IEICE TRANS. INF. & SYST., VOL.E102–D, NO.1 JANUARY 2019

LETTER

Fast Visual Odometry Based Sparse Geometric Constraint for RGB-D Camera

Ruibin GUO[†a)], ***Student Member***, **Dongxiang ZHOU**[†], **Keju PENG**[†], ***and*** **Yunhui LIU**[††], ***Nonmembers***

SUMMARY Pose estimation is a basic requirement for the autonomous behavior of robots. In this article we present a robust and fast visual odometry method to obtain camera poses by using RGB-D images. We first propose a motion estimation method based on sparse geometric constraint and derive the analytic Jacobian of the geometric cost function to improve the convergence performance, then we use our motion estimation method to replace the tracking thread in ORB-SLAM for improving its runtime performance. Experimental results show that our method is twice faster than ORB-SLAM while keeping the similar accuracy.

key words: *pose estimation, fast visual odometry, geometric cost function, iterative optimization, 3D reconstruction*

1. Introduction

Pose estimation is crucial for robotics mapping and control tasks. The visual odometry can estimate the robot's pose with onboard cameras which provides sufficient pose information. Methods employing monocular cameras have been proposed in [1], [2]. However, the trajectory computed by these methods are not the true scale of the real world. To estimate the absolute true scale factor, more information should be incorporated. This is achieved either by intergrating the monocular camera with Inertial Measurement Units [3] or by using the stereo cameras [4]. Equiped with color and depth information, the RGB-D cameras provide a more convenient way for Simultaneous Localization and Mapping (SLAM) problem. The methods that simultaneously recover camera poses and reconstruct 3D scene mapping by using RGB-D sensor can be divided into two classes: feature-based methods and dense methods. Each class of methods has its own advantages and disadvantages in practical application.

The feature-based methods use feature points to construct constraint and then solve the camera poses. RGB-D SLAM [5] was the first popular open-source system in which the motion estimation was computed by feature matching and ICP. Endrs et al. [6] presented a mapping system based on visual keypoints and provided an open-source implementation to stimulate scientific comparison and progress. Latter, the ORB-SLAM [7] that uses the same ORB features for tracking, mapping and place recognition tasks represents the state-of-the-art feature-based SLAM system. Since the feature-based methods require feature extraction and matching at each frame, it takes up most of the time for computing relative pose. In addition, the feature descriptors that used to matching keypoints are not robust to illumination change.

For dense methods, the pose is estimated with a dense front-end. KinectFusion [8] maintained the single scene model with a global volumetric, this system is limited to small workspaces due to its volumetric representation. Kinectinuous [9] was able to operate in large environments by using a rolling cyclical buffer and using place recognition for loop closing. ElasticFusion [10] is capable of capturing comprehensive dense globally consistent surfel-based maps of room scale environments. Kerl et al. [11] proposed a dense visual SLAM method for RGB-D cameras that minimized both the photometric and the depth error over all pixels. Nevertheless, the number of points processed at each frame is large (typically hundreds of thousands), which make the local optimization computationally infeasible in real-time.

In our work, a geometric cost function and its analytic Jacobian are derived for pose estimation and we combine our motion estimation method with ORB-SLAM for improving its performance. The contributions of this work include:

- We propose a motion estimation method based on sparse geometric constraint, which is robust to illumination changes and eliminates the feature-matching process to reduce the runtime of motion estimation.
- We derive the analytic Jacobian of the geometric cost function to improve the convergence performance.
- We improve the performance of ORB-SLAM by excluding the keypoints near the objects' edges and using our fast motion estimation method to estimate poses.

2. Proposed Method

In this section, we first derive a warping function and construct the geometric error function by using probabilistic theory; then the analytic Jacobian of the geometric cost function is derived for faster convergence; at last, we combine our motion estimation method with ORB-SLAM to improve its runtime performance and excluding the keypoints

Manuscript received June 6, 2018.
Manuscript revised September 13, 2018.
Manuscript publicized October 9, 2018.

[†]The authors are with College of Electronic Science, National University of Defense Technology, Changsha, 410073, China.
[††]The author is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China.
a) E-mail: guoruibin64@126.com

DOI: 10.1587/transinf.2018EDL8119

near the objects' edges for more accurate pose estimation.

