGB-D frames by concatenating the relative camera motion. We estimate the relative pose of every frame w.r.t. its predecessor using a robust feature-based 3D alignment algorithm based on the method of Arun [3] and RANSAC.

We use a graph-based map representation. The nodes of the graph correspond to the variables, i.e. camera poses $C_i \in SE(3)$ (with $i \in 1 \dots M$) and

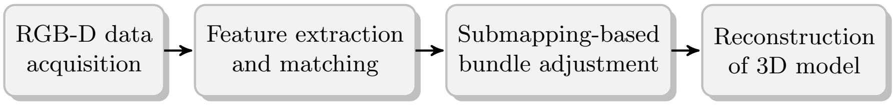

RGB-D data acquisition
Feature extraction and matching
Submapping-based bundle adjustment
Reconstruction of 3D model

Fig. 2: Processing pipeline of our 3D object reconstruction approach.

3D landmarks $\mathbf{X}_j \in \mathbb{R}^3$ (with $j \in 1 \dots N$). The edges of the graph represent landmark observations $k \in 1 \dots K$, i.e. $\mathbf{z}_k = (u_k, v_k, d_k)^\top \in \mathbb{R}^3$, where (u_k, v_k) is the observed 2D pixel coordinate and d_k is the observed depth.

To further reduce the drift in the graph, we detect loop closures by performing a 3D alignment of the current frame with 20 uniformly sampled previous frames. After all frames have been processed, we perform bundle adjustment to optimize over the camera poses and landmark positions. More details on this will be provided in Section 4 and 5.

After optimization, we use the refined camera poses to generate a dense colored 3D point cloud by fusing the RGB-D frames into an octree-based 3D model representation.

4 Full Bundle Adjustment for RGB-D Sensors

Our approach on 3D bundle adjustment is inspired by the work of Henry et al. [10], that we extended by integrating depth measurements from the RGB-D sensor as additional constraints similar to Scherer et al. [22]. The optimization goal becomes then to minimize the 3D alignment error instead of the 2D projection error.

Rigid body motion The camera parameters C_i to be optimized are 3D Euclidean transformations $g = (R, \mathbf{t}) \in SE(3)$, with translation vector $\mathbf{t} \in \mathbb{R}^3$ and rotation $R \in SO(3)$. To have a minimum number of parameters to be optimized, we represent the camera poses C_i with their twist coordinate representations $\boldsymbol{\xi}_i = (\omega_1, \omega_2, \omega_3,\ v_1, v_2, v_3)^\top \in \mathbb{R}^6$. While this representation is important for the efficiency of solving the NLS problem, we use in the following the notation $C_i \in SE(3)$ to denote the camera poses.

We define the transformations $\mathcal{T}(g, \mathbf{X})$ and $\mathfrak{T}(g, \tilde{g})$ for transforming 3D points $\mathbf{X} \in \mathbb{R}^3$ and transformations $\tilde{g} \in SE(3)$, respectively.

Camera model We use the basic pinhole camera model to describe the mapping of 3D points of a three-dimensional scene onto a 2D image plane. The projection function π maps a 3D point $\mathbf{X} = (x, y, z)^\top \in \mathbb{R}^3$ to a 2D image point $\mathbf{x} \in \mathbb{R}^2$. If a depth value d for a 2D image point $\mathbf{x} = (u, v)^\top \in \mathbb{R}^2$ is given, the backprojection of $\mathbf{x}$ to a 3D point $\mathbf{X} \in \mathbb{R}^3$ in the camera coordinate frame is given by:

$$\mathbf{X} = \rho(u, v, d) = \left(\tfrac{(u - c_x)d}{f_x}, \tfrac{(v - c_y)d}{f_y}, d \right)^\top , \tag{1}$$

with focal lengths f_x, f_y and coordinates of the camera center c_x, c_y.

