Original Loop-closure Detection Algorithm for Monocular vSLAM

Andrey Bokovoy\textsuperscript{1,3} and Konstantin Yakovlev\textsuperscript{2,3}

\textsuperscript{1} Peoples’ Friendship University of Russia (RUDN University), Moscow, Russia
1042160097@rudn.university
\textsuperscript{2} Higher School of Economics, Moscow, Russia
kyakovlev@hse.ru
\textsuperscript{3} Institute for Systems Analysis of Federal Research Centre "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia
\{bokovoy,yakovlev\}@isa.ru

Abstract. Vision-based simultaneous localization and mapping (vSLAM) is a well-established problem in mobile robotics and monocular vSLAM is one of the most challenging variations of that problem nowadays. In this work we study one of the core post-processing optimization mechanisms in vSLAM, e.g. loop-closure detection. We analyze the existing methods and propose original algorithm for loop-closure detection, which is suitable for dense, semi-dense and feature-based vSLAM methods. We evaluate the algorithm experimentally and show that it contribute to more accurate mapping while speeding up the monocular vSLAM pipeline to the extent the latter can be used in real-time for controlling small multi-rotor vehicle (drone).

Keywords: loop-closure, vision-based localization and mapping, unmanned aerial vehicle, SLAM, vSLAM.

1 Introduction

Vision-based simultaneous localization and mapping (vSLAM) is one of the most challenging problems in computer vision and robotics. SLAM methods, that rely only on the information gained from minimum set of miniature passive sensors (monocular or stereo camera, inertial measurement unit), lie at the core of navigation capabilities of various mobile robots. Especially, they are of great value for compact unmanned aerial vehicles (which can not be equipped by the heavy, powerful sensors by default).

Recently a notable progress in the field of UAV vSLAM methods was made, see \cite{1,2}, for example. However, there’s still a large set of real-world problems and scenarios that can not be successfully tackled by the existing vision-based SLAM algorithms. The main reasons for that are the following.

First is the image processing time. Modern embedded computers that can be installed on compact UAVs are not that powerful to execute typical vSLAM pipelines in real time. Using external sources for remote computations is not...
always the solution since it lowers the mobility (robotic system is forced to continuously exchange huge amount of information with remote control station, using wire or wireless channel) and prevents robotic system from being fully autonomous.

Second is poor image quality [3]. Small cameras typically mounted on compact UAVs are highly affected by the environment’s conditions (light, weather etc.) and often produce video stream containing numerous jitters, noises and other artifacts. Thus one needs to apply different filtering techniques to preprocess the video stream and thus to improve the efficiency of vSLAM methods.

On top of that, all vSLAM methods are prone to accumulating error [4] and that negatively affects the accuracy of constructed map and trajectory. One way to correct this error, and thus to increase the overall performance, is to handle, i.e. detect, loop-closures - see fig. 1. More precisely one needs to detect that the current image comes from an already perceived scene and, in case it’s true, correct the map and the trajectory.

![Fig. 1. Solving SLAM problem with and without use of the loop-closure algorithm. (a) A raw map obtained with monocular vision-based SLAM method. The inner curve represents the trajectory of mobile robotic system. The outer points represent the map. (b) Trajectory and map optimized with the loop-closure algorithm](image)

In this paper, we focus on improving the accuracy and performance of loop-closure detection algorithms. The ultimate goal is to keep the algorithm as robust and fast as possible along with making it compatible with dense, semi-dense and feature-based vSLAM methods. We introduce two enhancement steps (within the loop-closure detection algorithm) that contribute towards reaching this goal.

The latter of the paper is organized as follows. In section 2, we present a brief overview of existing methods. Section 3 introduces our implementation of loop-closure algorithm. The experimental results, showing the accuracy and performance of implemented algorithm, are given in section 4. Section 5 concludes.
2 Loop-closure methods

Accumulating error is one of the main bottlenecks of almost all known monocular vSLAM methods and algorithms. Even state-of-the-art algorithms suffer from this [5]. At the same time, results of numerous feasibility studies show, that detecting loop-closures can drastically improve the overall performance of monocular vSLAM. No wonder many of the vSLAM methods have loop-closure detection procedures built-in [6,7,8]. There exist also standalone loop-closure detectors [9,10] that may be plugged in to some of the vSLAM methods.

