Robotica: (2019) volume 37, pp. 1835–1849. © Cambridge University Press 2019
doi:10.1017/S0263574719000298

A Study of Sensor-Fusion Mechanism for Mobile Robot Global Localization

Yonggang Chen†Ł, Weinan Chen†*, Lei Zhu†, Zerong Su‡, Xuefeng Zhou‡, Yisheng Guan† and Guanfeng Liu§

†School of Electro-mechanical Engineering, Guangdong University of Technology, Guangzhou, China. E-mails: leizhu1016@gmail.com, ysguan@gdut.edu.cn
‡Guangdong Key Laboratory of Modern Control Technology, Guangdong Institute of Intelligent Manufacturing, Guangzhou, China. E-mails: zr.su@giim.ac.cn, xuefengzhou@vip.qq.com
ŁDepartment of Electro-mechanical Engineering, Dongguan Polytechnic, Dongguan, China. E-mail: yonggang44@163.com
§Guangdong Polytechnic Normal University, Guangzhou, China. E-mail: liugf1004@gmail.com

(Accepted February 26, 2019. First published online: April 22, 2019)

SUMMARY

Estimating the robot state within a known map is an essential problem for mobile robot; it is also referred to "localization". Even LiDAR-based localization is practical in many applications, it is difficult to achieve global localization with LiDAR only for its low-dimension feedback, especially in environments with repetitive geometric features. A sensor-fusion-based localization system is introduced in this paper, which has the capability of addressing the global localization problem. Both LiDAR and vision sensors are integrated, making use of the rich information introduced by vision sensor and the robustness from LiDAR. A hybrid grid-map is built for global localization, and a visual global descriptor is applied to speed up the localization convergence, combined with a pose refining pipeline for improving the localization accuracy. Also, a trigger mechanism is introduced to solve kidnapped problem and verify the relocalization result. The experiments under different conditions are designed to evaluate the performance of the proposed approach, as well as a comparison with the existing localization systems. According to the experimental results, our system is able to solve the global localization problem, and the sensor-fusion mechanism in our system has an improved performance.

KEYWORDS: Mobile robot; Global localization; Sensor fusion.

1. Introduction

Estimating the state of mobile robot relative to a given known environment representation has been studied popularly for decades; it is also called robot localization problem. Localization is an important function in most mobile robot applications. Without loss of generality, this problem is always divided into two types: tracking and global localization.

For the tracking problem, it assumes that an initial robot pose is given and the state of robot can be achieved by correcting incremental odometry information (wheel odometry, visual odometry, etc.), which try to maintain tracking over time in a known environment representation. However, localization without any prior knowledge[1] is much more difficult, which is referred to global localization. In a global localization problem, we try to estimate the state of robot by sensor feedback only. In the

* The first two authors contributed equally to this work. Corresponding author. E-mail: weinanchen1991@gmail.com

mobile robot area, kidnapping is defined as a problem that a well-localized robot is kidnapped and moved to another place while the robot has no information of the movement, and the relocalization of the robot is required after the kidnapping.

In recent years, vision-based localization and environment modeling for mobile robot systems are obtaining more and more interest. For vision-based localization, image-based localization (IBL) has been studied for a long time, and it solves the problem by matching a vision observation with a stored frame database, and then trying to estimate the pose of the frame observation. The IBL has been applied to global localization.[2] Also, this topic is popularly studied in robotics via VSLAM (Visual Simultaneous Localization and Mapping), which has been studied for over 20 years and many related studies have been proposed,[3–5] and the image matching process is also called "Place Recognition" in a VSLAM system. However, the robustness of monocular camera localization has not been solved sufficiently yet, because of its weakness in overcoming perspective changing and illumination changing for data association.

Regarding the mature localization algorithm,[6] to localize globally, it is difficult for the robot to recognize a visited place by exploiting the LiDAR feedback only, and such a problem is more obvious in the environment with geometric symmetry. Although a higher accuracy can be accomplished by the LiDAR sensor for robot localization, the feedback information of the LiDAR sensor always has a reduced dimension.[7] In most cases, due to the rich feedback information, visual-based localization owns a better performance than the LiDAR-based methods in terms of loop-closure detection. However, the visual sensor suffers from illumination changing, perspective changing, as well as the textureless situations,[8] those situations are quite common in practical.

In this paper, a sensor-fusion mechanism with visual information and LiDAR scan is designed, and we present a keyframe-based localization module built upon a mature particle filter (PF)[14] based localization algorithm. Also, a relocalization trigger mechanism based on motion continuity hypothesis is designed to address the kidnapped problem. For real-time performance, a sparse scan-to-map (Scan2Map)[1] score calculation approach is introduced to speed up the computation, as well as a proposed verification method based on the continuous motion assumption. The main contribution of this research is proposing a sensor-fusion mechanism for robot global localization, which is able to speed up the localization convergence, and a trigger mechanism is also introduced for kidnapped problem and relocalization verification. Our system is able to relocalize the robot after kidnapping, and supervise the current localization status online to re-initialize the tracking process properly.

The rest of the paper is organized as follows. In Section 2, we review the related studies about vision-based localization and PF-based localization. In Section 3, detailed introduction to the developed system, including scene representation, keyframe-based localization, and relocalization trigger mechanism, is presented. The evaluation results with our mobile robot platform are shown in Section 4, as well as the comparative experiments. Finally, Section 5 presents the conclusion and future work.

2. Related Studies

In computer vision area, IBL is always regarded as "image retrieval." If the similar existing frames (visited places) to the current observation can be found, the current pose of robot can be localized according to those frames. Two categories of existing localization methods can be found, which are divided according to the data association approach: image-to-image (2D–2D) approaches and image-to-map (2D–3D) approaches.[9]

For 2D–2D approaches, the correspondence of the frames is crucial. However, this type of localization can only work at places that have been mapped before. In 2D–3D approaches, the robot pose is able to be determined in different perspectives when the data associations can be built, because the reconstruction of environment by image triangulation has been established. However, the data association is hard to be established in many practical situations.

Ben et al.[2] present an efficient encoding of all the keyframes, which is achieved by exploiting random ferns[10] and matching image to image without any 3D reconstruction. Also, for robust image recognition, a keyframe sampling strategy by the minimum distance in pose space is applied. In ref.,[11] an accurate and robust direct 2D–3D matching algorithm is proposed. This work is based on associating 3D points with a visual vocabulary obtained from clustering feature descriptors. Also, FAB-MAP[12] is another work using the image representation of visual bag-of-words and modeling the dependencies between different visual words. The global descriptor is also utilized in the related studies; it is treated as a scene representation for image matching. Singh and Kosecka[13] proposed a

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

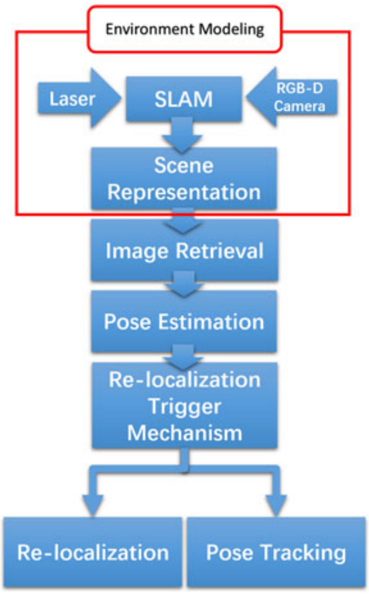

Fig. 1. Approach framework. After environment modeling, our proposed localization algorithm is conducted with the built scene representation.

method that extracts the GIST descriptor from the omnidirectional images for loop-closure detection in city environment. A GIST-based search space reduction (GSSR) system is introduced in ref.[9] to speed up the searching, where both context and global visual descriptor are used. At the same time, a blob visual descriptor (SIFT) is used for pose refining with a structure from motion (SFM) approach. However, the drawback of IBL has not been solved for the nature of monocular cameras.