2D reprojection error Regular bundle adjustment refines both camera poses and 3D structure of the scene by minimizing the 2D reprojection error. This error is the difference between the actual 2D measurement $\bar{\mathbf{z}}_k = (u_k, v_k)^\top \in \mathbb{R}^2$ of a 3D landmark in an image and its predicted 2D projection based on the current estimates of the respective camera pose and landmark. The error function minimized in (2D) bundle adjustment is usually defined as

$$\sum_{k=1}^{K} ||h_k(C_{i_k}, \mathbf{X}_{j_k}) - \bar{\mathbf{z}}_k||^2, \text{ with } h_k(C_{i_k}, \mathbf{X}_{j_k}) = \pi(\mathcal{T}^{-1}(C_{i_k}, \mathbf{X}_{j_k})). \quad (2)$$

where C_{i_k} and $\mathbf{X}_{j_k}$ are the variables to be optimized. We use the subscript k here to indicate that C_{i_k} and $\mathbf{X}_{j_k}$ are related by measurement k.

3D alignment error To utilize the depth data provided by the RGB-D camera, we integrate depth measurements as additional constraints into bundle adjustment to improve accuracy, robustness and convergence behavior. We compute the 3D position of a landmark from its measurement $\mathbf{z}_k = (u_k, v_k, d_k)^\top$ as

$$\mathbf{Z}_k = \rho(u_k, v_k, d_k) \in \mathbb{R}^3. \quad (3)$$

This allows us to directly minimize the 3D alignment error, which is defined as the difference between the measured and the predicted 3D position of a 3D landmark in the local camera coordinate frame. The prediction is accomplished by transforming the landmark $\mathbf{X}_{j_k}$ in global coordinates back into the local camera coordinate frame using the respective absolute camera pose C_{i_k}. With this, we define the error function as

$$\sum_{k=1}^{K} ||\hat{h}_k(C_{i_k}, \mathbf{X}_{j_k}) - \mathbf{Z}_k||^2, \text{ with } \hat{h}_k(C_{i_k}, \mathbf{X}_{j_k}) = \mathcal{T}^{-1}(C_{i_k}, \mathbf{X}_{j_k}). \quad (4)$$

The solution to this kind of NLS optimization problem is well-investigated and can be computed by applying a sparse LM algorithm. In our implementation, we employ the *Ceres Solver* [1] and *CXSparse* for solving bundle adjustment problems with the above cost functions.

5 Efficient Bundle Adjustment for RGB-D Sensors using Submapping

With the large amount of data that arises from commodity RGB-D sensors, full bundle adjustment with its high computational complexity quickly becomes intractable due to the large time and memory consumption. In order to make the optimization more efficient, we propose a novel submapping technique tailored to RGB-D sensors, consisting of the following four processing stages:

1. Partition the RGB-D bundle adjustment graph into submaps.

2. Optimize each submap individually.
3. Align the submaps globally.
4. Optimize each submap internally with fixed separator variables.

Note that in contrast to previous work, our approach applies submapping to the 3D (RGB-D) bundle adjustment problem (instead of 2D). In this way, we can achieve high reconstruction accuracies while keeping the computational complexity small. In the following, we present each step in more detail.

Graph partitioning into submaps To facilitate a submap-based optimization approach, the full optimization graph is first partitioned into L submaps. While challenging unordered image sets require advanced graph partitioning techniques [18], we apply a uniform partitioning scheme to achieve spatially coherent partitions, i.e. frames in the same partition have many common features. We split the input trajectory into segments of equal size $\tilde{M} = M/L$. To every submap, we assign a base node $B_l \in SE(3)$ with $l \in 1 \ldots L$ and initialize it to the first contained camera pose such that $B_l = C_{((l-1)\tilde{M}+1)}$. When the submaps are populated, all camera poses and landmarks are parametrized relative to the base node of the respective submap:

$$\tilde{C}_i^l = \mathfrak{T}^{-1}(B_l, C_i), \quad \tilde{\mathbf{X}}_j^l = \mathcal{T}^{-1}(B_l, \mathbf{X}_j) \tag{5}$$

We can directly use the measurements as submap measurements $\tilde{\mathbf{Z}}_k^l = \mathbf{Z}_k$.

