Bundle Adjustment revisited for Slam with RGBD sensors.

Kathia MELBOUCI [1] Sylvie Naudet Collette [1] Vincent Gay-Bellile [1]
Omar Ait-Aider[2] Mathieu Carrier [1] Michel Dhome [2]

[1] CEA, LIST, Laboratoire Vision et Ingnierie des Contenus
Point Courrier 94, Gif-sur-Yvette, F-91191 France

[2] Clermont Université, Université Blaise Pascal LASMEA
BP 10448 Clermont-Ferrand / CNRS, UMR 6602, LASMEA, AUBIERE

[1] Firstname.lastname@cea.fr [1] Fistname.lastnale@univ-bpclermont.fr

Abstract

We present a method of using depth information provided by an RGB-D sensor, for visual simultaneous localization and mapping (SLAM), in order to improve its accuracy. We present a constraint bundle adjustment which allows to easily combine depth and visual data in cost function entirely expressed in pixel. The proposed approach is evaluated on a public benchmark dataset and compared to the state of art methods.

1 Introduction

In computer vision, the process of estimating camera positions and scene geometry is known as the structure from motion problem (SFM) or Simultaneous Localization and Mapping (SLAM). Often Such methods [9, 6] perform a 3D reconstruction of the primitives extracted and matched in successive images. This reconstruction (usually sparse 3D points map) is used to estimate the relative motion of the camera. In keyframe-based SLAM, the camera poses and the 3D point cloud are simultaneously refined with a Bundle Adjustment (BA). However, visual SLAM suffers from some limitations. Indeed, with a single camera and without any assumptions or prior knowledge about the camera environment, rotation can be retrieved, but translation is up to scale. Furthermore, visual monocular SLAM is an incremental process prone to small drifts in both pose measurement and scale, which when integrated over time, become increasingly significant over large distances. It is possible to determine scaling factor and minimize drift by using a RGB-D camera. This device captures RGB images along with real scale per-pixel depth information. With the introduction of Microsoft's Kinect, that integrates cheap and lightweight 3D sensors with high resolution, research using RGB-D sensors greatly increased [1, 2, 3, 4, 10, 13].

The main contribution of this paper is providing a technique to integrate depth measurements into an existing monocular visual SLAM system. This consists of several rather straightforward changes but also on a way to use depth measurements as additional constraint in bundle adjustment. A similar approach has been proposed by *Scherer et al.* [11, 12] who revisit the PTAM algorithm [6] to integrate depth information. In this paper, we propose to revisit a keyframe based SLAM with temporal local BA [9] which is quite different to the revised PTAM of [11] and to its extension with graph optimization [12]. Compared to [11] and [12], we propose a new cost function in the BA which allows to easily combine depth and visual information without using any additional weight factor like proposed in [11], as it is totally expressed in pixel. We present this new approach and evaluate it on a publicly available RGB-D benchmark [1]. We compare its performances to the previously cited methods [11, 12], and other methods of the state of art.

The rest of this paper is organized as follows. In section 2 we discuss related work. Section 3 presents how we integrate depth information in the visual SLAM process. Experimental results are presented in section 4. Conclusion and further work are presented in section 5.

2 Related Work

There has been a large amount of work on SLAM using RGBD cameras. Given the corresponding depth for each feature the scale ambiguity can be resolved and the initialization of pose and structure is simplified [3, 11]. A most well-known approach using RGB-D sensor is KinectFusion [10]. This work showed that it is possible to obtain an accurate dense 3D model of a medium sized room by merging depth maps, the camera motion is estimated using an iterative closest point algorithm (ICP). *Chen et al.* [13] showed how KinectFusion can be extended for larger scenes through a more memory efficient representation. Other approaches called Dense Visual Odometry DVO [5], estimate the inter-frame motion by minimizing the dense photometric error between successive images using depth data. The approaches cited above can provide an accurate 3D geometry of the scene, but are computationally expensive, and often require modern powerful GPU processor to work in real time. *Henry et al.* [3], presented a near real time feature based approach using RGB-D camera, where features extracted from the RGB images are used for the initial camera pose estimation which is then refined by applying an (ICP) on the depth data (RGB-D mapping combines a sparse feature mapping system with dense RGB-D frame-to-frame alignement), the dense depth data is howver not fused into a full model in real time but through an off-line surfel recontruction step, the sustem has limited dense raconstruction quality but is capable of large scale drift free mapping). Another approach proposed by *Endres et al.* [1] who also uses matched visual features and their depth to initialise

