2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Real-Time 6-DOF Monocular Visual SLAM in a Large-Scale Environment

Hyon Lim[1], Jongwoo Lim[2] and H. Jin Kim[1]

***Abstract*— Real-time approach for monocular visual simultaneous localization and mapping (SLAM) within a large-scale environment is proposed. From a monocular video sequence, the proposed method continuously computes the current 6-DOF camera pose and 3D landmarks position. The proposed method successfully builds consistent maps from challenging outdoor sequences using a monocular camera as the only sensor, while existing approaches have utilized additional structural information such as camera height from the ground. By using a binary descriptor and metric-topological mapping, the system demonstrates real-time performance on a large-scale outdoor environment without utilizing GPUs or reducing input image size. The effectiveness of the proposed method is demonstrated on various challenging video sequences including the KITTI dataset and indoor video captured on a micro aerial vehicle.**

I. INTRODUCTION

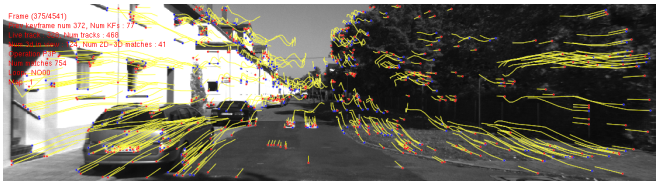
Simultaneous Localization and Mapping (SLAM) is the way of building a consistent map within an unknown environment while keeping track of the current location at the same time. This problem has been studied over the past three decades in robotics and recently in computer vision fields. As cameras become ubiquitous in many robot systems, visual SLAM research draws increasing interests. As a result, the visual SLAM problem has been acknowledged as a key challenge to enable fully autonomous robots without resorting to external positioning systems such as GPS or heavy and expensive laser range finders.

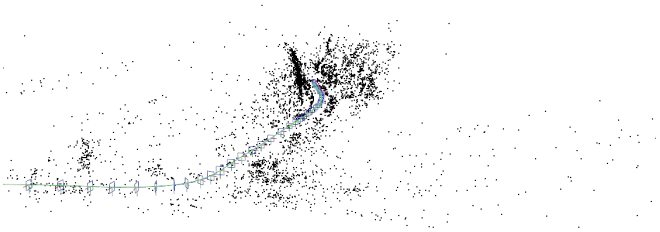
The visual SLAM methods are classified into two main categories by the number of cameras employed: monocular and stereo. The monocular systems have several advantages over stereo systems in terms of cost, flexibility, and computational efficiency. A single camera always costs less than stereo camera systems, and also provides flexibility in installation of the camera to robots. For example, a stereo camera should have more than a half meter baseline for enough disparity when it is operated in a car for outdoor navigation [1]. However, robots like micro aerial vehicles (MAVs) may not have the space for a wide baseline stereo camera at all.

Despite the advantages of a monocular camera, the nature of a monocular camera, which only provides bearing angles of visual features, has made monocular visual SLAM difficult [2], [3]. Therefore, a solid temporal feature association method is critical for monocular visual SLAM systems to achieve performance and stability comparable to stereo system.

In many robotics literature, the *visual odometry* (VO) also has been intensively studied. The main difference between VO and visual SLAM is *loop closing* capability. By closing loops, localization error caused by drift, especially in a large environment, can be reduced considerably. Once a relative pose of the current frame to another frame in the map is computed, it can be used to resolve the accumulated error along the path. Consequently, real-time loop closing will be a key challenge for real-time visual SLAM in a large-scale environment.

In this paper we propose a new approach for real-time monocular visual SLAM in a large-scale environment[1]. At its core lies a fast keypoint tracker and scene recognizer using binary feature descriptors, and use of a relative representation





(a) From monocular sequences, the proposed method builds 3D map successfully in a large-scale environment.

(b) A scale consistent environment map with closed loops which is denoted by red square. Ground-truth trajectory is displayed on the right side for qualitative comparison.

Fig. 1: **Input and outputs of the proposed method.** The environment map and camera poses obtained by the proposed method using monocular sequences.

[1]School of Mechanical and Aerospace Engineering, Seoul National University, Korea. {hyonlim,hjinkim}@snu.ac.kr
[2]Corresponding Author, Division of Computer Science and Engineering, Hanyang University, Korea. jlim@hanyang.ac.kr

[1]Video of the system is available at http://goo.gl/d4vA3Z

that is a hybrid between metric Euclidean environment mapping and topological mapping.

Our contributions can be summarized by the followings:

- Use of a homogeneous type of descriptor (binary descriptor) for tracking, mapping and loop closing in real-time. Existing approaches have been using separate descriptors (e.g., SIFT, SURF) for loop closure detection which require additional computation and storage space (see section III-A for further details).
- Construction of large-scale maps (several kilometers) using a monocular camera, and fully autonomous operation without a special map bootstrapping process.
- Use of a relative pose representation in the environment map to enable immediate loop closure without delay and efficient global adjustment process (see section II-E)

The remainder of the paper is organized as follows: in Section II, we review related works, in Section III the pipeline of the proposed method is described in detail, in Section IV, the mapping and optimization framework is presented and in Section V, we present experimental results. We discuss conclusion and future work in Section VI.