Submap optimization We optimize the submaps independently in the second stage to achieve local consistency. The bundle adjustment problem in each submap consists of its camera poses $\tilde{C}_l$, landmarks $\tilde{\mathbf{X}}_l$ and measurements $\tilde{\mathbf{Z}}_l$. Hence, we have to solve L small optimization problems instead of one large problem. We perform the optimization of submap l by minimizing the 3D alignment error as defined in equation 4 using bundle adjustment:

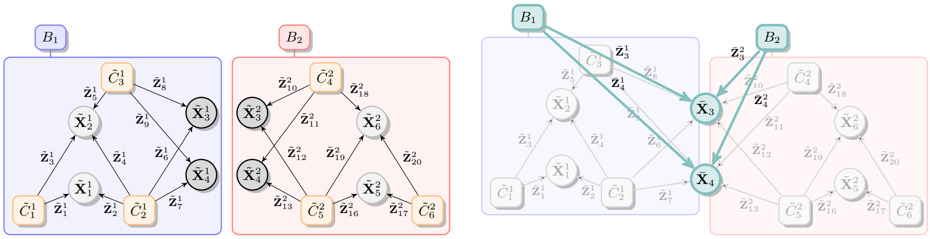
$$\sum_{k=1}^{K_l} ||\hat{h}_k(\tilde{C}_{i_k}^l, \tilde{\mathbf{X}}_{j_k}^l) - \tilde{\mathbf{Z}}_k^l||^2, \tag{6}$$

where K_l is the number of measurements in submap l. After the variables $\tilde{C}_i^l$ and $\tilde{\mathbf{X}}_j^l$ in all submaps have converged, we obtain an optimized local reconstruction for each submap relative to its base node B_l as depicted in Figure 3a.

Global submaps alignment If a landmark of a submap is seen by a camera pose contained in another submap, it is considered a separator landmark and expressed in global world coordinates:

$$\bar{\mathbf{X}}_j^l = \mathcal{T}(B_l, \tilde{\mathbf{X}}_j^l). \tag{7}$$

Further, we introduce inter-measurements $\bar{\mathbf{Z}}_k^l = \tilde{\mathbf{X}}_{j_k}^l$ between the submaps, which are the locations of $\bar{\mathbf{X}}_j^l$ relative to B_l. We make this approximation since



(a) (b)

Fig. 3: Example of a SLAM graph consisting of 6 camera poses, 6 landmarks and 20 observations. (a) The map partitioned into two submaps. All camera poses and landmarks are expressed relative to their assigned base node, B_1 and B_2, respectively. (b) The global submaps alignment consists only of the base nodes and separator landmarks. The locations of the separator landmarks relative to their connected submaps' base nodes are used as measurements.

we assume the relative locations of the submap landmarks to be optimal relative to the base nodes due to the performed local optimization.

After the internal submap alignment, we eliminate the global drift by moving the base nodes. Here, the local coordinate systems in the submaps play an important role, as the landmarks move with their base node and hence keep their locally optimal values from the previous stage. In the global alignment stage, we perform a full optimization of a graph consisting only of the base nodes and the separator landmarks as vertices, connected by the inter-measurements as edges as illustrated in Figure 3b. We omit the internal landmarks and camera poses, as they have already been used to determine the optimal relative locations of the separator landmarks in each submap. In particular, we minimize

$$\sum_{k=1}^{\bar{K}} ||\hat{h}_k(B_{l_k}, \bar{\mathbf{X}}_{j_k}) - \bar{\mathbf{Z}}_k^l||^2, \tag{8}$$

over B_l and $\bar{\mathbf{X}}_j$, where $\bar{K}$ is the number of inter-measurements of all submaps.