[1] http://vision.in.tum.de/data/datasets/rgbd-dataset

an ICP RANSAC. To compute globally optimal poses for the sensor positions, they use a graph-based optimization routine. Similar to the work of [3]. *Huang et al.* proposed a visual odometry method which is called FOVIS [4], that can run in near-real time on a single CPU. The methods previously cited exploit only points with provided depth data. Unfortunately the used RGB-D camera has a limited range and thus problems with direct or indirect sunlight, depth discontinuities and reflective or highly absorptive surfaces occur. This means that depth measurements are generally not available for the full camera image. To overcome this limitation, some methods [11, 12, 15] propose to exploit both visual information and depth information.

Scherer and al. [11] have recently proposed an extension of PTAM algorithm [6] to take into account depth data. They proposed several simple modifications of the PTAM, the main change concerns the BA. They extended the BA optimization of PTAM by adding depth error in the cost function. Their proposed cost function combines the conventional 2D re-projection error with the depth error. To be combined, these errors of two different dimensions (one in pixels and the second in meters), need to be scaled according to their expected uncertainties which rely on an experimental estimated scaling parameter a. As mentioned by the authors, a need to be changed to a higher value for complex scenario like fast movements. The same authors has recently proposed in a second paper [12], an extension of their previous work. Indeed, one of limitation of PTAM is the runtime complexity of its global BA when the map grows. To overcome this limitation, they proposed in [12] to replace the global BA by a pose graph optimization where the cost function combining depth and visual information is still used to refine the edge of the graph, i.e. the relative pose of a pair of keyframes.

Contribution Our method is an extension of the visual monocular keyframe SLAM. The main differences with the approach of *Scherer et al.* [11, 12] lie in:

- The optimization process: A temporal local BA in our case and a relative graph optimization in the case of [12].
- The cost function of the BA: Totally expressed in pixel in our case and a combination of two different dimension errors in the case of [11, 12].

The performance of this new approach is evaluated on the recently published benchmark dataset[1] and compared to state of the art methods: FOVIS [4], approach cited in [12], [11] revisited. Our approach is illustrated in Figure. 1. It consists in using in real-time and incrementally, conventional computer vision tools (matching, pose computation, triangulation and bundle adjustment). The modifications we made to this visual SLAM [9] are presented in dark.
Features extracted from the current frame are matched with the corresponding reconstructed features from the previous frame to obtain 2D-3D associations. Using these matched features, the current camera pose is computed using the Grunert's pose estimation algorithm [7] in a RANSAC process. While the camera poses are estimated for all the images, adding new features to the map is performed only for keyframes. For

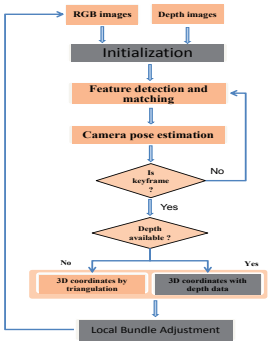

Figure 1: overview of our approach

each detected feature, we can obtain its corresponding 3D point by using the depth provided by the RGBD sensor. Unfortunately for some features, the depth measurements are not available. In this case, we apply a triangulation. The camera poses and the 3D features are then simultaneously refined with a BA. To achieve real time performance we use a local BA that optimizes sequentially only a limited number of camera poses and the observed 3D points. The most important modification is in the integration of depth in the BA, which is described in the next section.