II. RELATED WORK

In this section, existing approaches will be reviewed, and compared along with our contributions listed above.

A. Monocular Visual SLAM

A seminal monocular visual SLAM system was proposed by Davison *et al.* in 2003 [2]. The algorithm uses an Extended Kalman Filter (EKF) for map and pose estimation. Due to the complexity of EKF-based SLAM, this system works real-time up to approximately 100 features in the map. The EKF can minimize the error when a loop constraint is provided by feature association. It only works with smooth camera motion as image patches are used as the feature which are less robust compared to BRIEF, SIFT or SURF.

The parallel tracking and mapping (PTAM) algorithm by Klein *et al.* in 2007, introduced the idea of running tracking the camera pose and mapping of the environment in two simultaneous pipelines [3]. The system can manage thousands of features with real-time performance thanks to a separate mapping process. However, PTAM requires the initial user input with camera motion to be a straight translation to bootstrap the pose estimator, thus it is not fully autonomous system. Furthermore, the number of landmarks that can be managed by PTAM is inadequate for a large-scale environment.

Strasdat *et al.* proposed the method for scale consistency of monocular visual SLAM [4] based on the [3]. In [4], $Sim(3)$ representation which includes scale in the pose, was proposed which gives an additional degree of freedom to monocular bundle adjustment (BA) to remove scale drift.

To summarize, there are two approaches in monocular visual SLAM. One is based on filtering that represents all camera poses and the map as a single state vector [2]. The other one is based on BA to optimize pose and map from initial estimates [3], [4]. The main advantages of either approach have been evaluated and summarized in [5].

B. Monocular Visual Odometry

Apart from visual SLAM systems, visual odometry (VO) also has been actively studied, and shares some key technologies. Nister *et al.* demonstrated the seminal real-time monocular VO system [6] in 2004, where RANSAC based 3D-to-2D pose estimation is used to obtain relative pose between frames. To keep RANSAC process light, they utilize five-point relative pose estimation for motion hypothesis. This process pipeline is widely used in several other works including [3]. Mouragnon *et al.* proposed a real-time localization and 3D reconstruction [7] in 2006, which can handle a few thousands of landmarks in less than one hundred keyframes by utilizing local BA. However, no loop closure detection has been introduced in this work, which is the key to large-scale visual SLAM.

C. Graph-based Formulation

Recently, a graph-based formulation of SLAM has drawn attention. This formulation represents robot poses and landmarks as nodes. Sensor observations are represented as edges in the graph, which form constraints between robots and landmarks. Kümmerle *et al.* [10] proposed a general framework for optimization of nonlinear least squares problems on graph representation. Lim *et al.* [11] proposed a hybrid representation of metric Euclidean and topological map.

For efficient loop closing, Strasdat *et al.* [12] proposed double-window optimization framework. Therein, two types of optimization techniques were described based on co-visibility metric. One is landmark-pose optimization and the other one is pose-graph optimization. When a loop is detected, pose-graph optimization [4] gives fast convergence of poses while landmark-pose optimization enhances the details of the local area.

D. Loop Detection and Closing.

The result of the inherently incremental SLAM process produces drift [13]. This drift becomes large when the camera travels for a long distance. To overcome this problem loop detection and closure are required. The loop detection has been done by the appearance-based scene matching method [14] which shares the core idea with the vocabulary tree method [15] for fast large-scale image retrieval. Traditionally, distinguishable descriptors like SIFT or SURF are used in scene matching, but their computational overhead has degraded the performance of visual SLAM system. As a remedy, a new type of binary descriptor has been introduced [16] in 2010. After a year later, Gálvez-López *et al.* [17] proposed a fast scene recognition method using a vocabulary tree of binary descriptors.

E. Relative Representation

Several coordinate definitions have been proposed in SLAM literature. One typical approach is *global coordinates* which has an origin, and every new poses are described

with respect to it. Some researchers have introduced *robo-centric, ego-centric coordinates* [18]. The origin of this system is the current coordinate frame of the robot, and other poses and landmarks are defined by the robot's coordinate system. The disadvantage of these notations is that not all measurements are available between the current pose and all landmarks. Therefore, some of landmarks will be described by integration of error-corrupted poses. This error is difficult to be minimized or very slowly converges, which is impractical in many cases. Our representation resembles with *robo-centric* coordinate system. However, keyframes are described by relative transformation which is quite accurate compared to other estimates. A landmark is defined with its *anchor frame* which is a frame that observes the landmark for the first time.

Mei et al [19] proposed Continuous Relative Representation (CRR). The idea is similar as ours, but no specific preference of frame origin to describe keyframes. We describe the pose graph only with relative transformation as an edge between pose vertices.

III. MONOCULAR SLAM PIPELINES

Robust feature tracking is an important stage of the monocular visual SLAM. To compute a 3D position of a feature, it should be tracked until it has reasonable disparity to be triangulated. In this section, we describe the steps performed for each input image.