With the optimization graph consisting only of base nodes and a limited number of separator landmarks, the global optimization itself can be computed very efficiently. After this separator optimization, we obtain the refined poses of the base nodes and locations of the separator landmarks with respect to the global world coordinate system. The effect of moving the base node results in a reduced global drift between the submaps and a globally consistent 3D model.

Internal submap update After the global alignment stage, the locations of the separator landmarks may have changed relative to the base nodes. Therefore, we optimize in the fourth stage the internal landmarks given the refined separator landmarks. To this end, we first update the relative locations of the changed separator landmarks in each submap with the changed global locations:

$$\tilde{\mathbf{X}}_k^l = \mathcal{T}^{-1}(B_l, \bar{\mathbf{X}}_k^l). \tag{9}$$

Subsequently, we perform bundle adjustment with the same input as in stage 1, by minimizing the error function defined in equation 6 while keeping the separator landmarks fixed. This is illustrated again in Figure 3a, where the bold nodes $\tilde{\mathbf{X}}_3^1, \tilde{\mathbf{X}}_4^1, \tilde{\mathbf{X}}_3^2, \tilde{\mathbf{X}}_4^2$ are the separator landmarks. This optimization step allows the internal camera poses and landmarks to optimally fit to the separator while the separator landmarks stay in the same position. This usually also results in quick convergence and small movements of the internal poses and landmarks, since they were already optimized w.r.t. the base node's coordinate system.

In theory, it is reasonable to iterate over the third and the fourth step of the algorithm until convergence. In practice, we found that the results did not change significantly compared to only a single iteration.

To finally get the refined absolute camera poses and 3D landmark locations, we transform the content of each submap back into the global coordinate frame:

$$C_i = \mathfrak{T}(B_l, \tilde{C}_i^l) \quad \text{and} \quad \mathbf{X}_j = \mathcal{T}(B_l, \tilde{\mathbf{X}}_j^l) \tag{10}$$

Finally, we use the computed camera poses C_i to fuse the RGB-D frames into the 3D octree model as explained in Section 3.

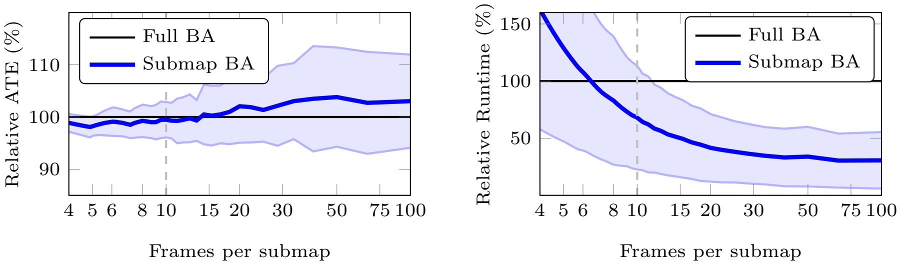
6 Evaluation and Experimental Results

The following experimental results demonstrate that (1) 3D bundle adjustment leads to a significant improvement in accuracy over 2D bundle adjustment, pose graph optimization and other techniques, (2) submapping strongly reduces the runtime while yielding a comparable accuracy, and (3) accurate 3D models can be acquired of various real agricultural vehicles.

We evaluated our approach on the TUM RGB-D benchmark [25] to allow for a quantitative comparison of its performance with other state-of-the-art methods.

Size of submaps First, we evaluated the performance of our algorithm over different submap sizes to find the right trade-off between speed versus accuracy. For this, we computed the Absolute Trajectory Error (ATE) between the estimated and the ground-truth camera trajectory as defined in [25]. We evaluated the ATE over 10 sequences from the TUM RGB-D benchmark and computed its mean and standard deviation.