Compared with the IBL, PF-based localization is a popular algorithm in the LiDAR-based localization area. It has the ability of converging multi-distribution position hypothesis globally with the capability of addressing the global localization problems theoretically. LiDAR-based AMCL[14] (Adaptive Monte Carlo Localization) is a mature localization system based on PF. However, the technique shows shortcomings in pose convergence speed without a known initial pose and solves the kidnapped problem. With respect to sensor information, those shortcomings are resulted from the low-dimension information of LiDAR scan. Referring to the LiDAR-based AMCL, the study[15] applies the principal component analysis (PCA) to GIST descriptors before weighting the observation likelihood in PF, which can be seen as a vision-based PF algorithm.

Regarding the existing localization algorithms, almost no localization algorithm is able to maintain tracking all the time. However, a continuous estimation is required for the autonomous system, and recovering from tracking failure is critical for practical autonomous robots. Referring to the earlier related work,[16–18] a sensor-fusion mechanism of visual information and LiDAR scan is introduced in this paper, and a keyframe-based localization module integrated into a popular PF-based localization algorithm is presented. We try to improve the robustness of the existing localization system, and re-initialize the tracking process properly by global relocalization based on our proposed sensor-fusion mechanism.

With respect to the visual data correspondence, a 2D–2D matching is applied in our system for image retrieval, and a localization optimization is designed through the local image descriptor, which is similar to the existing mature VSLAM systems.[3,4] Furthermore, compared with the existing algorithms, a keyframe clustering is introduced to improve the place recognition performance.

In addition, comparing with the existing sensor-fusion-based localization systems,[19–21] no GPS information is introduced in our system, and a series-type sensor-fusion mechanism is designed for solving the kidnapped robot problem. To our knowledge, refs.[32] and[33] are the most related studies to our work, where the sensor fusion is achieved by considering multiple observation in weighting function.

3. Framework

In this paper, a localization system of mobile robot by an LiDAR-vision sensor-fusion mechanism is put forward. The framework of our global localization system is shown in Fig. 1.

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

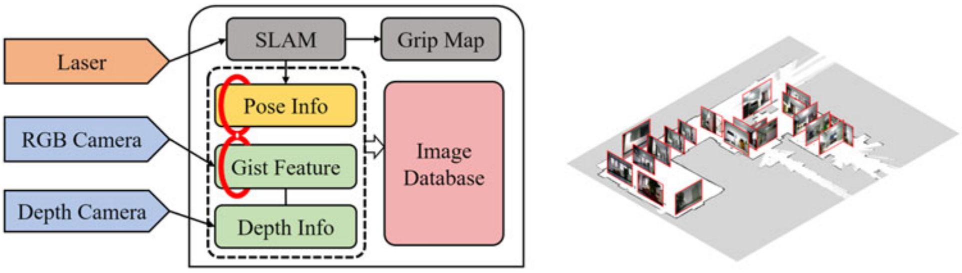

Fig. 2. Scene representation, where both cost map and keyframe set are provided.

At the first step, a hybrid environment representation is built in the mapping stage. For map building, the grid occupancy map is exploited to model the environment geometrically by LiDAR sensor. Also, a set of keyframes (involving images and poses) and their GIST descriptors are stored as a frame library using the information from the vision sensor, which represents the environment in appearance. The contribution of our vision-based localization is, for similarity calculation, a keyframe clustering method is introduced. We only consider the top-k keyframes ("k" is given manually) for similarity measurement in the keyframe library, and the GIST distance between the current observation and the keyframes is calculated for clustering.

In the second step, a PnP pose estimation by data association of ORB descriptors is performed to refine the relative pose between the query frame and the candidate keyframes. In the next stage, a localization evaluation based on our sensor-fusion mechanism is applied to further determining the correctness of the global relocalization, where an effective LiDAR-based evaluation method and a pose-update judgement is introduced. After global localization, the robot is supposed to maintain tracking using the LiDAR-based localization until the relocalization mechanism is triggered.

To summary the contribution of our system, a sensor-fusion mechanism in mapping and relocalization process is proposed for mobile robot pose estimation, whose performance is improved to handle the global localization problem. Also, to optimize the real-time performance and relocalization robustness, a keyframe clustering method and a relocalization trigger mechanism are designed.

4. Map Building

As mentioned in Section 3, in the environment modeling stage, the environment is modeled to build a hybrid occupancy map with the LiDAR feedback and camera feedback. By reference to the existing LiDAR SLAM methods, such as GMapping,[23] Karto SLAM[24] and Cartographer,[25] the occupancy map is grided into many two-dimensional square cells, and a occupancy confidence is associated in each grid. For grid-map building, the algorithm mentioned in ref.[23] is used in our system.

For introducing the visual-based localization into LiDAR method, a set of keyframes carrying the appearance information and depth images of the environment are captured by an RGB-D camera; this allows the retrieval of pose determination for tracking initialization as well as pose refining. A selection method for keyframes is designed in terms of the magnitude of robot movement, which is designed for implementing valid feature matching because the considered frames are almost identical within a short duration, and the size of the stored frame library is limited. Also, changes of the landscape observation in the environment will be caused by the changes of the robot pose.

Secondly, the observed objects may be dynamic as time goes on even if the robot is stationary. Therefore, we collected visual keyframes KF_i according to the odometer interval Δx_i and time interval Δt_i. Finally, a keyframe-based representation can be established based on the occupancy map, as shown in Fig. 2. In addition, the local visual descriptor is used for pose refining in the qualified keyframes after filtering through global descriptor, and this method will be discussed in the following section.

5. Visual-based Localization

After the map building, a hybrid environment representation is obtained. In this section, a localization method based on that hybrid map is introduced. First of all, the GIST descriptor for image matching is introduced. Then, we detail the way of image similarity calculation. After the image matching, the idea of refining the rough pose from image matching is shown.

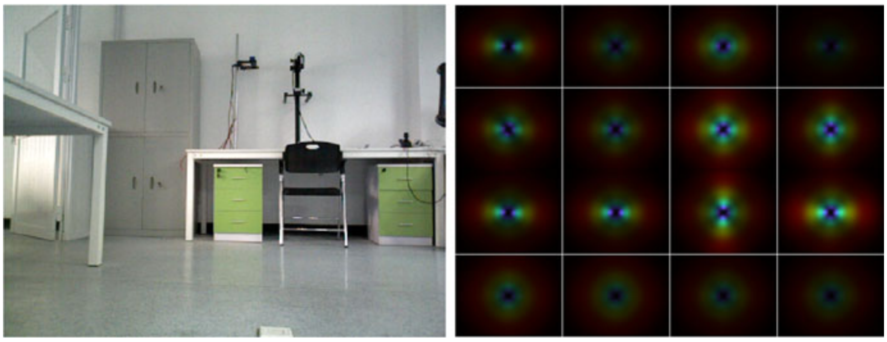

Fig. 3. GIST descriptor extraction: the left figure is the original image and the right figure is the correspondent GIST descriptor shown visually.