A. Pipeline overview

Keypoints are extracted by the FAST [20] detector. The keypoints are tracked in the following frames by matching them to the candidate keypoints within their local search neighborhood. The BRIEF [16] binary feature descriptor which is very light to compute, is used to match the keypoints. This fast tracker is tightly integrated with the scene recognition module that is using the bag of *binary visual words* approach [17] to find nearby keyframes and the matching 3D landmarks in the map. Thanks to the efficiency of binary descriptor matching, scene recognition can be done at frame rate.

Typically, the loop closing (i.e., scene recognition) has been performed by using separate full-featured descriptors like SIFT [21] or SURF [22] which are expensive to compute in real-time. The main reason is that those descriptors give considerably high matching rate among the existing approaches. However, those descriptors are not used in temporal feature association for pose estimation due to its high computation cost. In other words, these expensive descriptors should be extracted and managed separately from the features used by the keypoint tracker (e.g., image patch used with Kanade-Lucas-Tomasi tracker [23]) to query matching keyframes to the current keyframe in the database.

In this paper, we have developed unified keypoint tracking and scene recognizing system by using homogeneous type of descriptor: the BRIEF descriptor. A new keypoint tracker is developed based on a binary descriptor to exploit the efficiency of matching of binary descriptors for scene recognition. As a result, we are able to perform full visual SLAM tasks (feature tracking, map building and loop closure) at 10 fps on images from the KITTI dataset (image size of 1241×376 capture at 10 fps) without elaborating parallelization using GPUs. To our knowledge, this is the first approach that uses a single type of feature descriptor in both keypoint tracking and loop closure to achieve real-time performance.

The proposed system performs the following steps to every input image:

1) (FEATURE EXTRACTION) Extract keypoints and descriptors from a new image.
2) (FEATURE TRACKING) Associate new features to previously extracted features.
3) (KEYFRAME ADDITION) Add a keyframe when the number of tracked features are less than a threshold or the motion of the camera is larger than a threshold.
   - (KEYFRAME CREATION) A frame is created with relative pose with respect to the previous keyframe and inserted.
   - (LOOP CLOSURE DETECTION) Tracked BRIEF descriptors from step 2 are queried to the binary vocabulary tree. No additional descriptor extraction is required. In case a loop is detected, associate two frames with the 2D-3D pose obtained from the query.
   - (LOCAL BUNDLE ADJUSTMENT) Local bundle adjustment is performed with a fixed-size window including the inserted keyframe.
4) (POSE COMPUTATION) If enough number of features are matched to the most recent keyframe, the relative pose of the current frame is computed by RANSAC with the 3-point pose estimation [26] followed by non-linear refinement.

B. Keypoint extraction and tracking

We have used the multi-scale FAST corner algorithm which extracts FAST corner over multiple scale to deal with large-scale outdoor scene. Multi-scale feature gives longer tracks compared to single scale approaches at an additional cost. However, if computational resource is not enough for the multi-scale approach, we found that single scale with image partitioning also works in practice. The input image is partitioned with the predefined number of grids (e.g., 7×4). The FAST corners are extracted in each partitioned image. This approach makes feature location evenly distributed in an image, which will enhace pose estimation quality.

To track keypoints from frame to frame, the existing features tracked are compared to all the keypoints in the current frame within $w \times w$ window around its respective keypoint position in the previous frame. We extract the 256-bit BRIEF descriptor on the selected candidate within the window instead of precomputing all descriptors in the current frame. This will prevent unnecessary descriptor computation which will degrade the system performance. For the keypoints within the window, the BRIEF descriptor of two keypoints are compared, and the keypoint that has lowest Hamming distance s_0 is accepted as the best candidate. We further

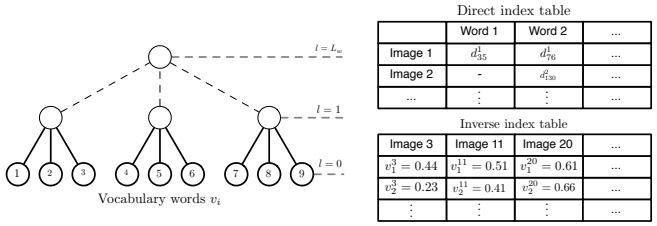


Fig. 2: **Vocabulary tree with direct and inverse indices.** The v_i^j denotes the value of visual word i for image j. The direct index d_j^i denotes the descriptor j that has a visual word i.

investigate Hamming distance of the second-best candidate s_1 to compute the ratio between two scores. When the ratio between the best and second-best match s_0/s_1 is less than τ, the best candidate is accepted as a matched keypoint. We set parameters $\mu = 48, w = 60$ and $\tau = 0.85$ in all experiments.

C. Pose estimation

In our implementation, two types of pose estimation exist. One is for tracking of map landmarks, and the other one is for keyframe initialization when 3D points are not available. Algorithm details are described below.