Figure 4 gives the result. We found that for small submap sizes, the ATE was significantly lower than the ATE of full 3D bundle adjustment. Because bundle adjustment is intrinsically prone to local minima, full (2D and 3D) bundle adjustment sometimes cannot converge from bad initial estimates. As the internal submap optimization in stage 2 already establishes local consistency before the global alignment, submapping can lead to better convergence and helps to find



(a) ATE

(b) Runtime

Fig. 4: Performance evaluation of submapping on 10 sequences. (a) Smaller submaps lead to more accurate reconstructions, while (b) larger submaps are more efficient. The black line indicates the average performance of full bundle adjustment (no submapping), while the blue line shows the average performance of our approach w.r.t. to the submap size. The shaded area corresponds to the standard deviation over all benchmark sequences.

a better solution, in particular for small submap sizes. In contrast, large submap sizes increase the efficiency but decrease accuracy.

As a good compromise between speed and accuracy, we found a submap size of 10 frames to be a reasonable choice that we used in all subsequent experiments. This choice is also indicated by the dashed vertical line in the plot. In values, this submap size leads both to a lower error (-0.5%) and faster computation (-30%).

Benchmark Sequences To study in more detail the effect of our cost function, we evaluated the performance of full bundle adjustment using the 2D reprojection error as well as using the 3D alignment error over 10 sequences of the TUM RGB-D benchmark and compared it to our approach. Furthermore, we also compared the performance to the RGB-D SLAM system [8] and a KinectFusion-based implementation [4] using the values reported in the respective publications.

On average, 2D bundle adjustment yields an ATE of 0.066m, while 3D bundle adjustment reduces this error to 0.047m (-29%). At the chosen setting of 10 frames per submap, our approach achieves a similar accuracy at a significantly reduced computational cost (-32%). Furthermore, the efficiency of bundle adjustment can be improved most for long sequences, with an improvement of up to 84% of the runtime of full bundle adjustment.

In direct comparison to existing approaches, 3D bundle adjustment outperforms the pose graph RGB-D SLAM method by 13% (0.047m vs. 0.054m) and direct SDF tracking by 17% (0.047m vs. 0.058m). We believe that the higher accuracy of 3D BA in comparison to the RGB-D SLAM system stems from the fact that RGB-D SLAM performs a simplified pose-graph optimization. KinectFusion-based methods iteratively integrate the new depth image into the signed distance function and therefore inevitably accumulate drift.

Examples of Submap-Based 3D Reconstructions After we have demonstrated the validity of our framework, we use it to generate 3D reconstructions of three different agricultural machines. The acquired datasets show a soil auger of dimensions $1 \times 1 \times 3$m, a lawn mower of size $1 \times 1 \times 2$m and an older model of a Renault farm tractor with dimensions $1 \times 1 \times 3$m. The reconstructed 3D models are illustrated in Figures 1 and 5, together with subsets of the acquired RGB images. The generated reconstructions have a compelling visual quality and metric accuracy, which is supported by the fact that no drift is visible in the models. This makes the reconstructions suitable for visual inspection, measuring tasks and reverse-engineering.