5.1. Visual global descriptor

GIST descriptor is a low-dimensional descriptor of an image. To extract a GIST descriptor, the structural information of the image is encoded by dividing the image into multiple blocks, and a description of the image can be obtained.[26] An example of GIST descriptor extraction is illustrated in Fig. 3, where the left figure is the original image and the right figure is the correspondent GIST descriptor shown visually.

In our localization system, the GIST descriptor is used for a fast image retrieval of coarse pose determination. Different from the scene classification processing steps, the input image is taken as a whole feature which describes the image of the statistical or semantic low-dimensional features. The motivation of this method is improving the robustness of place recognition. For the practical vision feedback is easily doped with random noise, and the noise can have a crucial influence on local processing. However, the global image descriptor can reduce this effect by averaging.[27]

Regarding the context-based approach, by identifying a global representation instead of a small set of objects in the image, it is unnecessary to deal with the change in noise and independent regions, which addresses the problem of segmentation and recognition classification.[26]

For each keyframe written as KF_i, the keyframe image is partitioned into a 4 by 4 tiles. Each of them is represented by the average of filters at [8,8,4] orientations per scale in the RGB color channels. Finally, a GIST descriptor g_i with a dimension of 960 (344(8+8+4) = 960) can be obtained for each keyframe.

In addition, there have been many existing studies introducing GIST descriptor in place recognition and loop-closure detection,[15] whose feasibility has been verified.

5.2. Similarity measurement

After the extraction of GIST descriptor for each keyframe, a comparison of the query image and keyframes in the frame library is run, and the similarity between two GIST vectors will be measured.

Let d_rs denotes the distance between two images respective GIST descriptor (the images' GIST are represented as $\mathbf{g}_r$ and $\mathbf{g}_s$), then the Minkowski distance, which is a metric defined on Euclidean space with ordered p, can be calculated as

$$d_{\mathrm{rs}}^{p} = \left(\sum_{n=1}^{k} \left| \mathbf{g}_{r(n)} - \mathbf{g}_{s(n)} \right|^{p} \right)^{\frac{1}{p}} \tag{1}$$

In our approach, we compute the L2-norm distances of GIST descriptor. If the distance d_{rs} is similar to the top one, then the keyframe can be treated as a potential referenced frame.

A correspondent pose of a potential referenced frame can provide a rough estimated pose of the robot in the built environment representation. However, because of the interval of keyframe sampling and the difference from the camera perspective, this pose estimation is not precious enough. To refine the rough estimated pose, we put forward a local image descriptor-based matching method, which will be discussed below in detail. Also, to improve the place recognition performance, we cluster the candidate frames after image retrieval, which will be discussed in Section 6.

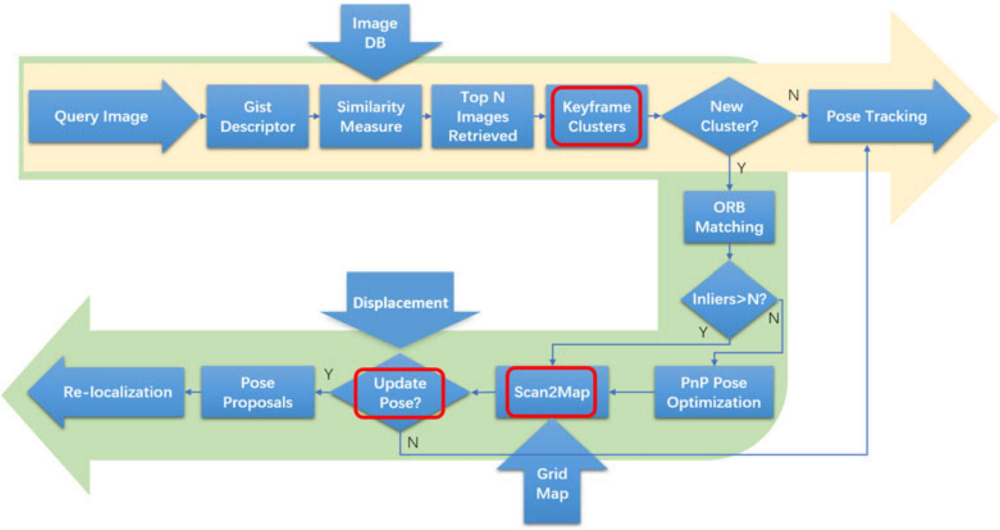

Fig. 4. Pipeline integration of our proposed relocalization trigger mechanism. Three strategies are designed to verify the localization status as is circled by red box.

5.3. *Localization refining*

After obtaining the rough pose estimation, to further improve the localization accuracy, only when more than 12 matching inliers via ORB descriptor are established in the proposed frames after RANSAC, the image will be registered and its pose can be refined by using PnP (n-point perspective).[28] ORB descriptor is a fast and robust local image feature, which is first presented by Ethan Rublee et al.[29] Since it is fast to extract and match, and invariant to viewpoint to some extent, it has a good performance in visual tracking and image matching. The threshold of 12 inliers is used in the reference.[30] This threshold also works well in the experiments and evaluations with our system.

Considering the detail of pose refining, only when the size of inlier correspondences after RANSAC is higher than 12, the potential keyframe can be used for further image-based pose refining. Otherwise, the pose information bound to the keyframe is considered as the query image rough pose, which is a non-optimal but feasible option, and it will be refined during the LiDAR-based localization. In the image-based refining stage, PnP method with RANSAC is exploited to estimate the robot pose. Firstly, in the pre-processing map building stage, we transform the local 3D points in the image coordinate to the global coordinate through the RGB image and depth image. Then, given the points position in the image space and the global coordinate, the transformation pose between the current observation and the proposed keyframe can be calculated.

6. Re-localization Trigger Mechanism

It is known that maintaining tracking all the time is almost impossible resulting from the noise of sensors, as well as the featurelessness of the environment, and the uncertainty of localization will increase during the whole tracking process. Therefore, our approach integrates a relocalization trigger mechanism to make our system more robust in some challenging situations, and re-initializes the tracking process properly to limit the increasing of tracking uncertainty. In our system, when a large displacement occurs in the mobile robot, or the image similarity matching occurs with a great mistake, the relocalization mechanism is supposed to be triggered, and a global relocalization will be executed to re-initialize the tracking.

As illustrated in Fig. 4, three strategies are involved in the whole algorithm flow (red boxes), which are designed based on the motion continuity hypothesis and our proposed sensor-fusion mechanism.

In the first strategy, the current image retrieval results $\boldsymbol{R}_t$ are leveraged to predict the corresponding situation of the next image retrieval results $\boldsymbol{R}_{t+1}$.