Keyframe initialization. When the system is initialized from scratch, there are no 3D landmarks available as we consider a monocular camera. We only have 2D-2D keypoint matches when the first keyframe is added. In this case, we use the 5-point algorithm to two selected frames [24] with RANSAC [25] to estimate the relative pose. The second view is selected if the number of features tracked from the first view is less than the threshold.

Environment map tracking. Each 3D landmark has an associated BRIEF descriptor, so it can be matched with 2D points for pose estimation. A 3D point is projected to the image space based on the prior pose, then we search 2D-3D correspondences with the idea of keypoint tracking in section III-B. When 2D-3D correspondences are available, the 3-point pose estimation [26] with RANSAC gives the relative pose (R, t). The computed pose is further optimized using the Levenberg-Marquardt algorithm by minimizing reprojection error.

Pose estimation with loop closure. In every keyframe addition, the tracked BRIEF descriptors are provided to the vocabulary tree. When we pass the descriptors to the loop closure detector, it seeks the best keyframe matched. We associate each 3D point to a keypoint which is passed to the vocabulary tree to utilize direct index. This will be explained in the following section.

D. Loop closing

Scene recognition. Our feature tracker will provide the most recent BRIEF descriptor tracked until the current frame. These descriptors are provided to the vocabulary tree as shown in Fig. 2. A binary descriptor is searched in the tree, by selecting at each level based on the Hamming distance.

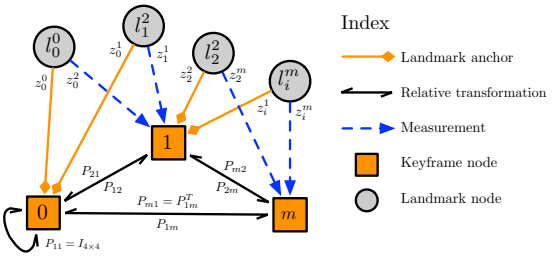


Fig. 3: **Graph representation of topological structure of the landmarks and keyframes.** Four landmarks and three keyframes are assumed in this example. Edges and vertices encode relative pose and landmark position. Outputs after traversing these topological nodes are metrically reconstructed environment map and keyframes.

Loop closing and guided matching. In stereo visual SLAM, loop closing is usually done by verifying 3D-3D point correspondences by comparing descriptors associated to the 3D points. In monocular cases, the 3D point in the current frame is not always available. Therefore, 2D-3D matching is known to be the most efficient and accurate way for pose estimation [6]. We used a vocabulary tree with $k_w = 10$ branches and $L_w = 6$ depth levels.

To mitigate additional overhead except querying the current image descriptors to the vocabulary tree, we utilize direct index [17] as an initial guide of loop closure (See Fig. 2). The direct index is the list of visual words associated with keypoint indices. This method is based on the fact that similar descriptors will have the same visual word. The direct index is updated when new image arrives by collecting features in each word in the vocuabulary tree. In addition to [17], we associate 3D landmark indices for relative pose computation. As similar descriptors will have the same visual word, associating descriptors with the same visual word mitigates the computational overhead in the separate feature matching process for 3D-2D pose computation. As a result, we can obtain 2D-3D matching right after all features traverse the vocabulary tree, which is required for place recognition.

Further geometric verification is performed with RANSAC. Outliers will be removed during the RANSAC process. If enough inliers are found, nonlinear optimization minimizing reprojection error is performed to obtain accurate relative pose with respect to matched frame. We add this relative pose and 2D-3D observations to the graph, further local bundle adjustment will be performed (local bundle adjustment with the breadth first search over this loop as described in [27]).

IV. REPRESENTATION AND OPTIMIZATION

In this paper, we use a hybrid representation of a fully metric Euclidean environment map and a topological map [27]. In this representation, only the relative poses between keyframes are encoded as edges in the *keyframe pose graph*

instead of the keyframes in the global coordinate system. The metric information (e.g., position and orientation with respect to origin) is embedded by integrating edges from the origin keyframe to the target keyframe [27] when required.

A. Environment representation

Suppose we have n landmarks observed from m keyframes in the environment map. The relative pose (camera projection matrix) between keyframes k and j is denoted by

$$P_j^k \in \mathbb{R}^{4\times4}, \qquad k, j \in \{0, \dots, m\}, \tag{1}$$

where k, j denote reference and target keyframe number respectively. The inverse relation $P_j^k = P_k^{j^{-1}}$ is also satisfied. A landmark is defined by

$$l_i^{k_i} \in \mathbb{R}^4, \qquad i \in [0, n], k_i \in [0, m], \tag{2}$$

where k_i is the unique anchor keyframe which defines the reference coordinate system of the landmark i. This means that the landmark i is observed for the first time in the anchor keyframe k_i. The location of the landmark i in another keyframe m, which has an anchor keyframe k_i is computed as follows:

$$l_i^m = P_{k_i}^m l_i^{k_i}. \tag{3}$$

The measurement of landmark i in the keyframe k is defined as the following normalized homogeneous image coordinates:

$$z_i^k \in \mathbb{R}^3, \qquad i \in [0, n], k \in [0, m]. \tag{4}$$

Given a landmark l_i^m whose anchor keyframe is m, the image coordinate of the landmark in another keyframe k can be computed as

$$z_i^k = [K_{3\times3}|0]_{3\times4} P_m^k l_i^m, \tag{5}$$

where the $K_{3\times3}$ is the intrinsics of the camera.