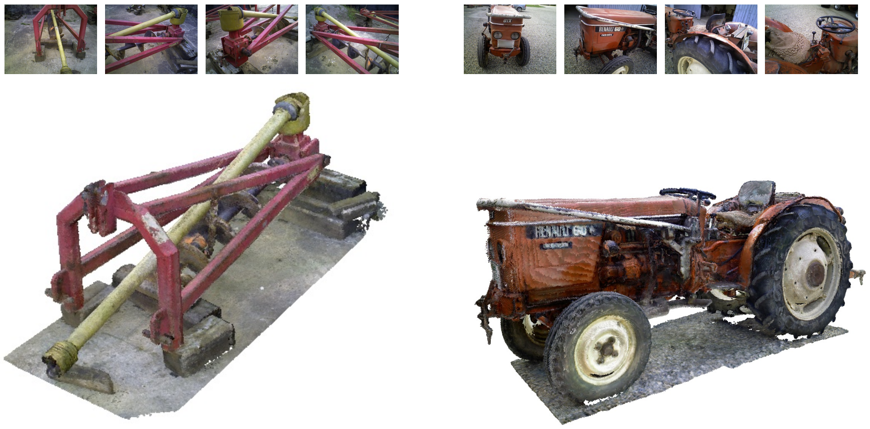


Fig. 5: Left: 3D reconstruction of a soil auger, reconstructed from 2349 RGB-D frames. Right: 3D model of a tractor, reconstructed from 2087 RGB-D frames.

7 Conclusion and Future Work

We presented a novel method for 3D object reconstruction from RGB-D data that applies submapping to 3D bundle adjustment. In contrast to prior work, we thereby fully exploit the available depth information during optimization while maintaining efficiency. In an extensive quantitative evaluation on publicly available datasets, we demonstrated that our approach reduces the average runtime by 32% compared to full bundle adjustment while achieving a similar accuracy. Furthermore, our approach outperforms several state-of-the-art approaches on benchmark datasets with respect to speed and accuracy. The 3D models of various objects reconstructed with our algorithm exhibit a compelling visual quality and metric accuracy, making it suitable for production quality control and reverse-engineering.

References

1. Agarwal, S., Mierle, K.: Ceres Solver: Tutorial & Reference. Google Inc., http://code.google.com/p/ceres-solver/
2. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: ICCV (2009)
3. Arun, K., Huang, T., Blostein, S.: Least-squares fitting of two 3-D point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence (5), 698–700 (1987)
4. Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Real-time camera tracking and 3D reconstruction using signed distance functions. In: RSS (2013)
5. Chiuso, A., Favaro, P., Jin, H., Soatto, S.: 3-D motion and structure from 2-D motion causally integrated over time: Implementation. In: ECCV (2000)
6. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH (1996)
7. Davison, A.: Real-time simultaneous localisation and mapping with a single camera. In: ICCV (2003)
8. Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W.: An evaluation of the RGB-D SLAM system. In: ICRA (2012)
9. Fuhrmann, S., Goesele, M.: Fusion of depth maps with multiple scales. ACM Trans. Graph. 30(6), 148 (2011)
10. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In: ISER. vol. 20, pp. 22–25 (2010)
11. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. International Journal of Robotics Research (IJRR) 31(5), 647–663 (April 2012)
12. Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., Dellaert, F.: iSAM2: Incremental smoothing and mapping using the Bayes tree. International Journal of Robotics Research (IJRR) 31, 217–236 (Feb 2012)
13. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: ISMAR (2007)
14. Konolige, K.: Large-scale map-making. In: AAAI (2004)
15. Lim, J., Frahm, J.M., Pollefeys, M.: Online environment mapping. In: CVPR (2011)
16. Lourakis, M., Argyros, A.: SBA: A software package for generic sparse bundle adjustment. ACM Trans. on Mathematical Software (TOMS) 36(1), 2 (2009)
17. Newcombe, R., Davison, A., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. In: ISMAR (2011)
18. Ni, K., Dellaert, F.: Multi-level submap based SLAM using nested dissection. In: IROS (2010)
19. Ni, K., Dellaert, F.: HyperSfM. In: 3DIMPVT (2012)
20. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3D reconstruction. In: ICCV (2007)
21. Pinies, P., Paz, L., Haner, S., Heyden, A.: Decomposable bundle adjustment using a junction tree. In: ICRA (2012)
22. Scherer, S., Dube, D., Zell, A.: Using depth in visual simultaneous localisation and mapping. In: ICRA (2012)
23. Sibley, G., Mei, C., Reid, I., Newman, P.: Adaptive relative bundle adjustment. In: RSS (2009)