The earlier work [11,12,13] mostly rely on the so-called global loop detection, when the current image was compared against all previous visual data. This approach is quite reliable, but comes at the cost of high computation load and memory usage as one needs to keep all the information (such as keypoints, intense areas, depth map etc.) for every image processed during algorithm runtime. This leads to poor scaling for large environment localization and mapping. The recent approaches [14,15,16,17,18] use different constraints (e.i. using keyframes for keypoint matching) to optimize the time required for loop detection and map correction, but their usage is usually limited to specific vSLAM method.

In general loop-closure detection algorithms can be classified into three groups [19]: map-to-map, image-to-map and image-to-image:

- map-to-map loop closure is done by splitting the global map into sub-maps and finding correspondences between them [20].
- image-to-map performs the search of the matches between image and a map and recovers the system’s position, relative to the map [21].
- image-to-image founds a correspondences between images, usually based on vocabulary of image features [22].

Map-to-map approach is very intense performance-wise, since it deals with large amount of information on each iteration while comparing sub-maps. As the result it scales poorly to large environments. Image-to-map approach is fast and accurate, but in practice it is very memory intensive because one needs to store both point-cloud map and all the image features. The image-to-image loop-closure scales well to large environments, and can be computed fast with feature based approaches, but highly relies on a vocabulary. Thus one can infer that a combination of different approaches is desirable to reach higher performance while keeping the accuracy and the robustness at the high level. In this work we propose a solution that contributes towards this goal.

Proposed loop-closure method aims to combine image-to-image and image-to-map approach to achieve scalability, robustness and accuracy of both approaches, while keeping moderate runtime and low memory usage. Besides the proposed method is compatible with a large number of existing vSLAM methods, including feature-based, semi-dense and dense vision-based SLAM methods (for monocular, stereo and RGB-D cameras) and can be seen as a general enhancement approach to loop-closure detection.
3 Proposed method

In a nutshell all loop-closure algorithms generally consist of the two steps: 1) loop detection, 2) global optimization. Loop detection aims at establishing that the particular image is part of the scene, that has already been captured by previous image sequences. The simple interpretation is that this may be a sign, that the robotic system has reached the place that had already been visited before. The global optimization is performed after the loop is detected. This step corrects the accumulated run-time error for both the map and the trajectory (in a background). The illustration of the loop-detection process is depicted on fig. 2.

![Fig. 2. Main steps of the loop-closure detection. (a) Given (current) setting. Green curve represents the trajectory of the robotic system, black points represent the mapped features. (b) The loop-detection step. Red points represent the matched features, e.g. the ones that are present on the current image (perceived in current position) and previously observed images. (c) Map and trajectory of the robot after loop-closure detection.](image)

Since the robotic system’s motion consists of continuous rotations and translations, we assume that the trajectory is continuous as well (unless vSLAM method’s tracking is lost), so loop-detection algorithm usually checks for trajectory loops once per $N$ images for performance optimization purposes. In cases, when tracking is lost, detection may be needed to recover the state and position of robotic system and rebuild the map.
We suggest 2 enhancement procedures to be performed while detecting the loop. They both aim at lowering down the number of features to be compared thus speeding up the algorithm. The enhancements include the image detection optimization and imposing geometric constraints. For fast and accurate image matching we found that storing a particular amount of informative keypoints (instead of all keypoints) for each image allows us to keep the image matching accuracy. Also, the keypoints search area can be reduced to the only mapped points. The image comparison search area can be reduced by the geometric constrained, that is based on current camera position. We choose only images from the field, that may be observed from the camera in current position. High-level pseudocode of the loop-detection algorithm with the abovementioned procedures built-in is shown as Algorithm 1.