We form a graph of keyframes and landmarks with edges P_j^k, z_i^k incrementally in the topological map as the camera moves. Extra edges will be added when loop closure is detected. Fig. 3 shows an illustrative example of our notation.

This representation has several advantages compared to the global coordinate system that stores keyframe pose and landmark from the origin. The main benefit of using the relative representation is that it is very easy and fast to maintain the local map accuracy even when a large loop closure occurs. However, metric consistency along the edges may be violated due to inaccurate pairwise pose estimates. To quickly fix this error, we use the pose optimization proposed in [8]. When enough computation resource is available, the environment map is further optimized via full bundle adjustment.

B. Optimization

To optimize the poses and landmarks, we create a metric environment map by embedding metric information to nodes by breadth-first search over graph. The breadth-first search weighted by its distance from the reference keyframe is performed, and the visited keyframes are registered in the temporary global coordinate system. This is illustrated in Fig. 4. We manage our landmarks and keyframes by utilizing topological map illustrated in Fig. 4.

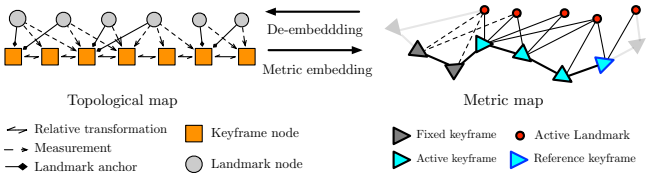
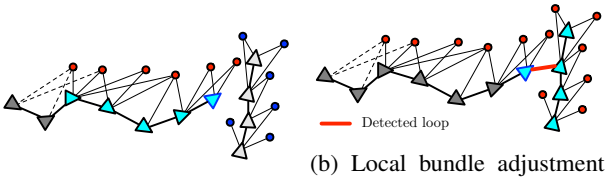


Fig. 4: **Our hybrid environment representation.** We use both topological and metric map to represent the environment. We determine active keyframes based on window size w for local bundle adjustment. The topological map stores only relative information in edges while the metric map contains location of nodes with respect to the specified origin.



(a) Local bundle adjustment. (b) Local bundle adjustment with a detected loop.

Fig. 5: **Local bundle adjustment (LBA).** Active keyframes are selected based on graph search with distance weight. (a) an example of normal LBA situation without loop closure. The fixed keyframes are selected based on a common landmark. (b) Matched loop segments will be included in LBA as breadth-first search will active the keyframes. We perform the pose graph optimization first, to make all poses metric consistent.

As the topological map lacks the metric information for optimization (e.g., keyframe pose in some defined coordinate system), we perform a metric embedding operation over the topological map with a fixed window size. After poses and landmarks are optimized, de-embedding of metric information is performed. This operation updates the relative transformation between two adjacent keyframes in the topological map.

A local adjustment is performed when a new keyframe is captured. Local adjustment is illustrated in Fig. 5. The size of the sliding window and number of iterations can be adjusted considering the system performance.

V. Experimental Result

We evaluate our visual SLAM system using the KITTI dataset [1] and a monocular sequence from a micro-aerial vehicle (MAV). The computation was performed on a laptop with an 2.3 Ghz Intel Core i7. GPU and multi-theading are not utilized except within the ceres solver [28].

A. Datasets description

The KITTI dataset provides 22 sequences in total. Among 22 sequences, 11 sequences are provided with ground truth data. The images are undistorted 1241×376 monochromatic. This dataset is quite challenging because the vehicle speed varies from 0 to 90 kph, the frame capture rate is low as

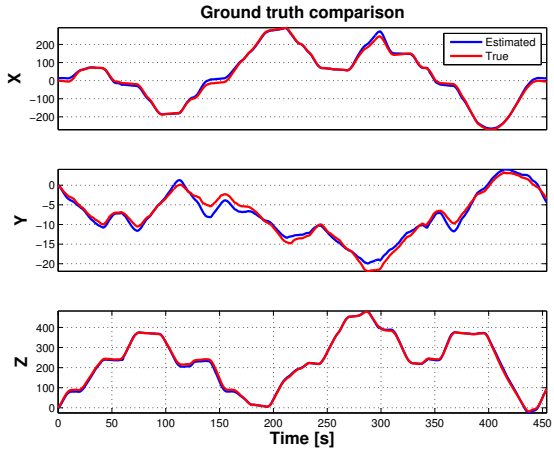
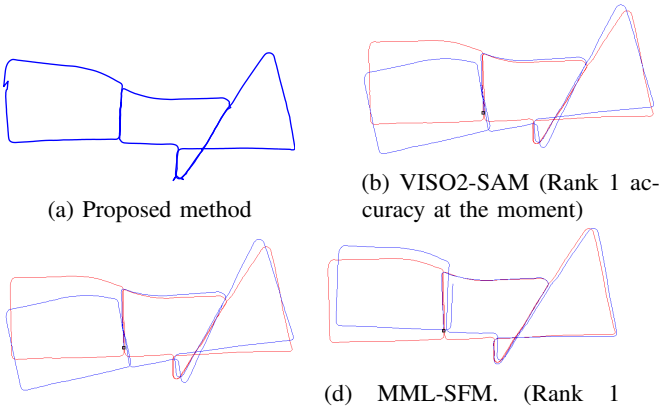


Fig. 6: **Ground truth comparison result of sequence 0 of the KITTI dataset.** The global arbitrary scale was obtained by the optimization process described in section V-B. The position error (mean $\pm$ std. dev.) is $2.9383 \pm 8.9118\ m$.



