Adaptive keyframe selection for light-weighted SLAM using RGB-D cameras

Qiaomeng Qin[1,2], Hongying Meng[1], and Yanxi Yang[2]

[1] Brunel University London, London UB8 3PH, UK,
[2] Xi'an University of Technology, Xi'an 710048, China

Abstract. Simultaneous Localization and Mapping (SLAM) is a critical system for Unmanned Aerial Vehicles (UAV), Autonomous Navigation (AN), Augmented Reality (AR) and Virtual Reality (VR). However, due to volume and computation limitations, the current SLAM systems on mobile devices are not robust enough for practical applications. In this paper, a new SLAM system is proposed, which can be applied to AR or VR. With the RGB-D input data, this SLAM system can produce 3D scene reconstruction and camera pose estimation in almost real time. The parameters of the system are optimized for use on mobile devices. In addition, an adaptive keyframe selection method is proposed to balance trade-off between accuracy and speed of the system. Experimental results demonstrated its superior performance in comparison with the state-of-the-art methods in the same public benchmark dataset.

Keywords: SLAM, Keyframe selection, Augmented Reality, Virtual Reality

1 Introduction

Simultaneous Localization and Mapping (SLAM) is an umbrella name for a highly active research area in the field of computer vision and robotics for the goal of 3D scene reconstruction and camera pose estimation from imaging sensors [1]. SLAM finds applications in all scenarios in which a prior map is not available and needs to be built [2]. The SLAM community has made witnessing a steady transition of this technology to industry. A thorough historical review of the first 20 years of the SLAM problem is given by Durrant–Whyte and Bailey in this survey [3].

Recently, more and more mobile devices are equipped with Depth cameras, such as Apple iPad Pro 2020 and DJI Mavic2 Pro. When it comes to the SLAM systems applied to mobile devices, it is an important problem that how to balance the accuracy and speed of real-time SLAM systems. Therefore, in this paper, one of the state-of-the-art SLAM systems, BAD-SLAM, is optimized for mobile devices as an instance. The optimization procedure is also suitable for other SLAM systems.

2 Related work

Bundle Adjustment (BA) is a robust and widely used method in SLAM,, and it is efficient in real-time SLAM in a state-of-the-art method BAD-SLAM [2]. Among all the current SLAM algorithms with RGB-D input for mobile devices, BAD-SLAM [2] has the best result. Its front end uses the ICP algorithm and the back end uses the BA optimization algorithm. However, in BAD-SLAM, many parameters are not optimized for mobile devices. Therefore, in this paper, improvements are made on the basis of BAD-SLAM. It will balance accuracy and speed.

There is a crucial step in the front end of SLAM, which is the keyframe selection method. A simple and popular keyframe selection method in many SLAM systems ([2], [4], [5], [6]) is to select keyframes after a constant number of interval frames. Researchers choose an optimal number of frames according to different scenarios and equipment. The advantage of this method is that the computation is small, but the disadvantage is also very obvious, that is, poor adaptability. When the moving speed is slow, it is easy to generate too close keyframes, which wastes computing resources. When the movement speed is too fast, large errors will occur. Another popular method is to choose the keyframe once the pose transformation is large than a constant distance from the last keyframe [8] [11]. The advantage of this method is that it is flexible, suitable for fast and slow motion. but the disadvantage is the larger calculation. When the moving speed is slow, it is easy to finish the calculation before the next frame. When the movement speed is too fast, the system will need more resources. The keyframe selection method of BAD-SLAM uses an interval of a fixed number of frames, which is far from efficient for mobile devices.

3 Methodology

3.1 System overview

The basic outline of the proposed SLAM algorithm is shown in Figure 1. It is improved from an existing state-of-the-art SLAM system, BAD-SLAM [2]. There are three main improvements as shown in the green parts of Figure 1, bilateral filter, keyframe selection method and Robust kernel function. Firstly, While a new frame comes, a common bilateral filter will smoothen the depth map, some noise and large depth measurements will be removed. Secondly, the pose of the current frame will be estimated via the ICP method. Then, the keyframe selection method will be executed to determine whether the frame is selected as a keyframe. Thirdly, if the current frame is selected as a new keyframe, the Loop Closure Detection will be conducted. Finally, all the information will be given to the Back End to conduct the Direct BA method.