**Algorithm 1** The proposed loop-closure detection algorithm.

1. Get an image from video flow
2. Extract keypoints and get their descriptors from corresponded mapped points on image
3. Get and store $K$ informative features from image
4. if The trajectory loop is in camera search area
5. Match corresponding images in search area with current image
6. if the correspondence found
7. Perform the map optimization
8. endif
9. endif

First procedure (line 3) affects the feature extraction area of image. Dense and semi-dense vision-based SLAMs points of image with high gradient of intensity for depth map computation and mapping purposes. Thus, we reduce the extraction area by using only high gradient pixels, that were previously chosen to reconstruct the 3D space from 2D image. This allows us to avoid the image areas that are not going to be mapped anyway and provides an opportunity to reduce feature extraction process time. We limit the keypoints amount per any image in video flow to $K$. This $K$ keypoints with their descriptors are stored during loop-closure algorithm run-time since the number of keypoints per frame is relatively small (see section 4).

Second procedure (line 4) is the loop detection search area limitation. This allows to identify the patch on the whole trajectory that, with high probability, has a loop-closure point in it (i.e. the place, where the robot has already been). Assuming the robotic system’s motion is mostly horizontal, we project the motion vector and continue it with a straight line. Then we draw a perpendicular to this line. If the perpendicular intersects the built trajectory, then we draw a $\alpha$ degree line between normal and the projected motion line. The closest position (with corresponding image) to the point of intersection is going
Fig. 3. Loop detection search area. Red vectors shows the bounds for image comparison. Blue dot represents the start point for loop detection algorithm. And the trajectory picked out with blue color is our search area.

to be a start point for loop detection algorithm with the whole loop detection area constrained by two points - the intersection of normal and motion line with trajectory.

If motion line has no intersection point, then a search starts from the initial position of vision-based SLAM algorithm. The illustration of suggested method is demonstrated in fig. 3.

More formally one can put it as follows. Assuming, that raw localization and mapping (without optimization) for each moment of time $t$ is done by vSLAM method. Thus, for a given moment of time $t$, we have a point cloud $M = \{m_i\}, i \in N$, that represents the map, sequence of images $I = \{I_1, I_2, ..., I_t\}$. For each image $I_t$ we have corresponding observation $z_t \in M$ and position vector $x_t$.

As a part of loop-closure detection algorithm, we project each position $x_t$ on $xy$ plain $x'_t = Px_t$. For each $x'_k$, $k < t - 1$, $k \in N$ we check if vector $\vec{w} = x'_{t-1} - s x'_t$, $s > 0$ has intersection point $p$ with any of vectors $\vec{v} = x'_{k-1} - x'_k$ and vector $x'_1 (-lx'_2), l > 0$. We assume the position $x'_k$ closest to intersection point $p$ to be the starting point for image matching. As an end point for image matching, we take the intersection point $p'$ of perpendicular $h$ to vector $x'_t, x'_{t-1}$. As the result, the current image matching with corresponded images from positions between points $p$ and $p'$.
3.1 Implementation

As the main image identifier for loop detection we’ve chosen ORB detector\[23\] as one of the most fast, robust and efficient feature detector. For each image we extract at least $K$ ORB features and store their oriented and rotated BRIEF\[24\] descriptors, that have high element sum, with associated images. The requirement of having element sum in BRIEF descriptors comes from their interpretation. Higher values mean higher intensity gradient at this points, that provides more robust feature matching. That means that such a keypoints are informative and can be stored for further image matching.

As a part of map and trajectory global optimization, we use one of the most popular and effective graph optimization framework $g^2o\[25\]$. That allows us to keep a high accuracy while optimizing map and trajectory in comparison to other modern vSLAM methods.

4 Experimental results

For performance and accuracy testing purposes of the developed method we use a Robot Operating System (ROS)\[26\], that provides a powerful tools for robotic algorithms researches in general and in vision-based SLAM testing in particular. The open-source realizations of ORB-SLAM and LSD-SLAM were taken as ones of the most popular feature-based and semi-dense SLAMs respectively.