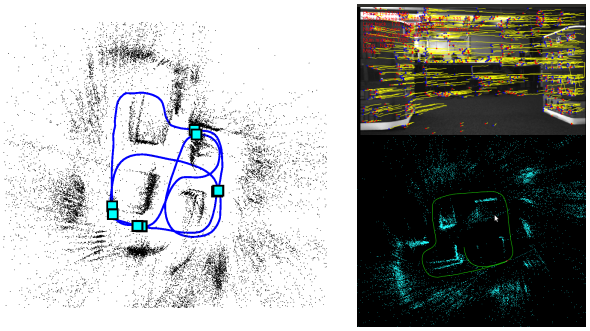
(a) Proposed method

(b) VISO2-SAM (Rank 1 accuracy at the moment)

(c) State-of-the-art stereo visual odometry [29]

(d) MML-SFM. (Rank 1 monocular visual odometry algorithm.

Fig. 7: **Qualitative comparison with state-of-the-art.** As our result is available up to scale, only a qualitative comparison is available at the moment. The proposed algorithm shows the best results compared to the best ranked algorithm in KITTI ranking system. (Blue) estimated trajectory (Red) ground truth



(a) Reconstructed result by proposed method.

(b) Screen capture of the system.

Fig. 8: **Indoor MAV sequence.** The experimental results with hand-carried MAV sequence. Squares denote loop closures.

10 Hz, and there exist many moving moving objects in the scene. Among the datasets we have chosen six sequences which include loop closures as shown in Fig. 11 and Fig. 1.

The MAV sequence is evaluated to demonstrate the 6-DOF capability of our algorithm. The relatively small camera displacements presents an additional challenge due to the reduced estimation quality (See Fig. 8).

Failure case. In the KITTI dataset, nine sequences have loop closures. However, two sequences (sequence 2,14) have a loop with images taken from the front and rear of the loop. This case is very difficult to correctly match, so we have skipped those sequences. Our algorithm failed to close the loop in sequence 9 because not enough frames were matched for loop closure.

B. Comparison with ground truth data

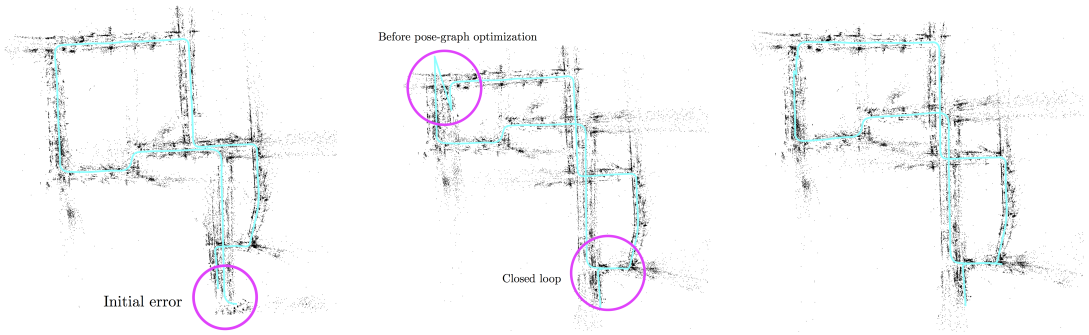
We consider a monocular camera without known geometry of the vehicle. As a result, the poses are available only up to arbitrary scale. To compare with the ground truth trajectory, we find a global transformation T. Our result is transformed by the obtained T to compare with the ground truth. The quantitative comparison result with ground truth is shown in Fig. 6.

Qualitative comparison to state-of-the-art is shown in Fig. 7. Unlike other visual odometry approaches, we have successfully detected most loop closures. As a result we have the most similar trajectory compared to the ground truth while others show drifted trajectory. Fig. 7b and 7d are results from the current best algorithm according to the KITTI dataset ranking system [1].

A large-scale loop closure is demonstrated in Fig. 9. In Fig. 9a, the binary vocabulary tree detects a keyframe matched to the current keyframe. The relative pose between the current and matched frames is computed by 2D-3D correspondences obtained from direct indexing of the vocabulary. No additional match is required. Immediately after the loop is detected and closed, the large accumulated drift is corrected, as shown in Fig. 9b. However, the metric property is broken at this moment (spike on pose graph). In Fig. 9c shows the accumulated errors are effectively distributed by optimization of pose graph. The metric property is recovered as shown in the above environment map (no spike in the pose graph).