3.2 Optimization for parameters

The Bilateral filter can smoothen the depth map. It constructs a convolution kernel based on the adjacent pixels for every pixel. The convolution kernel of the

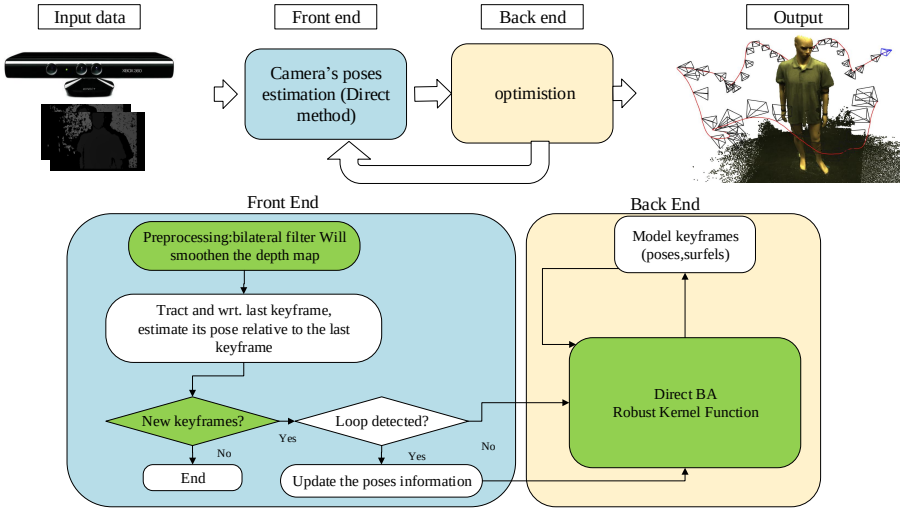

Fig. 1. Top: the basic structure of a SLAM system. Bottom: the outline of the proposed SLAM algorithm

(r, c)th pixel $Kernel(h, w)$ is given by

$$\begin{aligned} Kernel(h,w) &= Kernel_{closeness}(h,w) \cdot Kernel_{similarity}(h,w) \\ &= \exp\left(-\frac{\left(h-\frac{winH-1}{2}\right)^2+\left(w-\frac{winW-1}{2}\right)^2}{2\sigma_1^2}\right) \\ &\cdot \exp\left(-\frac{\|\boldsymbol{I}(r,c)-\boldsymbol{I}\left(r+\left(h-\frac{winH-1}{2}\right),c+\left(w-\frac{winW-1}{2}\right)\right)\|^2}{2\sigma_2^2}\right) \end{aligned} \tag{1}$$

where $Kernel_{closeness}(h, w)$ is a gauss matrix relative to the distance between (h, w) and kernel center, $Kernel_{similarity}(h, w)$ is a gauss matrix relative to the similarity between (h, w) and (r, c), $0 < h < winH$, $0 < w < winW$, $winH$ represents the hight of the kernel, $winW$ represents the width of the kernel, σ_1 and σ_2 represent the standard deviation of two gauss matrices.

In the BA optimization, the errors need to be minimized as the objective function. However, there is a serious problem: the optimization algorithm cannot distinguish outliers. These outliers can be generated for reasons such as moving objects or shadow changes. optimization method will treat all data as edges to be optimized. In order to let the optimization algorithm distinguish the outliers, the Tukey's biweight robust loss function is applied to errors before the optimization.