3 Using Depth in BA Process

We propose here a new cost function in BA, which allows to easily combine both depth and visual information without using an additional weight factor as in [11, 12].

Notation We use the basic pinhole camera model to describe the projection of $3D$ points onto a $2D$ image plane. We denote by P, the transformation matrix performing world coordinates into camera frame coordinates.

$$P = \begin{bmatrix} R & t \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{1}$$

Let $Q = (X, Y, Z, 1)^T$ be the homogeneous $3D$ point with respect to world frame and $q = (x, y, z)^T$ the $2D$ homogeneous point with respect to camera's frame such that $q = KPQ$. K describes the intrinsic parameters of the camera:

$$K = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

with focal lengths f_u, f_v and the coordinates of the camera center u_0, v_0. Further, $\pi(q) = (x/z, y/z)^T$ is the perspective projection of q. The $3D$ point with respect to camera's frame is defined by $Q = dK^{-1}q$. Here d is the depth of point Q, provided by the $RGBD$ camera.

Bundle Adjustment Local bundle adjustment optimizes the camera poses and the 3D points by minimizing the $2D$ re-projection error in the N latest keyframes. This error is the difference between the estimated projection of point Q_i through the camera P_j, and it's corresponding observation $q_{i,j}$. To achieve time-efficient performances, the local BA optimizes only the poses of the N_c latest cameras and the N_p 3D points observed in the N latest images with $N \geqslant N_c$. The cost function is defined as :

$$\varepsilon_{\text{slam}}\Big(\{P_j\}_{j=0}^{N_c}, \{Q_i\}_{i=0}^{N_p}\Big) = \sum_{i=0}^{N_p} \sum_{j\in A_i} \rho_s\Big(q_{i,j} - \pi(KP_jQ_i),\ a_s\Big)$$

where A_i is the set of keyframes indices observing Q_i, $\rho_s(.,a_s)$: the Geman-McClure estimator, and a_s is the rejection threshold estimated with the MAD median absolute deviation.

Integrating Depth in Bundle Adjustment In addition to the conventional 2D error mentioned above, we integrate depth measurements as further constraints into BA to improve accuracy and robustness. In contrary to *Scherer et al* [11] who relies the conventional 2D re-projection error with an additional 1D constraint for the depth re-projection error, we propose to combine depth and visual information in a cost function totally expressed in pixel.
We compute the $3D$ position of each $2D$ feature in the camera frame k, from its observation $q_{i,k}$ and its available measured depth $d_{i,k}$. Then we transform it in global coordinate using the Equation. 4.

$$\pi^{-1}(q_{i,k}, d_k) = d_{i,k} K^{-1} q_{i,k} \tag{3}$$

$$Q_{i,k} = P_k^{-1} \pi^{-1}(q_{i,k}, d_{i,k}) \tag{4}$$

We measure the $2D$ projection error of each $3D$ point $Q_{i,k}$ in every frame that observes it, with ($j \in A_i$ and $j \neq k$).
The resulting cost function is given by:

$$\varepsilon_{\text{depth}}\Big(\{P_j\}_{j=0}^{N_c}\Big) = \sum_{i=0}^{N_p} \sum_{j \in A_i} \sum_{k \in A_i}^{k \neq j} \rho_d\Big(q_{i,j} - \pi\Big(K P_j P_k^{-1} \pi^{-1}(q_{i,k}, d_{i,k})\Big),\ a_d\Big).$$

To handle outliers due to mismatched points or erroneous depths, optimisation is performed with a Geman-McClure M-estimator $\rho_d(.,a_d)$, with a_d being the rejection threshold estimated with the MAD median absolute deviation. The resulting cost function, taking into account re-projection errors and depth constraints, is defined by :

$$\varepsilon\Big(\{P_j\}_{j=0}^{N_c}, \{Q_i\}_{i=0}^{N_p}\Big) = \varepsilon_{\text{depth}} + \varepsilon_{\text{slam}} \tag{5}$$

Equation 5 is minimized by the Levenberg-Marquardt algorithm [8]. Note that the proposed BA retains the sparse blocks structure of the matrices involved in the optimization. Like in the classical BA, it is thus possible to implement it effectively, taking into account sparse structure blocks, as described in [14].