With the proposed distribution provided by image retrieval and keyframe clustering, the second strategy takes the LiDAR data into consideration to further assessing the likelihood that the robot is more likely to be. Also, a sparse Scan2Map calculation is introduced for real-time performance.

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

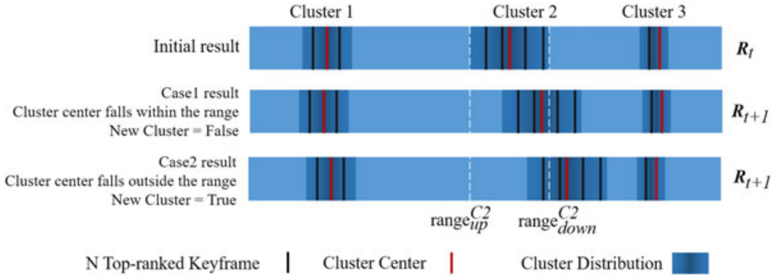

Fig. 5. Keyframe clustering, where horizontal axis represents the indexes of images.

In the third strategy, we check the relocalization results based on the confidence of visual sensor observation and the transformation distance of the relocalization result, which is performed in the "Pose Update" judgement.

6.1. *Keyframe clustering*

After computing the distance between GIST descriptors by the similarity measurement, the n top-ranked potential keyframes can be obtained (n is set manually). Then, we run the k-means clustering method to cluster the candidate keyframes, while the centers of cluster are replaced with the candidate frames whose score is the highest one. Also, the range of the cluster ${}_{t}^{Ci}$ is determined by the number of candidate keyframes KF falling on the cluster; it is denoted by $KF_t^{fall\_Ci(k)}$, as shown in Fig. 5. As for the number of cluster, it is given initially and incrementally in the localization process:

$$range_t^{Ci} = \alpha \cdot \omega_{v_{max}} \cdot \sum_{k=1}^{n} \left\{ p\left(KF_t^{fall\_Ci(k)} \right) \right\} \tag{2}$$

With the maximum speed $\alpha \cdot \omega_{v_{\max}}$ of the motion model, it is introduced to measure the maximum distance of each movement. With $\alpha \cdot \omega_{v_{\max}}$, the cluster range can be used to assess the possible distribution of the robot movement, which also indicates the motion continuity of the mobile robot. When a new observation is captured, the clustering method will be utilized, and a comparison between the new clustering result and the existing clustering result will be conducted. If the center of current k-means cluster does not fall within the range of the existing cluster, the new cluster result will be treated as a new cluster.

The advantage of keyframe clustering makes full use of the results of image retrieval for each loop-closure detection and avoids excessive trust of a false positive value which will cause repeated relocalization. Also, such a clustering can improve the robustness of localization, since more than one frames are considered in the image-based localization process.

To explain the localization process mathematically, the detail is shown in Formula. 3. As is known, the use of traditional image-based localization quickly concentrated the possible locations of the robot to several possibilities. In our approach, we integrate the vision and LiDAR information together and refine the robot pose with a maximum likelihood framework. The robot pose at time t is determined as follows:

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \left\{ p\left(z_t^l | x_t, m^l\right) \cdot p\left(x_t | \hat{x}_{t-1}, z_t^v, m^v\right) \cdot p\left(x_t | z_t^v, m^v\right) \right\}, \tag{3}$$

where x_t represents the status of robot at time t, m^l is the cost map, m^v is the vision map, and z_t^l and z_t^v are the observation of LiDAR and vision at time t. The probability $p\left(x_t | z_t^v, m^v\right)$ stands for the model of keyframe-based localization to predict the robot pose, and $p\left(x_t | \hat{x}_{t-1}, z_t^v, m^v\right)$ models the effect of keyframe clustering on continuous pose evaluation. The way of calculating the confidence of estimated pose according to the LiDAR feedback will be introduced in the next section.

6.2. *LiDAR-based score*

To calculate the confidence of estimated pose in terms of LiDAR information, a sparse counting is introduced to speed up the Scan2Map[1] score $p\left(z_t^l | x_t, m^l\right)$ calculation. Regrading the high-resolution LiDAR scan sensor, a large number of beams are gotten within a short time, which results to a great

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

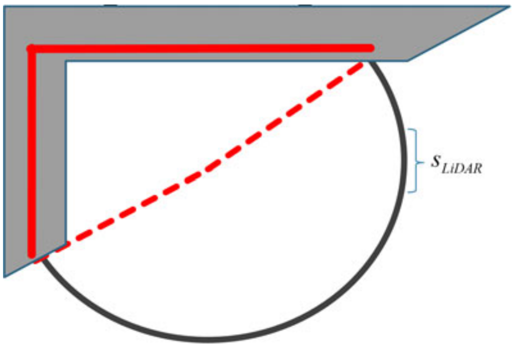

Fig. 6. Illustration of sparse Scan2Map score calculation. s_{LiDAR} is considered in the black arc for a check with interval, and a continuous checking is performed in the obstacle detected part (red).

time consumption for Scan2Map score calculation, because much information is taken into account. Also, for building a trigger mechanism supervising the current localization in real time, a time-saving score calculation is required. To speed up the score calculation, a sparse Scan2Map score calculation is introduced based on the continuous obstacle assumption.

Instead of checking all the beams, a given parameter s_{LiDAR} is introduced; we check the feedback with a s_{LiDAR} interval. However, the interval is not considered all the time. For the continuity assumption of a obstacle, we perform a dense checking (all the beams are checked) when a obstacle is detected by one of the beams, and stop to check with an interval when no more obstacle is detected, an illustration is shown in Fig. 6.

By using the sparse score calculation, less beams need to be checked, and the time consumption is saved especially for the high-resolution LiDAR sensor. Compared with the traditional method with a fixed interval, the Scan2Map score with our designed method is much higher for the obstacle detected scan, since more object-detected beams are considered. With a higher Scan2Map score for the obstacle detected scan, the signal-to-noise ratio can be improved, and the confidence of localization status checking is also improved.

To explain the sparse counting method mathematically, let m^l represent the LiDAR map and z_t^l=$\{z_t^1, z_t^2, , z_t^n\}$ is the LiDAR scan which contains n individual measurements (n corresponding LiDAR beams). The translation of the range reading from the LiDAR according to the proposed global robot pose is run. The grid cell hit by the LiDAR end-point z_t^k is denoted by $m^{\text{hit}(k)}$. If cell $m^{\text{hit}(k)}$ is being occupied, the occupancy score of the cell will be added into the hitting score. The likelihood of the sensor measurement can be represented by the final voting score:

$$\left(z_t^l | x_t, m_l\right) \propto \sum_{k=1}^{n} \left\{p\left(m^{hit(k)}\right), m^{hit(k)}\ is\ occupied\right\}. \tag{4}$$

At last, the sort of keyframes cluster is re-ordered according to the Scan2Map score. If the new cluster gets the highest score after the calculation of Scan2Map score, the re-order of cluster will be confirmed. It triggers the mechanism to make the corresponding position of the referenced frame be the potential pose estimation.