| Dataset       | Method   | ORB Features |
|---------------|----------|--------------|
|               | ORB-SLAM | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Sequence 13   | ORB-SLAM | + | + | + | + | + | + | + | + | + | + | + |
|               | LSD-SLAM | - | - | - | + | + | + | + | + | + | + | + |
| Sequence 14   | ORB-SLAM | + | + | + | + | + | + | + | + | + | + | + |
|               | LSD-SLAM | - | - | - | + | + | + | + | + | + | + | + |
| Sequence 15   | ORB-SLAM | + | + | + | + | + | + | + | + | + | + | + |
|               | LSD-SLAM | - | - | - | + | + | + | + | + | + | + | + |
| Machine       | ORB-SLAM | - | - | - | + | + | + | + | + | + | + | + |
|               | LSD-SLAM | - | - | - | + | + | + | + | + | + | + | + |
| Foodcourt     | ORB-SLAM | - | - | - | + | + | + | + | + | + | + | + |
|               | LSD-SLAM | - | - | - | + | + | + | + | + | + | + | + |

The introduced method is used with raw point cloud output of this methods. The experiment was made using LSD-SLAM Dataset\[4\] KITTI vision benchmark suit\[5\] and Malaga Dataset\[6\] which video fragments was divided into

---

4 http://vision.in.tum.de/research/vslam/lsdslam
5 http://www.cvlibs.net/datasets/kitti/eval_odometry.php
6 http://www.mrpt.org/MalagaUrbanDataset
subsequences (distinguishing fragments with loops) to make the experimental research more relevant.

Fig. 4. Overall success rate of loop-closure detection algorithm with different amount of keypoints.

Fig. 5. The histogram shows run time (in seconds) for loop-closure algorithms used in ORB-SLAM and LSD-SLAM in comparison with our algorithm.

Fig. 6. Average translation error over 1.5 km distance for ORB-SLAM and LSD-SLAM with proposed loop-closure detection method.

Fig. 7. Average rotation error over 1.5 km distance for ORB-SLAM and LSD-SLAM with proposed loop-closure detection method.

We took the Sequences 13, 14 and 15 from KITTI dataset and Machine and Foodcourt Sequences from LSD-SLAM dataset, because that sequences contain trajectories with loop-closures. KITTI dataset includes ground truth, that allows us to compare the optimized trajectory with real one. For LSD-SLAM datasets we only test the performance of our algorithm and the accuracy in comparison with trajectories, built by LSD-SLAM and ORB-SLAM. For Malaga Dataset we took the whole 6th, 7th and 8th sequence, since they present single loop, and sequences 10 and 13 (which contain multiple loops) where divided into 7 and 3 single loop subsequences respectively. Thus, we used 13 sequences from Malaga Dataset.

The first experiment was made to test the minimum required ORB features (K value) for loop detection algorithm to function successfully (e.g. with 100%
success rate). The fig. 4 shows the results of such an experiment. The table 1 shows the if the loop was successfully detected depending on number of ORB features ($R$) used for image matching.

As was already mentioned in section 2, we need to store at least $K$ features to successfully match the images if the loop-closure occurred. The experimental data shows that $K = 15$ is a minimum value for loop to be detected. The presented result also allows us to dramatically reduce the memory usage, since we don’t have to store hundreds of BRIEF descriptors, and increase the overall performance by the average of 7-10% in comparison with LSD-SLAM’s and ORB-SLAM’s loop-closure algorithms as shown in fig. 8.

For KITTI and Malaga sequences, the trajectory ground truth is presented, so we tested our algorithm using the available data. Fig. 6 shows the ground truth trajectory and the trajectory optimized with our method. The overall error values vary from 1.5% to 2.5% that is comparable with LSD-SLAM’s and ORB-SLAM’s loop-closure precision.

![Ground Truth comparison between LSD-SLAM, ORB-SLAM and developed method for Sequence 13.](image)

The overall precision depends on trajectory’s length and geometry. We found that longer trajectories with multiple loops give more accurate trajectory optimization for our method, while being more time consuming.
5 Conclusion

We have developed the original loop-closure method, that can be used for dense, semi-dense and feature-based vSLAM methods. The introduced optimization techniques showed, that the combination of image-to-image approach for loop detection and image-to-map approach for global optimization keeps an accurate trajectory error correction (around 1.5-2.5% translation error) while decreasing process time by 7-10%.

We found, that introduced method works in large outdoor environment without major issues. The experimental results showed, that described method can be used for mini unmanned aerial vehicle autonomous navigation tasks, even onboard.

Acknowledgment This research was supported by Russian Foundation for Basic Research. Grant 15-07-07483.