In order to let the optimization algorithm distinguish the outliers, the Tukey's biweight robust loss function is applied to errors before the optimization. Its

equation is given by Equation 2. Now, the loss function is given by Equation 3.

$$Tukey(e) = \begin{cases} e\left(1 - \frac{e^2}{c^2}\right)^2 & \text{for } |e| < c \\ 0 & \text{for } |e| > c \end{cases} \tag{2}$$

$$\|f(\boldsymbol{x} + \Delta\boldsymbol{x})\|^2 = \left\|Tukey(\boldsymbol{e}) + \boldsymbol{F}\Delta\boldsymbol{x}_c + \boldsymbol{E}\Delta\boldsymbol{x}_p\right\|^2 \tag{3}$$

Where c represents the parameter for Tukey's biweight robust loss function. Several experiments will be conducted on different values of c to determine the most suitable value for the indoor environment. The Jacobian matrices $\boldsymbol{E}$ and $\boldsymbol{F}$ here are the derivatives of the overall objective function to the overall variables. In our system, the optimized values σ_1, σ_2 and c were determined empirically according to experiments and results.

3.3 Adaptive Keyframe Selection

The keyframe selection method aims to determine the suitable density of the keyframes. In order to find the best keyframe selection method for real-time SLAM systems in the indoor environment, three different methods will be tested. A simple and popular keyframe selection method is to select keyframes after a constant number of interval frames. In mathematics, the Euclidean distance or Euclidean metric is the ordinary straight-line distance between two points in Euclidean space. The Euclidean distance can be applied to determine the distance between two frames in this keyframe selection method, once the Euclidean distance between the current frame and last keyframe is large than the maximum distance D_{Euc} or K_{max} frames have passed from the last keyframe, set the current frame as a new keyframe. Keyframe selection method with Euclidean distance will be compared with the proposed method.

Proposed Mixed Distance There are two different parts in the camera poses' vector $\boldsymbol{\xi}$, translation and rotation movement. And the impacts of these two parts are different, SLAM systems are more sensitive to rotation movement. Therefore, in this project, two coefficients for translation and rotation are proposed in a new keyframe selection method.

A new mixed Euclidean distance is proposed which combines translation and rotation movement. This mixed Euclidean distance is given by equation

$$\begin{aligned} d_{\text{mixed}} &= \mu_t \left\|\boldsymbol{t}_1,\ \boldsymbol{t}_2\right\| + \mu_q \left\|\boldsymbol{q}_1\ \boldsymbol{q}_2\right\| \\ &= \mu_t \sqrt{\sum_{i=1}^{3} \left(\boldsymbol{\xi}_{1_i} - \boldsymbol{\xi}_{2_i}\right)^2} + \mu_q \sqrt{\sum_{i=4}^{7} \left(\boldsymbol{\xi}_{1_i} - \boldsymbol{\xi}_{2_i}\right)^2} \end{aligned} \tag{4}$$

where μ_t and μ_q represent the coefficients of translation and rotation movement, they satisfy the relationship shown in Equation 5. Therefore, once the value of one of μ_t or μ_q is determined, another one is determined as well. The value of μ_q from 0.1 to 0.9 will be tested in several experiments, and the best values of these coefficients will be determined. $\boldsymbol{\xi}_k$ represents the 7 DOF camera pose of the kth keyframe.

$$\mu_t + \mu_q = 1 \tag{5}$$

4 Experimental results

4.1 Dataset and parameters

An experimental evaluation is provided to validate the contributions of the proposed method in terms of tracking accuracy and computation cost, by means of a comparison against the state of the art on a public benchmark dataset, ETH3D dataset, which can be found in the ETH3D website [12], which is one of the latest datasets. ETH3D SLAM datasets have 61 training datasets and 35 test datasets. Only the ground truth of training datasets is available on the website. Therefore, the training datasets will be adopted to conduct the experiments.

A laptop with an Intel Core i5 4200H CPU and a Nvidia GTX 850M graphic card is used in this work. The whole project is implemented on an Ubuntu 18.04 operating system.