6.3. Pose update judgement

As is mentioned above, we supervise the current localization status online and try to relocalize the robot when it is needed. However, since the similarity of the environment (appearance and geometry), some relocalization results are false positive (the system is overconfidence with a false result from the vision sensor or the LiDAR scan), and the relocalization may lead to a local optima. Therefore, the verification of relocalization result is crucial for a practical global localization system.

To verify the relocalization result for online refining, a pose update judgement is designed based on continuous motion assumption and vision similarity d_{rs}. The basic idea of the judgement is: we calculate the distance (represented as d_{pose}) between the estimated pose of relocalization and the current pose (such a judgement is not performed when no current pose is given), and a threshold T_{pose} in terms of d_{pose} is set to decide accepting the relocalization result or not. Only when d_{pose} is less than T_{pose}, the relocalization result will be accepted. For the value of T_{pose}, instead of a fixed

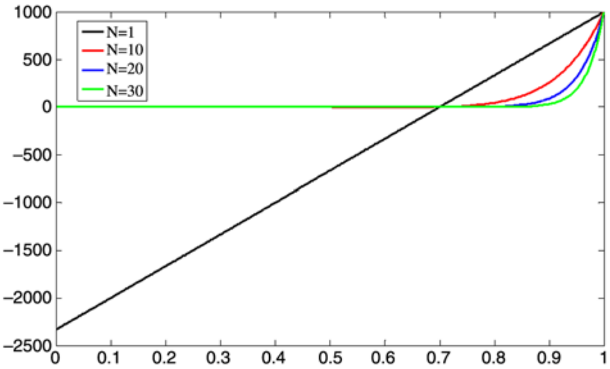

Fig. 7. An example of the curves of T_{pose} with different parameters, where $n = 1$, $n = 10$, $n = 20$ and $n = 30$, s_{map} is 1000 and l_{nrs} is set to be 0.7.

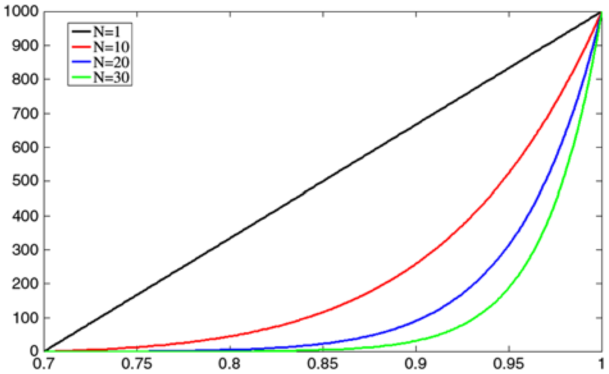

Fig. 8. Plot in the range of $[l_{\text{nrs}}, 1]$, where only the image matching results with a d_{nrs} larger than l_{nrs} are considered and the maximum of d_{nrs} is 1.

threshold mentioned in some related studies, a dynamic adaptive value is given according to the image matching similarity.

To set the value of T_{pose}, it is calculated according to d_{rs}. When the similarity between the current image and the selected image is high, confidence of the relocalization is also high and a bigger displacement is allowed. Therefore, a formula in terms of T_{pose} and d_{rs} is designed. Firstly, we normalize d_{rs} of all the matching results to be in the range of [0,1] (by maximum and minimum normalization), and a new similarity score d_{nrs} is calculated by $1 - d_{\text{rs}}$, making the image similarity score proportional to the confidence. The relationship between T_{pose} and d_{nrs} is calculated as follows:

$$T_{\text{pose}} = \frac{s_{\text{map}}}{1 - l_{\text{nrs}}} \bullet \left({d_{\text{nrs}}}^{n} - l_{\text{nrs}} \bullet {d_{\text{nrs}}}^{(n-1)}\right), \tag{5}$$

where s_{map} is the diagonal size of the cost map, l_{nrs} is the least d_{nrs} required (only the image matching results with a d_{nrs} larger than l_{nrs} are considered for verification), and n is a given hyper-parameter.

For a clear explanation, an example of the formulas with different parameters n is illustrated in Fig. 7. To focus on the range of $[l_{\text{nrs}}, 1]$, only that range is indicated in Fig. 8, where X-axis is d_{nrs} and Y-axis represents T_{pose}. When n equals 1, the relationship between d_{nrs} and T_{pose} is linear, the correlation coefficient is always positive and static. With the increasing of n, the X value of inflection point also increases, and the correlation coefficient becomes bigger with the increasing of d_{nrs}.

In such a curve, a big enough d_{nrs} is required for a large motion threshold T_{pose}, and only when d_{nrs} equals 1, will the motion having a magnitude of the diagonal map size be considered. After several tests on different datasets, n is set to be 20 in our system.

7. Experiments

Some evaluations and comparisons are performed in this section. A mobile robot platform in our lab is exploited for data collection, as well as an LiDAR sensor and an RGB-D camera. The set-up of experiments will be detailed in the first subsection. To evaluate the overall performance of our system,

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

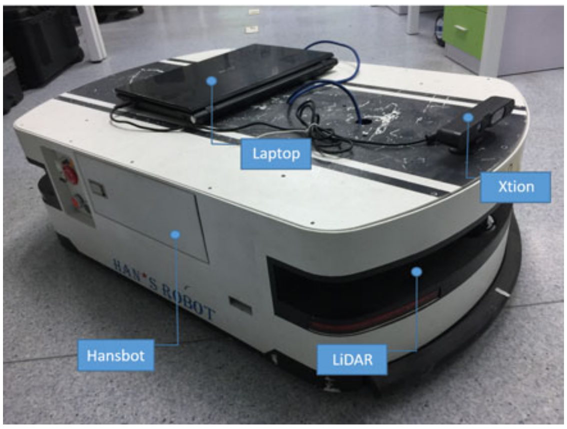

Fig. 9. Mobile robot platform used in our experiments.

as well as the keyframe clustering method for place recognition, some localization experiments and image retrieval results are shown in the second subsection. Also, to evaluate the localization precision and the capability in solving kidnapped problem quantitatively, some comparative experiments are carried out in the third subsection.

Instead of the brief introductions and evaluations in our previous work,[31] both quantitative and qualitative experiments and comparisons of the complete localization system are shown here, as well as the discussion. In addition, a demonstration of re-initialization of AMCL is provided in Supplementary Material.

7.1. Experiment set-up

We experiment on our mobile robot in the indoor environment and make assessment of the performance of the developed localization system. As can be seen in Fig. 9, a Hansbot AGV is equipped with an LiDAR (Hokuyo UST-10LX) and a Xtion Pro live camera (capturing the RGB image and depth image). The localization system is run on a computer with a configuration of Intel Core i5 (2.30 GHZ) CPU and 12 GB memory. It is remarkable that our localization system is run in real time.

Some datasets are collected to verify the feasibility and evaluate the robustness and precision of our sensor-fusion mechanism. The image sequence is recorded for keyframe building and image matching. In addition, the LiDAR scan is also recorded for cost map building and Scan2Map score calculation.

7.2. Evaluations

In this evaluation, Hansbot is randomly placed in different positions. Then, the robot tries to localize itself by using our localization system without any prior knowledge after mapping. As soon as the Hansbot achieves converged localization, we teleport it to other localization, for testing its ability on recovering from kidnapping.