References

1. Fu, C., Olivares-Mendez, M.A., Suarez-Fernandez, R., Campoy, P.: Monocular visual-inertial slam-based collision avoidance strategy for fail-safe uav using fuzzy logic controllers. Journal of Intelligent & Robotic Systems 73(1-4) (2014) 513–533
2. Weiss, S., Scaramuzza, D., Siegwart, R.: Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. Journal of Field Robotics 28(6) (2011) 854–874
3. Handa, A., Whelan, T., McDonald, J., Davison, A.J.: A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In: Robotics and automation (ICRA), 2014 IEEE international conference on, IEEE (2014) 1524–1531
4. Strasdat, H., Montiel, J.M., Davison, A.J.: Visual slam: why filter? Image and Vision Computing 30(2) (2012) 65–77
5. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. In: arXiv:1607.02565. (July 2016)
6. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European Conference on Computer Vision, Springer (2014) 834–849
7. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Transactions on Robotics 31(5) (2015) 1147–1163
8. Mur-Artal, R., Tardós, J.D.: Visual-inertial monocular slam with map reuse. IEEE Robotics and Automation Letters 2(2) (2017) 796–803
9. Angeli, A., Filliat, D., Doncieux, S., Meyer, J.A.: Fast and incremental method for loop-closure detection using bags of visual words. IEEE Transactions on Robotics 24(5) (2008) 1027–1037
10. Strasdat, H., Davison, A.J., Montiel, J.M., Konolige, K.: Double window optimisation for constant time visual slam. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2352–2359
11. Botterill, T., Mills, S., Green, R.: Bag-of-words-driven, single-camera simultaneous localization and mapping. Journal of Field Robotics 28(2) (2011) 204–226
12. Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V., Fua, P.: View-based maps. The International Journal of Robotics Research 29(8) (2010) 941–957
13. Cummins, M., Newman, P.: Probabilistic appearance based navigation and loop closing. In: Robotics and automation, 2007 IEEE international conference on, IEEE (2007) 2042–2048
14. Mur-Artal, R., Tardós, J.D.: Probabilistic semi-dense mapping from highly accurate feature-based monocular slam. In: Robotics: Science and Systems. (2015)
15. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In: Experimental robotics, Springer (2014) 477–491
16. Vokhmintsev, A., Timchenko, M., Yakovlev, K.: Simultaneous localization and mapping in unknown environment using dynamic matching of images and registration of point clouds. In: Industrial Engineering, Applications and Manufacturing (ICIEAM), International Conference on, IEEE (2016) 1–6
17. Buyval, A., Gavrilienkov, M.: Vision-based pose estimation for indoor navigation of unmanned micro aerial vehicle based on the 3d model of environment. In: Mechanical Engineering, Automation and Control Systems (MEACS), 2015 International Conference on, IEEE (2015) 1–4
18. Afanasyev, I., Sagitov, A., Magid, E.: Ros-based slam for a gazebo-simulated mobile robot in image-based 3d model of indoor environment. In: International Conference on Advanced Concepts for Intelligent Vision Systems, Springer (2015) 273–283
19. Eade, E., Drummond, T.: Unified loop closing and recovery for real time monocular slam. In: BMVC. Volume 13. (2008) 136
20. Clemente, L.A., Davison, A.J., Reid, I.D., Neira, J., Tardós, J.D.: Mapping large loops with a single hand-held camera.
21. Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., Tardós, J.: An image-to-map loop closing method for monocular slam. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE (2008) 2053–2059
22. Cummins, M., Newman, P.: Fab-map: Probabilistic localization and mapping in the space of appearance. The International Journal of Robotics Research 27(6) (2008) 647–665
23. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2564–2571
24. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. Computer Vision–ECCV 2010 (2010) 778–792
25. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g 2 o: A general framework for graph optimization. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE (2011) 3607–3613
26. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. Volume 3, Kobe (2009) 5
27. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research 32(11) (2013) 1231–1237
28. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2012)
29. Blanco-Claraco, J.L., Moreno-Dueñas, F.Á., González-Jiménez, J.: The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. The International Journal of Robotics Research 33(2) (2014) 207–214