For visual SLAM systems, the global consistency of the estimated trajectory is an important quantity. The global consistency can be evaluated by comparing the absolute distances between the estimated and the ground truth trajectory [13]. Therefore, the RMSE of ATE will be used as the final indicator to judge the accuracy of the SLAM system. For each dataset, the ground truth given by ETH3D will be compared to the results from experiments.

To get the best results for this real-time SLAM system, all undefined parameters mentioned in section 3 have been tested in a proper range, and corresponding ATE RMSE results will be compared with the results from BAD-SLAM [2], all their original results can be found in the paper and website [10]. We will use "Original" to represent the results from BAD-SLAM in the comparison.

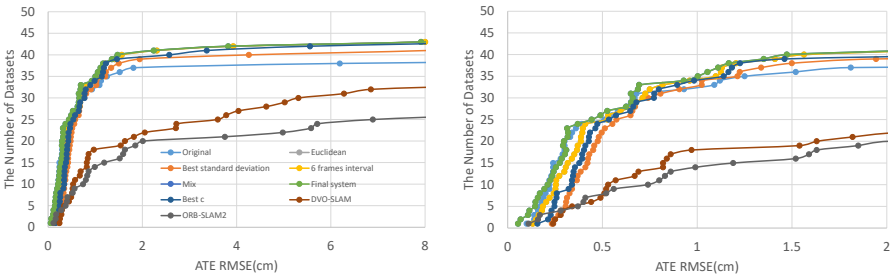

Fig. 2. The comparison between its performance and other algorithms. For a given threshold on the ATE RMSE (x-axis), the graph shows the number of datasets for which the evaluated variant has a smaller ATE RMSE. ATE RMSE stands for the Root Mean Square Error of Absolute Trajectory Error, which should be as small as possible. Left: the number of datasets for which the ATE RMSE is smaller than 8. Right: the number of datasets for which the ATE RMSE is smaller than 2.

Several experiments are conducted with different $\sigma_{1,2}$ in Equation 1, c in Equation 2 and μ_q in Equation 4. The best result is 5.89 cm which is given by

$\sigma_1 = 14$ and $\sigma_2 = 0.4$ 3.72 cm which is given by $c = 7$ and 1.26 cm which is given by $\mu_q = 0.7$.

4.2 Performance

The final system will adopt all the best parameters and the keyframe selection method with mixed distance, and several experiments for the final system have been conducted. The comparison between the final system, each system with an optimized parameter and the original system is shown in Figure 2. In order to display the difference between two keyframe systems clearly, the numbers of datasets with the ATE RMSE less than 2 for both methods are also shown in Figure 2. To demonstrate the overall performance of the system with different improvements, the corresponding distributions of the ATE RMSE that mentioned above are shown in Table 1.

To show the improvement of the final system intuitively, the comparisons of the estimated trajectory and the ground truth of dataset $cables - 3$ from final and original systems are shown in Figure 3.

Table 1. The distribution of the ATE RMSE for the system with best σ. AKS represents adaptive keyframe selection, SD represents standard deviation

Method	AKS	Mean(cm)	SD(cm)
DVO-SLAM [9]	Yes	8.62	19.77
ORB-SLAM2 [16]	Yes	9.62	20.33
BAD-SLAM [2]	No	7.91	19.25
With the best σ	No	5.89	12.44
6 frames interval	No	3.63	10.22
Euclidean distance	Yes	1.31	2.65
With the best c	No	3.72	11.72
Mixed distance	Yes	1.26	2.65
Mixed distance with best parameters	Yes	**1.01**	**2.55**

4.3 Resources utilization

For the real-time SLAM system, the resource consumption of the system is critical as well. In order to evaluate the resource consumption of the proposed system, the time consumption and VRAM consumption of each system with different keyframe selection methods will be shown in this section. The average time consumption and VRAM consumption of 61 datasets from each system are shown in Table 2.

5 Conclusion

This paper improves the current state-of-the-art RGB-D SLAM system for mobile devices with optimized parameters and better keyframe selection method.