2.1 Warping Function for Depth Map

For two consecutive depth images $\mathbf{d_{k-1}}$ and $\mathbf{d_k}$ shown in Fig. 1, an image point $\mathbf{u_1} = (u_1, v_1)^T$ at depth $\mathbf{d_{k-1}}$ is transformed to the current depth plane $\mathbf{d_k}$ by the warping function $W(\cdot)$ and its pixel coordinate $\mathbf{u_1'}$ is predicted.

Firstly, we reconstruct the 3D point $\mathbf{p_1} = (x, y, z)^T$ corresponding to the pixel $\mathbf{u_1} = (u_1, v_1)^T$ by using the inverse projection function:

$$\mathbf{p_1} = \boldsymbol{\pi}^{-1}(\mathbf{u_1}, d_{k-1}(\mathbf{u_1})) = d_{k-1}(\mathbf{u_1})\left(\frac{u_1 - c_x}{f_x}, \frac{v_1 - c_y}{f_y}, 1\right)^T \tag{1}$$

where f_x, f_y are the focal lengths on x axis and y axis, and $(c_x, c_y)^T$ is the camera centra coordinate.

Then, $\mathbf{p_1}$ is transformed to $\mathbf{p_1'} = \mathbf{R}_{k,k-1} \cdot \mathbf{p_1} + \mathbf{t}_{k,k-1}$ in $\mathbf{d_k}$ coordinate by using rigid-body transformation matrix $\mathbf{T}_{k,k-1} = \begin{bmatrix} \mathbf{R}_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$, where $\mathbf{R}_{k,k-1}$ is a 3×3 rotation matrix and $\mathbf{t}_{k,k-1} = (t_x, t_y, t_z)^T$ is a 3×1 vector represents a translation from frame $\mathbf{d_{k-1}}$ to frame $\mathbf{d_k}$.

Lastly, $\mathbf{p_1'}$ is projected to depth map $\mathbf{d_k}$ by the projection function $\boldsymbol{\pi}(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}^2$, and the warping function is:

$$\begin{aligned} \mathbf{u_1'} &= W(\mathbf{T}_{k,k-1}, \mathbf{u_1}, d_{k-1}(\mathbf{u_1})) \\ &= \boldsymbol{\pi}(\mathbf{R}_{k,k-1} \cdot \boldsymbol{\pi}^{-1}(\mathbf{u_1}, d_{k-1}(\mathbf{u_1})) + \mathbf{t}_{k,k-1}) \end{aligned} \tag{2}$$

where $\boldsymbol{\pi}(\mathbf{p}) = (f_x \frac{x}{z} + c_x, f_y \frac{y}{z} + c_y)^T$ and $\mathbf{p} = (x, y, z)^T$.

2.2 Constructing Geometric Cost Function

For a pixel $\mathbf{u_i}$ at depth image $\mathbf{d_{k-1}}$, the residual error as the geometric difference between $\mathbf{d_{k-1}}$ and $\mathbf{d_k}$ is:

$$\begin{aligned} r(\mathbf{u_i}) = d_k(W(\mathbf{T}_{k,k-1}, \mathbf{u_i}, d_{k-1}(\mathbf{u_i}))) \\ - \lfloor \mathbf{R}_{k,k-1} \cdot \boldsymbol{\pi}^{-1}(\mathbf{u_i}, d_{k-1}(\mathbf{u_i})) + \mathbf{t}_{k,k-1} \rfloor_3 \end{aligned} \tag{3}$$

where $\lfloor \cdot \rfloor$ represents the Z-component of a 3D point.

The distribution of the residual error can be described by a conditional probabilistic $p(r_i|\xi)$, where $\xi = (\omega, t)^T$ is the corresponding twist coordinates of $\mathbf{T}_{k,k-1}$, ω is the angular velocity and t is the linear velocity. The twist coordinate ξ is mapped to $SE(3)$ by exponential map [12]:

$$\mathbf{T}_{k,k-1}(\xi) = \exp(\hat{\xi}) \tag{4}$$

where $\hat{\xi} = \begin{pmatrix} [\omega]_\times & t^T \\ 0 & 0 \end{pmatrix}$, and $[\omega]_\times$ is the skew symmetric matrix of ω^T.