4 Evaluation and Experimental Results

We first evaluate our system on a long synthetic sequence ($154m$) representing corridors and offices. We also implement the cost function proposed by [11] in our SLAM framework using the scale factor recommended by the authors ($a = 3.331\ 10^{-3}$). This cost function is illustrated in Figure. 2 .We call this method DSLAM. The results, shown in Figure. 2. and Table. 1 demonstrate that integrating depth data in the visual SLAM reduces significantly its drift. Note that our method DSLAM is different to [11] and [12]. Indeed [11] uses a global BA causing a computational time increasing when the map grows. DSLAM has not this limitation since it uses a local BA. To overcome the limitation of [11] the authors have proposed in [12] a relative graph optimization instead of the global BA. We will show that DSLAM is more accurate in comparison with [12].

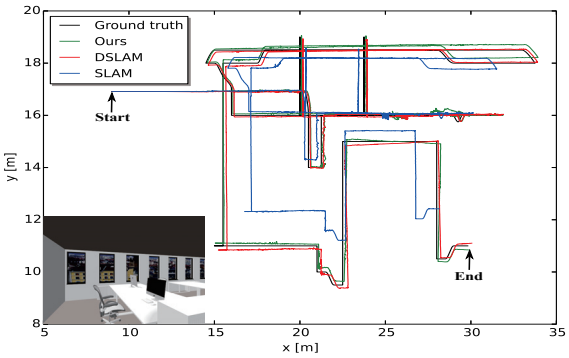

Figure 2: Estimated trajectories on a synthetic sequence.

Error [m] / Methods	RMSE	STD
OURS	0.228	0.187
DSLAM	0.209	0.192
VISUAL-SLAM	1,101	0.575

Table 1: Results on the long synthetic sequence

For a quantitative comparison of the performance of our approach with the other state-of-the-art methods, we evaluate all approaches on the TUM RGB-D benchmark[1]. The dataset contains both RGB images and depth maps provided by the *Microsoft Kinect* ($V1$) with time-synchronized ground truth poses obtained from a high accuracy motion capture system. Our method is compared to the visual SLAM [9], previously mentioned DSLAM , FOVIS [4] that is publicly available [2], and to Scherer's method employing results published in [12]. We evaluate the accuracy of each approach by comparing the estimated trajectory with its ground truth using the absolute translation error (ATE) provided by the tool included with the benchmark dataset.

The results presented in Figure. 3 and Table. 2 demonstrate that integrating depth data in BA of SLAM process improves significantly the accuracy: The ATE is reduced by a factor 10. Our solutions outperforms FOVIS and Scherer's method. However, Ours yields a comparable performance in comparison to DSLAM. The difference between these two approaches is the cost function in the BA. In contrast to DSLAM which requires a specific scaling factor 'a' to combine tow different dimensional errors, our solution uses a cost function entirely expressed in pixel and does not required specific scale. An important point concerns the comparison between DSLAM and Scherer [12] which both use the same cost function in the BA. The difference between these two approaches relies on the optimization process. Indeed, Scherer's approach in [12] is based on a graph optimization where BA is performed only between two keyframes to refine the poses of keyframe pairs corresponding to the edges in the graph. Our method is based on a SLAM where the BA is performed on a temporal sliding window of N keyframes where the N_c latest poses are optimized ($N = 10$ and $N_c = 3$). Therefore, in our case, BA optimizes simultaneously map points and 3 poses compared to 2 in the case of Scherer, using the $2D$ projection error in 10 frames compared to 2 in the case of Scherer. This main difference seems demonstrate that