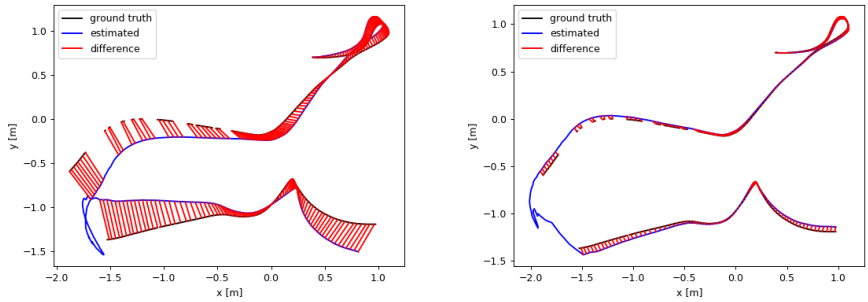

Fig. 3. The comparisons of the estimated trajectory and the ground truth of dataset *cables* − 3 from the original system. The red line represents the difference between the estimated trajectory and the ground truth. Therefore, the shorter the red line, the more precise the estimated trajectory. Left: the estimated trajectory of original system. Right: the estimated trajectory of the final system.

Table 2. The time and VRAM consumption for each system.

Method	Final System	Mixed	Euclidean	6 frames Interval	BAD-SLAM
Average Time (seconds per 100 frames)	11.6	11.3	11.1	22.3	9.5
VRAM (Megabyte per 100 frames)	50.33	49.23	55.83	77.83	42.9

The mean ATE RMSE has been improved from 7.91cm to 1.01cm. In addition, the time consumption of the proposed system is 23% larger than the original system, and the VRAM consumption of the proposed system is 19% larger than original system, which is still acceptable for mobile devices. In contrast to the original system with 6 frames interval keyframe selection method, the proposed system has a better performance but 46% less time consumption. Overall, the problem to balance the accuracy and the speed for mobile devices, is solved by the proposed keyframe selection method with mixed distance and better parameters.

6 Acknowledgement

This research work is partially funded by the National Key R&D Program Network Collaborative Manufacturing and Smart Factory Key Special Project (No. 2018YFB1703000) and Shaanxi Province Modern Equipment Green Manufacturing Collaborative Innovation Center Project (No. 304-210891702).

References

1. C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," in IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309-1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.
2. T. Schöps, T. Sattler and M. Pollefeys, "BAD SLAM: Bundle Adjusted Direct RGB-D SLAM," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 134-144, doi: 10.1109/CVPR.2019.00022.
3. T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," in IEEE Robotics & Automation Magazine, vol. 13, no. 3, pp. 108-117, Sept. 2006, doi: 10.1109/MRA.2006.1678144.
4. C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp. 15-22, doi: 10.1109/ICRA.2014.6906584.
5. D. Caruso, J. Engel and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 141-148, doi: 10.1109/IROS.2015.7353366.
6. J. Engel, J. Stückler and D. Cremers, "Large-scale direct SLAM with stereo cameras," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 1935-1942, doi: 10.1109/IROS.2015.7353631.
7. K. Tateno, F. Tombari, I. Laina and N. Navab, "CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6565-6574, doi: 10.1109/CVPR.2017.695.
8. R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.
9. C. Kerl, J. Sturm and D. Cremers, "Dense visual SLAM for RGB-D cameras," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013, pp. 2100-2106, doi: 10.1109/IROS.2013.6696650.
10. ETH3D SLAM Benchmark. http://www.ncbi.nlm.nih.gov
11. R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2320-2327, doi: 10.1109/ICCV.2011.6126513.
12. ETH3D SLAM Datasets. https://www.eth3d.net/slam_datasets
13. J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012, pp. 573-580, doi: 10.1109/IROS.2012.6385773.
14. Van der Heijden, F. & Duin, Robert & Ridder, D. & Tax, David. (2004). Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB. 10.1002/0470090154.
15. Clark, Ronald & Wen, Hongkai & Markham, Andrew & Trigoni, Niki. (2017). VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem.
16. R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.