For n pixels $\mathbf{u_i}$ with $i = 1, 2, \ldots, n$, the likelihood of the whole residual becomes $p(r|\xi) = \prod_i p(r_i|\xi)$. Using Bayes' rules, the posterior likelihood of a camera motion ξ is:

$$p(\xi|r) = \frac{p(r|\xi)p(\xi)}{p(r)} \tag{5}$$

The prior $p(\xi)$ models the potential knowledge of the states before we do the measurements. If we know nothing, $p(\xi)$ is a uniform distribution with constant value. By maximizing the posterior probability, ξ can be estimated:

$$\begin{aligned} \xi_{MAP} &= \arg\max_\xi p(\xi|r) = \arg\max_\xi \prod_i p(r_i|\xi)p(\xi) \\ &= \arg\min_\xi -\sum_i \log p(r_i|\xi) - \log p(\xi) \\ &= \arg\min_\xi -\sum_i \log p(r_i|\xi) \end{aligned} \tag{6}$$

The minimum is obtained by setting the derivative of the logarithmic likelihood function to zero:

$$-\sum_i \frac{\partial \log p(r_i|\xi)}{\partial \xi} = -\sum_i \frac{\partial \log p(r_i|\xi)}{\partial r_i} \frac{\partial r_i}{\partial \xi} = 0 \tag{7}$$

By defining $w(r_i) = \partial \log p(r_i|\xi)/\partial r_i \cdot 1/r_i$, we obtain $\sum_i \partial r_i/\partial \xi \cdot w(r_i) \cdot r_i = 0$, and the optimization problem in (6) is equivalent to the weighted least squares problem:

$$\xi_{MAP} = \arg\min_\xi -\frac{1}{2}\sum_i w(r_i) \cdot r_i^2 \tag{8}$$

As the depth noise n_d of all pixels is independent and follows Gaussian distribution [13], i.e. $n_d \in N(0, \sigma_d^2)$, and $p(r_i) \propto \exp(-r_i^2/\sigma_d^2)$, then $w(r_i)$ is a constant value, and we get the geometric cost function:

$$\xi_{MAP} = \arg\min_\xi \sum_{\mathbf{u_i}} \|r(\xi, \mathbf{u_i})\|^2 \tag{9}$$

2.3 Analytic Jacobian Solution for Iterative Optimization

To minimize the geometric cost function (9), the Gauss-Newton algorithm can be used, whose main idea is to approximate the error function by its first order Taylor expansion that determined by Jacobian matrix around the initial guess of variable $\mathbf{T}_{k,k-1}(\xi)$. However, the Jacobian matrix in g2o library [14] is computed numerically by a stable small constant, the convergence of iterative optimization is slow. Therefore, it is important to derive the analytic Jacobian matrix for faster convergence:

$$\frac{\partial r(\xi, \mathbf{u_i})}{\partial \xi} = \begin{pmatrix} g_y & -g_x & 0 & 0 & 0 & 1 \end{pmatrix}$$

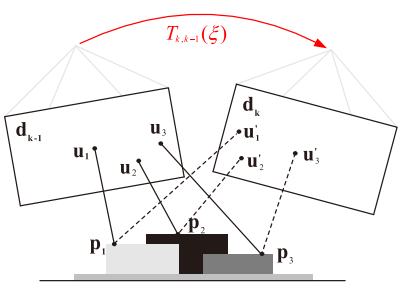


Fig. 1 Estimation for the relative pose through minimizing the geometric difference between image pixels corresponding to the same 3D points.

$$+\begin{pmatrix} \frac{\partial d_k(\mathbf{u}_i')}{\partial u_i'} \cdot f_x \\ \frac{\partial d_k(\mathbf{u}_i')}{\partial v_i'} \cdot f_y \end{pmatrix}^T \begin{pmatrix} -\frac{g_x g_y}{g_z^2} & 1+\frac{g_x^2}{g_z^2} & -\frac{g_y}{g_z} & \frac{1}{g_z} & 0 & -\frac{g_x}{g_z^2} \\ -1-\frac{g_y^2}{g_z^2} & \frac{g_x g_y}{g_z^2} & \frac{g_x}{g_z} & 0 & \frac{1}{g_z} & -\frac{g_y}{g_z^2} \end{pmatrix}$$

where $\mathbf{g} = (g_x, g_y, g_z)^T = \mathbf{R}_{k,k-1} \cdot \boldsymbol{\pi}^{-1}(\mathbf{u_i}, d_{k-1}(\mathbf{u_i})) + \mathbf{t}_{k,k-1}$ and $\mathbf{u}_i' = (u_i', v_i')^T = \boldsymbol{\pi}(\mathbf{g})$.

2.4 Improvement for ORB-SLAM

ORB-SLAM system uses ORB features for tracking, mapping and place recognition. We use our motion estimation method to replace the tracking thread in ORB-SLAM for improving its performance. The flow chart of our tracking method for consecutive frames is shown in Fig. 2.