In the preprocessing stage, a mapping algorithm which is developed based on[22] is executed with the Hansbot platform in our office room, and the built cost map is shown in Fig. 10. After the mapping process, a hybrid map of grid cost map and keyframes information is built, where 402 keyframes are established.

In aspect of image matching, to detail the relocalization process, given a new visual feedback, the Hansbot is supposed to relocalize itself in the global map. Then another query image contributes to find a more accurate pose by the local feature descriptors. As shown in Fig. 11, the mobile robot still considers itself in the previous position although it has got kidnapped, and the new visual observation after kidnapping triggers the relocalization mechanism and the robot is relocalized by our developed localization system.

An example of global relocalization is illustrated here. Figure 12 illustrates that after the robot is kidnapped, it can be relocalized (referred to re-initialization for AMCL). Through the localization trigger mechanism, our approach outputs an estimated pose for AMCL initialization after observing and verification. The feasibility of the relocalization results is analysed here, where both heading

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

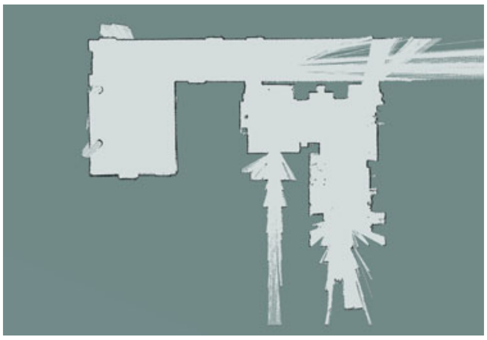

Fig. 10. Cost map of our office room. It is built by using GMapping software package.

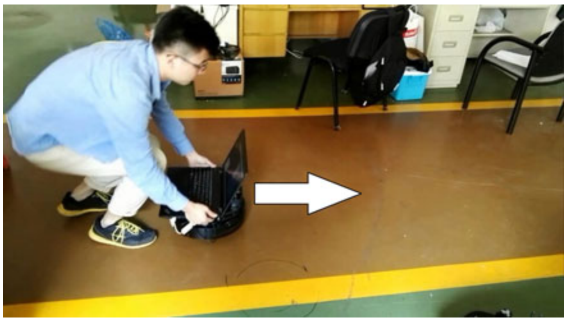

Fig. 11. Kidnapping manually. The robot is in the current position before kidnapping, and it is moved manually along the arrow.

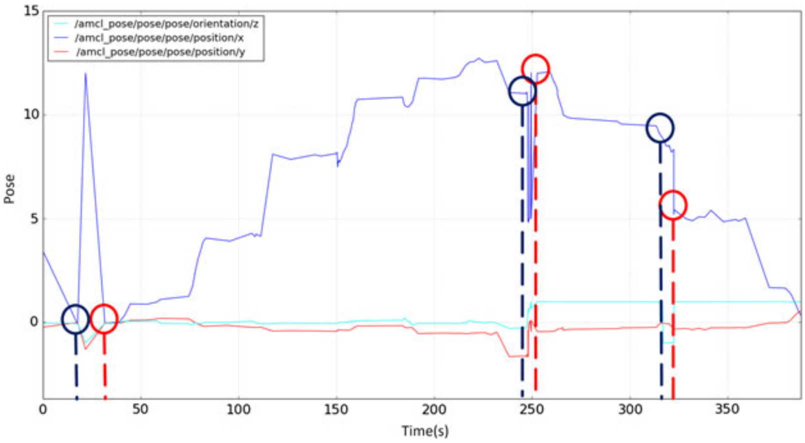

Fig. 12. Plot of AMCL estimated pose, where the deep blue lines are the moments of kidnapping while the red lines indicate the moments of relocalization. Both the pose (orientation: *z*) and position (position: *x*, *y*) are drawn.

and position are drawn. Three kidnappings are indicated by the deep blue dotted lines, and the correspondent relocalization results are indicated by the red dotted lines. After relocalization, the curves become steady, which verifies the relocalization results qualitatively.

The detail of image matching result is shown. Figure 13 demonstrates the top nine candidate frames that are similar to the given image. In the left figure, the selected results of image retrieval are shown, even there are dynamic objects in the scene and some objects are removed. Because the GIST global descriptor is invariant to the illumination changing, local objects change to some extent. Meanwhile, other keyframes also have a similar appearance. Such a result is the result of introducing keyframes clustering into the image matching, which gives more valuable matching results

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

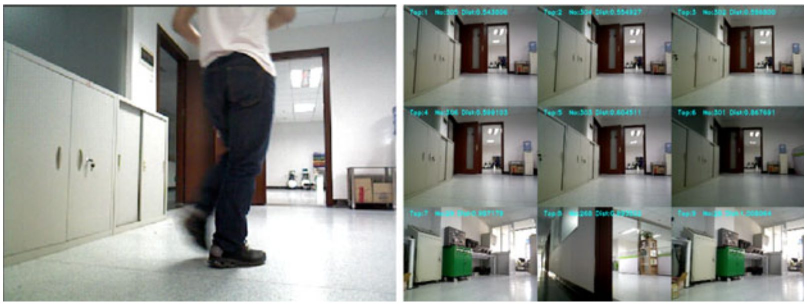

Fig. 13. Result of image matching. The left figure is the current observation and the right figure is the result. The top six matching results are correct, even there is a new object in the query image.

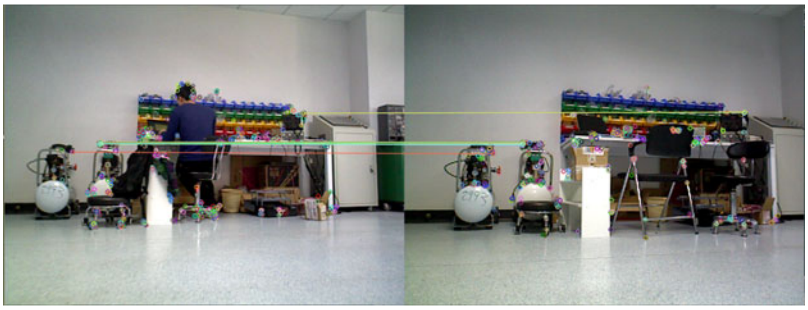

Fig. 14. Local descriptor matching with dynamic objects, where more than 12 inlier correspondences are detected and the frame is recognized as the referenced frame.

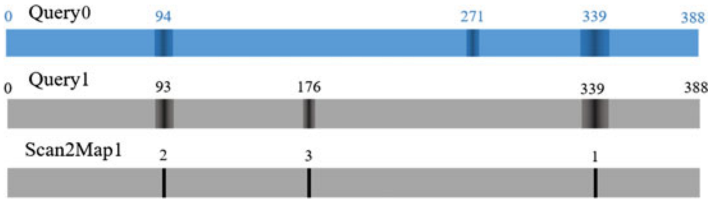

Fig. 15. Results of keyframe clustering and Scan2Map score calculation, where the first two rows are the result of two continuous query frames (query0 and query1). The third row is the correspondent result of LiDAR-based scoring, where the number is the rank of score.

for improving the robustness of place recognition, since building loop closure by a single place recognition result is fragile. Also, an example of blob descriptor matching is shown in Fig. 14.