[2]https://code.google.com/p/fovis

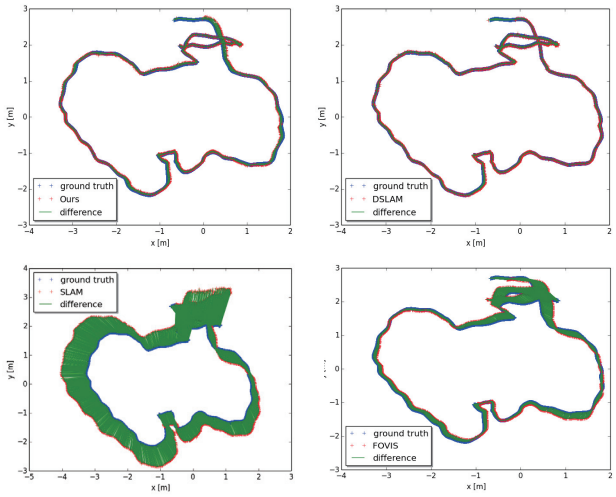

Figure 3: Estimated trajectories on the freiburg3-long-office-household sequence.

BA in N images compared to BA only in the edges of the graph, helps keeping the map consistent and may explain the better results obtained with our framework.

Error [m] / Methods	RMSE	STD
OURS	0.068	0.039
DSLAM	0.069	0.052
Scherer in[12]	0.136	-
FOVIS	0.207	0.117
VISUAL-SLAM	0.652	0.204

Table 2: Comparaison of ATE on the Freiburg3-long-office-household sequence.

On the Freiburg3 sequence, we also evaluate the computational performances of our system on a sequential single-threaded using 1 core from an Intel Xeon W3570 at 3.20 GHz. We measured that the mean run-time required for process each frame is about 25 ms (The inter-frame pose estimation requires an average of 20 ms and BA 38 ms).

5 Conclusion and Further Work

This paper presents a technique to integrate depth measurements into an existing monocular visual SLAM system. The main idea is to use sparse depth information as additional constraints in bundle adjustment. A similar idea has been investigated by [11] who revisited the PTAM algorithm. In this paper, we augment a visual SLAM with temporal local BA and present a new cost function which allows combining depth and visual information without using the additional factor as needed by [11]. The proposed method is evaluated on a public benchmark dataset and compared to the recent state of art methods, as well as the method of [11, 12]. This evaluation shows that using depth information reduces the scale drift and improves the accuracy of a visual SLAM. The proposed cost function allows to keep the sparse structure of the matrices involved in the BA and allows to obtain convincing processing times of $25ms$.

Our future work will concern tests on more sequences and make our SLAM more robust to the texture-less environments. Note also that the proposed solution is not limited to RGBD cameras and can be used with other sensors like a laser scan coupled with a camera.

References

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

[2] N Engelhard, F Endres, Jürgen Hess, J Sturm, and W Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, 2011.

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

[4] 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. ISRR, 2011.

[5] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. ICRA, 2013.

[6] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. ISMAR, 2007.

[7] M. Li and A. I Mourikis. High-precision, consistent ekf-based visual–inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[8] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. 1978.

[9] E. Mouragnon, Maxime Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. CVPR, 2006.

[10] R A. Newcombe, A J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. ISMAR, 2011.

[11] S. A Scherer, D. Dube, and A. Zell. Using depth in visual simultaneous localisation and mapping. ICRA, 2012.

[12] S. A Scherer and A. Zell. Efficient onbard rgbd-slam for autonomous mavs. IROS, 2013.

[13] W. Thomas, K. Michael, F. Maurice, J., Hordur, L John, and M John. Kintinuous: Spatially extended kinectfusion. 2012.

[14] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. ICCVW, 2000.

[15] Ji Zhang, M. Kaess, and S. S. Real-time depth enhanced monocular odometry.