Runtime Performance: Feature extraction and matching are the necessary steps before solving the camera poses in ORB-SLAM. However, the point correspondences and the relative camera motion can be obtained simultaneously by minimizing the geometry cost function in our method, we need not to extract the ORB features for each frame, unless it is determined to be a keyframe.

Accuracy Performance: Since the gradients near edges are larger than those in smooth regions, which result in larger geometric error near the objects' edges affected by depth noise. We exclude the feature points near the geometric boundary and use the optimized point set to estimate camera poses:

$$\xi_{MAP} = \arg\min_{\xi} \sum_{i \in \tilde{\mathcal{R}}_k} \|r(\xi, \mathbf{u_i})\|^2 \tag{10}$$

where $\tilde{\mathcal{R}}_k = \{\mathbf{u} | \mathbf{u} \in \mathcal{R}_{k-1} \wedge \mathbf{u} \notin \mathcal{B}_{k-1} \wedge \boldsymbol{\pi}(\mathbf{R}_{k,k-1} \cdot \boldsymbol{\pi}^{-1}(\mathbf{u}, d_{k-1}(\mathbf{u})) + \mathbf{t}_{k,k-1}) \in \Omega_k\}$, $\mathcal{R}_{k-1}$ is the set of feature points in frame F_{k-1}, Ω_k is the image domain of depth $\mathbf{d}_k$, and $\mathcal{B}_{k-1}$ is the set of keypoints near objects' boundary.

Robustness to Illumination Changes: The matching points and estimated poses in our method are computed directly by using geometric values in the depth images rather than color images, so it is robust to illumination changes. Figure 3 shows the comparison of matching point pairs obtained by feature matching method and our method on illumination change images, there are 48 and 788 point correspondences, respectively. The absolute translational error (ATE) of relative pose for these two images obtained by feature-based method and our method is 0.0333(m) and 0.0201(m), respectively. The lower the ATE value is, the higher the accuracy of the method is. Therefore, our method is more robust and with higher accuracy than the feature-based method on illumination change images.

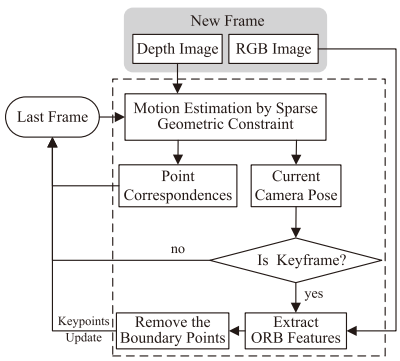


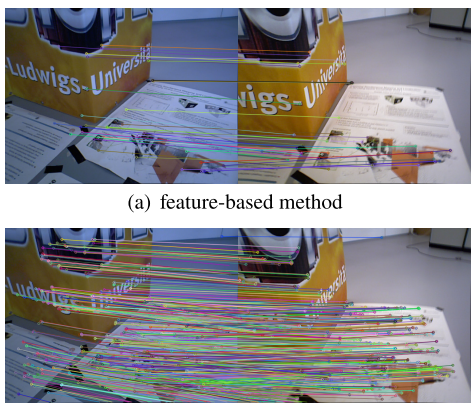
Fig. 2 Overview of our tracking method.

3. Experiments and Evaluation

The TUM RGB-D dataset [15], which contains indoor sequences from RGB-D sensors grouping with several categories, is applied to evaluate SLAM/odometry methods and the ORB-SLAM system is taken as comparision. We have run both our approach and ORB-SLAM in an Intel Core i7 desktop computer with 16GB RAM, ubuntu 16.04 platform. For each dataset, we run 5 times and the average results of the accuracy and runtime are computed.

3.1 Accuracy

Table 1 shows the accuracy of our visual odometry method with/without excluding points near objects' boarder and ORB-SLAM, the root mean square error (RMSE) of the absolute translational error for keyframes' poses is used as the performance metrics. The result shows that proposed method with excluding boarder's points performs better than no boarder suppression, and it works with almost the same precision compared with ORB-SLAM. Figure 4 shows the point clouds obtained by back-projecting the sensor depth maps from the computed keyframe poses (represented as blue box) in three sequences: fr3/office, fr3/st and fr2/xyz. These pointcloud reconstructions show the accuracy of estimation poses that obtained by our method intuitively.



(a) feature-based method

(b) proposed method

Fig. 3 Point pairs' matching result on illumination change images.