Regarding keyframe clustering, according to the optimized performance during testing, k is set to be 3 initially, which is the parameter of k-means clustering of the top ranking keyframes. Figure 15 indicates the results of keyframe clustering, including the cluster centers (the number indicates the center position of cluster) and the range of each cluster (the gradient rectangle indicates the range area of the cluster), as well as the ranking of Scan2Map score. The experimental result is shown in Fig. 15, those sporadic unwanted match results are abandoned, while the No.339 frame is proposed. The clustering result of query image demonstrates that the observed image has a high possibility to fall in the cluster around image No.339 and No.93, and these hypothesis are verified by the Scan2Map score calculation and pose update judgement.

The time consumption of the experiments on two collected datasets is recorded. According to the recorded results, the average time consumptions without pose refining are 0.0083s and 0.0027s, and the average time consumptions of pose refining are 0.851s and 0.363s. As for the average working rate in our experiment, whose numbers of considered frames are 402 and 121, the average framerate are 34.33 Hz and 19.4 6Hz. Compared with the state-of-the-art in VSLAM area,[3] our system has a similar real-time performance, which is able to meet the requirement of online controlling in most mobile robot applications.

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

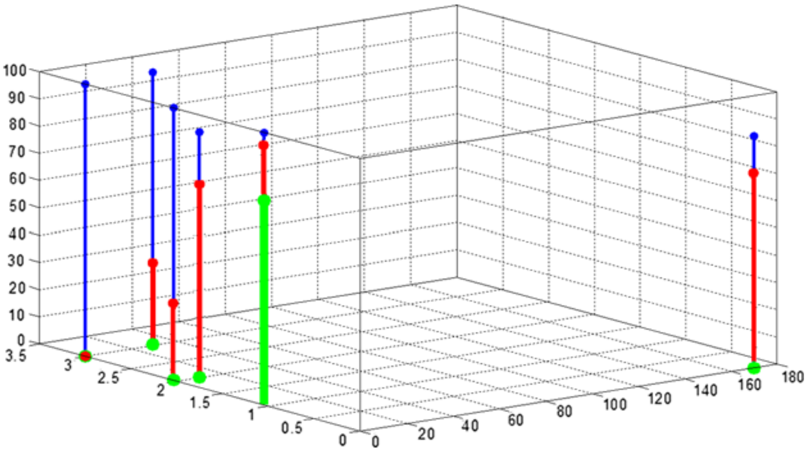

Fig. 16. The performance of three methods. Six kidnappings are performed and drawn as six bar clusters, where the Y-axis is the norm of translation, the X-axis is the norm of orientation (degree), and the Z-axis is the true positive percentage. The blue bars are the results of our system, the red bars are the results of fusion AMCL, and the green bars are the results of original AMCL.

To show the feasibility of our system, a demonstration working with AMCL algorithm can be seen in the attached video. For kidnapping manually (Fig. 11), a Turtlebot mobile platform is used instead of Hansbot, and the Xtion sensor is fixed on the Turtlebot as well as a laptop. A rough LiDAR scan is simulated by the Xtion feedback in that demonstration.

7.3. Comparisons

To show the superiority of our system in solving kidnapped problem, a comparison with the LiDAR-based AMCL is performed, which is written as “original AMCL” here. In “original AMCL,” the “random particle generation” is carried out to solve the kidnapped problem, whose weighted function is related to LiDAR observation only. Also, the sensor-fusion approach mentioned in ref.[32] are utilized for comparison, which is also quite similar to the fusion method proposed in ref.[33] This fusion approach is represented as “Fusion AMCL” here.

A dataset is collected in one of our office rooms using our mobile robot platform. Given a known environment representation (from the mapping process), we teleport the robot manually and change the robot pose, the L2 norms of kidnapping translation and orientation are recorded by measurement. Also, the correspondent LiDAR and vision sensor feedback are collected, which are the input of our localization systems.

The global localization of those three methods are judged according to the error between the estimated poses and ground truth poses. Only when the error is lower than a given threshold, the result is regarded as true positive. We run the dataset repeatedly and count the true positive percentage of these three methods to evaluate their capability in kidnapping solving; the result is indicated in Fig. 16.

As shown in Fig. 16, the successful percentage of our system is always the highest one, and the second one is the fusion AMCL. According to the experimental results, comparing the original AMCL and the fusion AMCL, we can say that the vision sensor information is able to improve the ability of global localization. Also, comparing the performance of the fusion AMCL and our system, the sensor-fusion mechanism in our system has a better performance.

To evaluate the precision of relocalization, we compare the trajectory of the original LiDAR-based AMCL and our system by using the same dataset (containing image sequence, wheel odometry, LiDAR scan, and a known environment representation) without any kidnapping (since kidnapping is hard to be solved by original AMCL). However, even there is no kidnapping in this dataset, we execute the relocalization of our system manually (instead of being required by the online checking) to evaluate the relocalization results, which are also demonstrated in Fig. 17.

In theory, the estimated trajectory of our system should be the same as that of original AMCL, since our system is built based on it. However, if the relocalization gives a false result, the trajectories of these two systems would be different. As shown in Fig. 17, the trajectories of two systems are quite similar, which means the online supervise method always makes a correct judgement of the current

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press

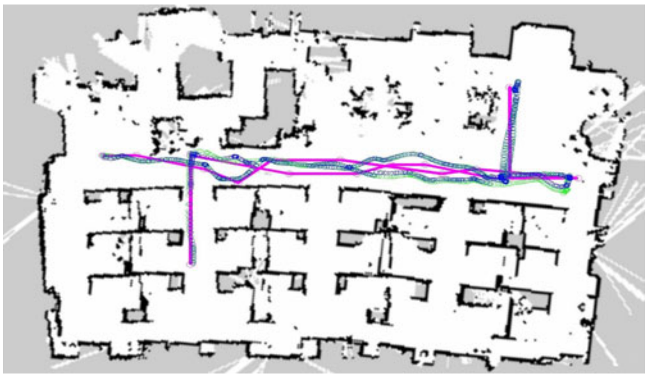

Fig. 17. Three trajectories are drawn in the figure. The green one is the trajectory estimated by the original AMCL, the blue one is estimated by our system, and the purple one is the connected line of all the relocalization results of our system (the hollow circles are the exact relocalization poses).

localization status, and the A-RMSE (Absolute Root Mean Square Error) of the trajectories between our system and original AMCL is 0.113 m.

In addition, with respect to the results of relocalization, all the results are around the correspondent AMCL estimated poses, and the A-RMSE between the relocalization poses and the correspondent poses is 0.162 m.

8. Conclusion

In this paper, a sensor-fusion mechanism for mobile robot global localization is proposed, which has the capability of addressing the global relocalization problem, where both LiDAR and camera sensors are intergraded. After building a hybrid environment representation with visual keyframes and occupancy map in the mapping processing, an image global descriptor matching is applied to search the referenced keyframes according to the current visual observation, and a refining pipeline is designed to improve the localization accuracy.