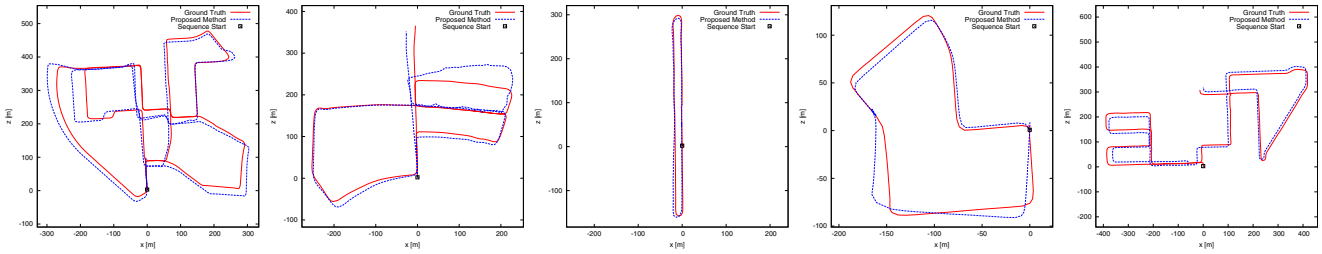
The accuracy of our method can also be qualitatively judged from Fig. 10, which shows the camera trajectories from our method and from the GPS ground truth. Except for a few regions (some parts of sequence 5 and sequence 6), the two trajectories are well-aligned. This confirms the scale consistency and accuracy of the position estimates. Some drifts are caused by the change in the scene quality due to vehicle speed and bumpy motion.

The mapping result in a large-scale environment is shown in Fig. 11. The proposed method successfully reconstructed landmarks around camera path.



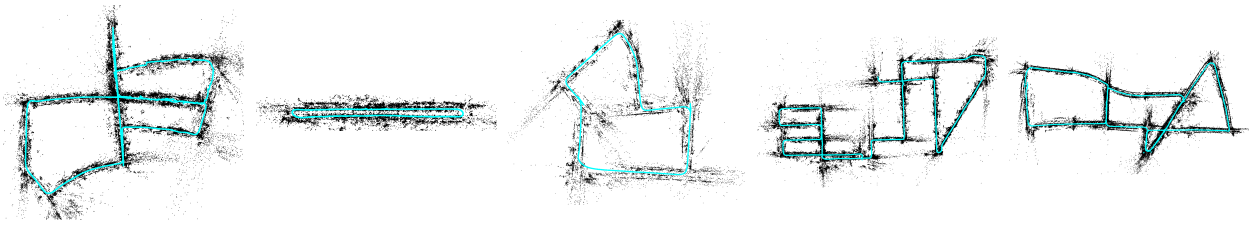
(a) (b) (c)

Fig. 9: **Loop closing.** Large-scale loop closure detected in sequence #0 of the KITTI dataset. The loop closure is marked in the figure. (a) A frame before the loop closure detection. Long distance travel without loop closing makes the current camera location drift. (b) Loop closure is performed by shifting the pose graph. This will correct drift immediately but the metric property of map may not invalid. (c) After pose-graph optimization, the corrected (pose error and scale) map is obtained.



(a) Sequence #00 (b) Sequence #05 (c) Sequence #06 (d) Sequence #07 (e) Sequence #08

Fig. 10: **Ground truth comparison.** Trajectories estimated by the proposed method (blue dotted line) and ground truth (red line) in KITTI dataset.



(a) Sequence #05 (b) Sequence #06 (c) Sequence #07 (d) Sequence #08 (e) Sequence #13

Fig. 11: **Environment mapping results of KITTI dataset.** Landmarks (black dots) are drawn with camera trajectory (cyan line). Our approach worked well for those sequences.