Table 1 Comparison of translation RMSE (m) on TUM RGB-D dataset.

Dataset	Our method	No boarder suppression	ORB-SLAM
fr2/xyz	0.00452	0.006814	**0.00384**
fr2/rpz	**0.00324**	0.010492	0.00528
fr3/office	**0.0128**	0.034242	0.0132
fr3/st	**0.0115**	0.027164	0.0148

(a) fr3/office (b) fr3/st (c) fr2/xyz

Fig. 4 Dense pointcloud reconstructions from estimated keyframe poses and RGB-D information.

Table 2 Comparison of convergence performance.

Evaluation	Analytic method	Numerical Method
ATE (m)	**0.0043**	0.0047
runtime (ms)	**11.70**	45.08

Table 3 Runtime evaluation for pose evalution time (ms).

Dataset	Our method	Numerical method	ORB-SLAM
fr2/xyz	**20.7**	69.5	31.2
fr2/rpz	**19.3**	64.2	27.5
fr3/office	**25.8**	74.2	40.4
fr3/st	**18.3**	67.9	29.3

3.2 Runtime Evaluation

To depict the effectiveness of our proposed analytic Jacobian, experiments have been performed on two consecutive frames with 1008 point correspondences to estimate relative pose. We have computed the runtime of iterative process and the ATE of relative pose, as shown in Table 2. It demonstrates that proposed analytic jacobian is 4 times faster than the numerical jacobian used in g2o.

Table 3 shows the average runtime required for frames by using our visual odometry method with analytic/numerical jacobian and ORB-SLAM, this runtime consists of 2 processes: (1) extracts feature points and pose estimation for consecutive frames in front-end, (2) local map optimization and loop detection in back-end. The processing speed of the proposed method is more than 50 frames per second. While the corresponding processing speed for ORB-SLAM is 25 frames per second. The main reason for the significant speed-up is that we use our proposed analytic jacobian to estimate motion and our method does not need feature matching procedure. In addition, our method minimizes the geometric cost function by using sparse reliable ORB keypoints, which are extracted when a new keyframe inserted instead of being extracted in each new frame.

4. Conclusion

In this paper, we proposed a sparse geometric-based motion estimation method without feature-matching procedure, and we derived the analytic jacobian to minimize our geometric cost function. Our method could estimate motion directly from geometric values without intensity-consistent assumption, so it is robust to illumination changes. The computational efficiency of ORB-SLAM is significantly improved by using our method to track sparse points in front-end, and our method keeps the similar accuracy by using optimized point set that excludes the boarders' points to estimate poses.

References

[1] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," International Symposium on Mixed and Augmented Reality, pp.1–10, 2007.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," IEEE International Conference on Robotics and Automation, pp.15–22, 2014.

[3] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," International Journal of Robotics Research, vol.34, no.3, pp.314–334, 2015.

[4] T. Pire, T. Fischer, J. Civera, P. De Cristóforis, and J.J. Berlles,"Stereo parallel tracking and mapping for robot localization," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1373–1378, 2014.

[5] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox,"RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," Experimental Robotics, Springer Tracts in Advanced Robotics, vol.79, pp.477–491, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[6] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," IEEE Trans. Robotics, vol.30, no.1, pp.177–187, 2014.

[7] R. Mur-Artal and J.D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," IEEE Trans. Robotics, vol.33, no.5, pp.1255–1262, 2017.

[8] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," IEEE International Symposium on Mixed and Augmented Reality, pp.127–136, 2011.

[9] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J.J. Leonardet, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," International Journal of Robotics Research, vol.34, no.4-5, pp.598–626, 2015.

[10] T. Whelan, S. Leutenegger, R.F. Salas-Moreno, B. Glocker, and A.J. Davison, "ElasticFusion: Dense SLAM without a pose graph," Robotics: Science and Systems, 2015.

[11] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," IEEE/RSJ International Conference on Intelligent Robots

and Systems, pp.2100–2106, 2013.

[12] D.E. Holmgren, "An invitation to 3-d vision: From images to geometric models," Photogrammetric Record, vol.19, no.108, pp.415–416, 2004.

[13] C.V. Nguyen, S. Izadi, and D. Lovell, "Modeling Kinect sensor noise for improved 3D reconstruction and tracking," International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, pp.524–530, 2012.

[14] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," IEEE International Conference on Robotics and Automation, vol.7, pp.3607–3613, 2011.

[15] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," Intelligent Robots and Systems, pp.573–580, 2012.