To supervise the localization status and detect the kidnapping in real time, we put forward a relocalization trigger mechanism involving keyframe clustering and Scan2Map score, and they are executed as the evaluation of the matching results. Also, it is used for tackling the global relocalization problem. Besides, a sparse Scan2Map calculation is designed to improve the real-time performance, and the motion continuous verification is introduced to overcome the false positive relocalization results. Some experiments under different environmental conditions are performed to evaluate the performance of our proposed approach. In addition, our system is able to give a precise localization estimation and has a better performance in solving the kidnapped problem than the popular localization algorithms.

According to the experimental results, we can draw the conclusion that the sensor-fusion mechanism proposed in our system is practical for mobile robot localization, and such a mechanism is able to improve the global localization and robustness.

As for the future work, one of the improvements of our system is introducing the advanced image matching method to our system for a more robust performance. Another future work for improving the performance is an automatical tuning of keyframe clustering parameters according to the data characteristics and reliability of keyframe clustering. Also, the evaluations of our system in the large-scale environment and long-term running should be experimented.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant number 61603103], [grant number 61673125]; the Natural Science Foundation of Guangdong of China [grant number 2016A030310293]; and the Major Scientific and Technological Special Project of Guangdong of China [grant number 2017B090910012].

Supplementary Material

To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574719000298.

References

1. T. Sebastian, B. Wolfram and F. Dieter, *Probabilistic Robotics* (MIT Press, Cambridge, MA, USA, 2005).
2. G. Ben, S. Jamie, C. Antonio and I. Shahram, "Real-time RGB-D camera relocalization via randomized ferns for keyframe encoding," *IEEE Trans. Visual. Comput. Graph.* **21**(5), 571–583 (2015).
3. M. Raúl and T. Juan D., "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robot.* **31**(5), 1147–1163 (2015).
4. E. Jakob, S. Thomas and C. Daniel, "LSD-SLAM: Large-scale Direct Monocular SLAM," *European Conference on Computer Vision*, Zürich, Switzerland (2014) pp. 834–849.
5. F. Christian, P. Matia and S. Davide, "SVO: Fast Semi-direct Monocular Visual Odometry," *Robotics and Automation (ICRA), 2014 IEEE International Conference,* Hong Kong, China (2014) pp. 15–22.
6. H. Wolfgang, K. Damon, R. Holger and A. Daniel, "Real-time loop closure in 2D LIDAR SLAM," *Robotics and Automation (ICRA), 2016 IEEE International Conference,* Stockholm, Sweden (2016) pp. 1271–1278.
7. K. Karel, V. Vojtech, K. Miroslav and P. Libor, "Comparison of shape matching techniques for place recognition," *Mobile Robots (ECMR), 2013 European Conference,* Barcelona, Spain (2013) pp. 107–112.
8. B. Guillaume and M. Remi, "Robust large scale monocular visual SLAM," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* Boston, MA, USA (2015) pp. 1638–1647.
9. A. Charbel, "Efficient Image-Based Localization Using Context," *University of Waterloo* (2015).
10. O. Mustafa, C. Michael, L. Vincent and F. Pascal, "Fast keypoint recognition using random ferns," *IEEE Trans. Pattern Anal. Machine Intelligence* **32**(3), 448–461 (2010).
11. S. Torsten, L. Bastian and K. Leif, "Fast image-based localization using direct 2d-to-3d matching," *Computer Vision (ICCV), 2011 IEEE International Conference,* Barcelona, Spain (2011) pp. 667–674.
12. C. Mark, N. Paul, L. Vincent and F. Pascal, "FAB-MAP: probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.* **27**(6), 647–665 (2008).
13. S. Gautam and K. J, "Visual loop closing using gist descriptors in manhattan world," *ICRA Omnidirectional Vision Workshop*, Anchorage, AK, USA (2010).
14. T. Sebastian, F. Dieter, B. Wolfram and D. Frank, "Robust Monte Carlo localization for mobile robots," *Artif. Intelligence* **128**(1–2), 99–141 (2001).
15. L. Yang and Z. Hong, "Visual loop closure detection with a compact image descriptor," *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference,* Vilamoura, Portugal (2012) pp. 1051–1056.
16. A. Charbel, A. Daniel C, F. Adel H and Z. John S, "Filtering 3D Keypoints Using GIST For Accurate Image-Based Localization," *Proceedings of the British Machine Vision Conference (BMVC)*, York, UK (2016) pp. 127.1–127.12.
17. M. José, C. Andrew and M. Walterio, "Enhancing 6D visual relocalisation with depth cameras," *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference*, Tokyo, Japan (2013) pp. 899–906.
18. K. Jungho, Y. Kuk-Jin and K. In So, "Bayesian filtering for keyframe-based visual SLAM," *Int. J. Robot. Res.* **34**(4–5), 99–141 (2015).
19. P. Cristiano and N. Urbano, "Fusing LIDAR, camera and semantic information: a context-based approach for pedestrian detection," *Int. J. Robot. Res.* **32**(3), 371–384 (2013).
20. C. Nicholas, M. Anush, M. James R and E. Ryan M, "Visual localization in fused image and laser range data," *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference*, San Francisco, CA, USA (2011) pp. 4378–4385.
21. M. Colin, F. Paul and B. Timothy D, "Towards lighting-invariant visual navigation: an appearance-based approach using scanning laser-rangefinders," *Robot. Autonom. Syst.* **61**(8), 836–852 (2013).
22. G. Giorgio, S. Cyrill and B. Wolfram, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robot.* **23**(1), 34–46 (2007).
23. "Gmapping ROS package," *https://github.com/ros-perception/slam_gmapping* (2018).
24. "Karto SLAM ROS package," *https://github.com/ros-perception/slam_karto/* (2017).
25. "Cartographer ROS package," *https://github.com/googlecartographer/cartographer* (2018).
26. T. Antonio, O, Aude, C. Monica S and H. John M, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search," *Psychol. Rev.* **113**(4), 766 (2006).
27. L. Jing, *Research on the Fast Scene Classification based on Gist of a Scene* (Jilin University, Jilin, China, 2013).
28. K. Laurent, S. Davide and S. Roland, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference*, CO, USA (2011) pp. 2969–2976.
29. R. Ethan, R. Vincent, K. Kurt and B. Gary, "ORB: An efficient alternative to SIFT or SURF," *Computer Vision (ICCV), 2011 IEEE International Conference*, Barcelona, Spain (2011) pp. 2564–2571.
30. L. Yunpeng, S. Noah and H. Daniel P, "Location recognition using prioritized feature matching," *European conference on computer vision* (2010) pp. 791–804.
31. Z. Su, X. Zhou, T. Cheng, H. Zhang, B. Xu and W. Chen, "Global localization of a mobile robot using lidar and visual features," *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference, Macau, China* (2017) pp. 2377–2383.
32. D. Perea, J. Hernandez, A. Morell and et. at, "MCL with sensor fusion based on a weighting mechanism versus a particle generation approach," *International IEEE Conference on Intelligent Transportation Systems, 2013*, The Hague, Netherlands (2013) pp. 166–171.
33. B. Yim, Y. Lee and J. Song and W. Chung, "Mobile Robot Localization Using Fusion of Object Recognition and Range Information," *IEEE International Conference on Robotics and Automation*, Rome, Italy (2007) pp. 3533–3538.

https://doi.org/10.1017/S0263574719000298 Published online by Cambridge University Press