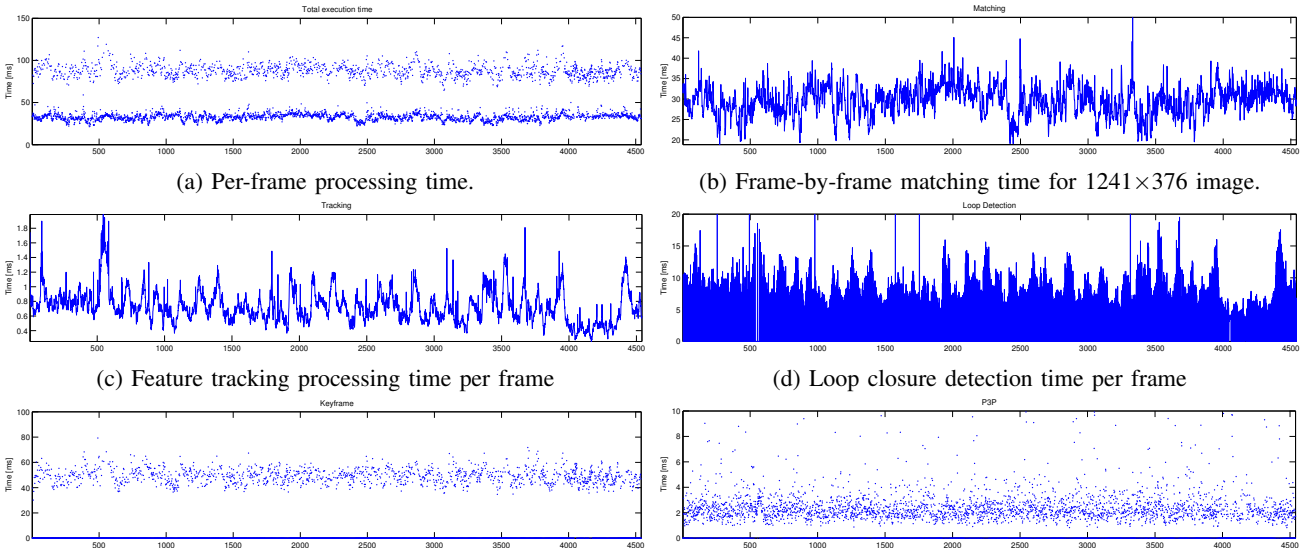
The run-time performance analysis of the system is shown in Fig. 12. In most cases, the proposed algorithm runs within 100 ms which denotes proposed algorithm is real-time for the KITTI dataset which was captured 10 fps.

VI. Conclusion

In this paper, we have presented a real-time monocular visual SLAM system in a large-scale environment. One unified binary feature descriptor is used both for tracking and scene recognition, which saves additional computation cost required in other systems. Our algorithm efficiently combines keypoint tracking with a bag of binary visual words to detect loop closures, which provides valuable information to prevent severe drift. Also we adopted relative representation for the environment map to achieve instant loop closure and pose-only optimization for efficient global structure adjustment. Our implementation can process the KITTI dataset at video rate (10 fps) without massive parallization, and the resulting maps have the higher quality compared to the state-of-the-art monocular visual SLAM systems.

Acknowledgement

This work was partly supported by the IT R&D program of MSIP/KEIT (No. 10047078), the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (No. 2013-013911),



(a) Per-frame processing time.

(b) Frame-by-frame matching time for 1241×376 image.

(c) Feature tracking processing time per frame

(d) Loop closure detection time per frame

(e) Keyframe addition operation time per frame.

(f) P3P with RANSAC processing time.

Fig. 12: **Runtime performance.** Per-frame processing time for sequence 0 of the KITTI dataset.

and a grant to Bio-Mimetic Robot Research Center funded by Defense Acquisition Program Administration (UD130070ID).

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361.

[2] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *ICCV*. IEEE, 2003, pp. 1403–1410.

[3] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *ISMAR*. IEEE, 2007.

[4] H. Strasdat, J. Montiel, and A. Davison, "Scale drift-aware large scale monocular slam," in *Proceedings of Robotics: Science and Systems (RSS)*, 2010, p. 5.

[5] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?" in *ICRA*. IEEE, 2010, pp. 2657–2664.

[6] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *CVPR 2004.*, vol. 1. IEEE, 2004, pp. I–652.

[7] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3d reconstruction," in *CVPR*, vol. 1. IEEE, 2006, pp. 363–370.

[8] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1066–1077, 2008.

[9] N. Snavely, S. M. Seitz, and R. Szeliski, "Skeletal graphs for efficient structure from motion," in *CVPR*, vol. 2, 2008.

[10] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *ICRA*, 2011.

[11] J. Lim, J.-M. Frahm, and M. Pollefeys, "Online environment mapping," in *CVPR*. IEEE, 2011, pp. 3489–3496.

[12] H. Strasdat, A. J. Davison, J. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in *ICCV*. IEEE, 2011, pp. 2352–2359.

[13] C. Engels, H. Stewénius, and D. Nistér, "Bundle adjustment rules," *Photogrammetric computer vision*, vol. 2, 2006.

[14] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[15] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*. IEEE, 2006.

[16] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," *Computer Vision–ECCV 2010*, pp. 778–792, 2010.

[17] D. Galvez-Lopez and J. D. Tardos, "Real-time loop detection with bags of binary words," in *IROS*. IEEE, 2011, pp. 51–58.

[18] B. Williams and I. Reid, "On combining visual slam and visual odometry," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3494–3500.

[19] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "Rslam: A system for large-scale mapping in constant-time using stereo," *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, 2011.

[20] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *ECCV*. Springer, 2006.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[22] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV*. Springer, 2006, pp. 404–417.

[23] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674–679.

[24] D. Nistér, "An efficient solution to the five-point relative pose problem," *PAMI*, vol. 26, no. 6, pp. 756–770, 2004.

[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[26] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *CVPR*. IEEE, 2011, pp. 2969–2976.

[27] J. Lim, J.-M. Frahm, and M. Pollefeys, "Online environment mapping using metric-topological maps," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1394–1408, 2012.

[28] S. Agarwal, K. Mierle, and Others, "Ceres solver," https://code.google.com/p/ceres-solver/.

[29] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 963–968.