24. Steinbruecker, F., Kerl, C., Sturm, J., Cremers, D.: Large-scale multi-resolution surface reconstruction from RGB-D sequences. In: ICCV (2013)
25. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IROS (2012)
26. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment - a modern synthesis. Vision algorithms: theory and practice pp. 153–177 (2000)
27. Whelan, T., Kaess, M., Leonard, J., McDonald, J.: Deformation-based loop closure for large scale dense RGB-D SLAM. In: IROS (2013)
28. Wurm, K., Hornung, A., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In: ICRA Workshop on best practice in 3D perception and modeling for mobile manipulation (2010)
29. Zhou, Q.Y., Koltun, V.: Dense scene reconstruction with points of interest. In: SIGGRAPH (2013)
30. Zhou, Q.Y., Miller, S., Koltun, V.: Elastic fragments for dense scene reconstruction. In: ICCV (2013)

Supplemental Material for Submap-based Bundle Adjustment for 3D Reconstruction from RGB-D Data

Robert Maier, Jürgen Sturm, Daniel Cremers

TU Munich, Germany
{maierr,sturmju,cremers}@in.tum.de

A. Experimental Results

Section 6 of the paper evaluates the performance of submap-based bundle adjustment w.r.t. accuracy (measured using the Absolute Trajectory Error (ATE)) and efficiency. We evaluated our method in comparison with full bundle adjustment using the 2D reprojection error as well as using the 3D alignment error. Moreover, we compared our approach with other state-of-the-art methods, namely the RGB-D SLAM system [3] and two KinectFusion-based implementations [1, 2] using the values reported in the respective publications. Table 1 shows the detailed results for 10 sequences of the TUM RGB-D benchmark [4] at a chosen setting of 10 frames per submap. All experiments were performed on a desktop PC with Intel Core i7-3770 CPU with 3.40GHz and 8GB RAM.

Sequence	No BA	Full 2D	Full 3D		Submap-based 3D BA					RGB-D SLAM	KinFu	Direct
	ATE	ATE	ATE	time	submaps	ATE	±(%)	time	±(%)	ATE [3]	ATE [1]	ATE [2]
FR1/360	0.108	0.099	0.077	12.66	74	0.079	+3.6	22.62	+78.6	0.079	0.591	0.119
FR1/desk	0.047	0.021	0.022	28.97	57	0.022	-1.5	21.96	-24.2	0.023	0.068	0.035
FR1/desk2	0.098	0.044	0.030	27.23	62	0.031	+3.4	21.36	-21.5	0.043	0.635	0.062
FR1/plant	0.048	0.023	0.042	66.27	112	0.043	+1.7	49.36	-25.5	0.091	0.281	0.043
FR1/room	0.275	0.228	0.085	125.46	135	0.086	+1.7	77.30	-38.4	0.084	0.304	0.078
FR1/rpy	0.046	0.058	0.027	67.56	69	0.027	-1.6	23.69	-64.9	0.026	0.081	0.042
FR1/teddy	0.277	0.060	0.056	67.88	140	0.057	+1.3	68.06	+0.3	0.076	0.337	0.080
FR1/xyz	0.015	0.013	0.013	96.87	79	0.013	-7.9	39.72	-59.0	0.014	0.025	0.023
FR2/desk	0.201	0.080	0.079	2355.26	289	0.076	-3.3	372.20	-84.2	-	-	-
FR3/office	0.176	0.039	0.036	1290.24	248	0.035	-3.0	242.88	-81.2	-	0.061	0.040
average	0.129	0.066	0.047			0.047	-0.5		-32.0	0.054	0.264	0.058

Table 1. Performance evaluation of benchmark sequences.

References

1. KinectFusion Implementation in the Point Cloud Library (PCL). http://svn.pointclouds.org/pcl/trunk/
2. Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Real-time camera tracking and 3D reconstruction using signed distance functions. In: RSS (2013)
3. Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W.: An evaluation of the RGB-D SLAM system. In: ICRA (2012)
4. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IROS (2012)