Robust 2D Indoor Localization through Laser SLAM and Visual SLAM Fusion

Shao-Hung Chan
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
shaohung1113@gmail.com

Ping-Tsang Wu
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
alanwu6fssh@gmail.com

Li-Chen Fu
NTU Center for Artificial Intellengence & Advanced Robotics
Taipei, Taiwan
lichen@ntu.edu.tw

***Abstract*— A novel method for laser SLAM and visual SLAM fusion is introduced to provide robust localization in this paper. The architecture can be applied to fusing any two kinds of laser-based SLAM and monocular camera-based SLAM together. While laser-based SLAM and monocular camera-based SLAM have their own strengths and drawbacks, the integration of these two kinds of SLAM algorithm can promote the algorithmic effectiveness. Instead of using feature matching methods to achieve fusion procedure, trajectories matching is proposed with an attempt to achieve the generalization over all different kinds of SLAM algorithms, since localization is a natural function associated with any SLAM algorithm. It turns out that the hereby proposed approach is very lightweight during the run time, and the calculation can run in real-time without unnecessary computation waste. The experimental results show the localization error in terms of the real distance can be less than 5%. Furthermore, through the experiment the proposed system is able to improve the localization when the sensors are not very powerful.**

Keywords—laser-based SLAM, monocular camera-based SLAM, SLAM fusion, service robot

I. INTRODUCTION

In order to complete tasks safely and independently, a robot must realize its position in the environment. This problem can be viewed as SLAM (Simultaneous Localization And Mapping) problem in robotic. Since SLAM has already been developed for three decades, a bunch of SLAM algorithms and papers have been developed and written [1]-[25].

One of the most popular sensors adopted for developing SLAM algorithms is the laser rangefinder [4]. In [2], Rao-Blackwellized algorithm is proposed, in which the problem depletion arising in particle filter-based SLAM was solved. On the other hand, a scan matching method and a multi-resolution likelihood map generation algorithm were proposed for unmanned ground vehicle (UGV) localization in large-scale indoor environments, as shown in [4]. This method reduces the searching space during the matching process. When using laser rangefinder to estimate robot's position, it is possible to precisely determine the position depending on how accurate the laser rangefinder is. However, it is difficult to estimate the pose in environments such as a large space or long corridors without variety of observation, since the depth information obtained from the laser rangefinder does not change with time and is considered featureless. In other words, a robot is easy to get lost when the geometry of the environment is quite simple, such as a public space surrounded by a few long walls or circular walls. Another commonly used sensor for estimating the pose of the robot should involve camera, known as visual SLAM, which has grown interests in the research during the recent years [5]-[8]. Engel *et al*. proposed a monocular camera-based SLAM called LSD-SLAM [9], which is able to build large-scale semi-dense maps by using direct methods instead of bundle adjustment over features. In [11], Mur-Artal *et al*. proposed an indirect visual SLAM system, called ORB-SLAM, which divides the visual SLAM problems into three parallel sub-tasks: tracking, mapping, and optimizing. However, with the use of a single monocular camera, it is insufficient to solve the scaling problem. In addition, the camera is sensitive to illumination changes, and errors may occur during the robot localization process.

With the use of sensor fusion, one can always get more robust and better results [7],[13]-[15],[26]. Since laser rangefinder and monocular camera are the most common equipment for robots, the fusion of these two sensors will be useful. There are some literatures show the improved results with the use of sensor fusion [7],[13]-[15],[26]. However, most of these algorithms simply focus on feature extractions. Although feature extraction improved by sensor fusion betters the performance of localization, such algorithms are limited to specific SLAM algorithms. In our proposed work, we apply the fusion on the perspective of trajectories from SLAM. Since trajectory is essential for every SLAM algorithm, our proposed method can be applicable to any fusion of laser-based and monocular camera-based SLAM algorithm. Through our fusion method, the scaling problem in monocular camera-based SLAM can be complemented by accurate distance information provided by laser-based SLAM. On the other hand, with abundant of features in monocular camera-based SLAM, robot can localize itself in featureless environments. Therefore, in our proposed system, we keep both the properties of laser-based SLAM and monocular camera-based SLAM. In the experiment, we choose Rao-Blackwellized particle filter-based SLAM algorithm, known

2577-1655/18/$31.00 ©2018 IEEE
DOI 10.1109/SMC.2018.00221

as Gmapping in Robot Operating System (ROS), as laser-based SLAM algorithm and ORB-SLAM2 as monocular camera-based SLAM algorithm.

II. System Oveview

Rather than perform fusion on the extracted features, the fusion of two SLAM algorithms starts from collecting trajectories. Next, the process is triggered whenever the loop closure takes place in both laser-based SLAM and monocular camera-based SLAM or simply after both maps are built. Another novelty of our proposed system is that the computation methodology behind our work is similar to that in image processing. In short, we try to map the trajectories into images, where the information can be much easier to visualize. Moreover, the trajectories can be scaled, translated, rotated at the same step, which are introduced in Section III. As shown in Fig. **1**, the collected trajectories will be mapped into images, and the following process will be just like image processing. We introduce two filters, curvature filter and pyramid filter for extracting the information inside two maps. Finally, we find the essential matrix between two sets of filtered trajectories, and subsequently apply it to monocular camera-based SLAM to achieve better results.

III. Methodology

In this section, the details of every part of our system is described together with the pseudo codes.

A. Trajectory extraction

As shown in *Section II*, the robot builds maps and records the trajectories of both SLAM algorithms respectively. The fusion process will be triggered either when the loop closure occurs or the mapping procedure terminates. We extract and scale the coordinates of the trajectories corresponding to the same timestamps into the same boundary. In other words, those scaled positions are capable of being fit into two individual images with the same width w and height h, named as *trajectory images*. The translation and scaling process will generate transformation matrices, T^{L}_{traj} for the laser-based SLAM and T^{V}_{traj} for visual SLAM, with the following structure:

$$T_{traj} = \begin{bmatrix} \frac{1}{P_x} & 0 & 0 & -\frac{x_{\min}}{P_x} \\ 0 & \frac{1}{P_y} & 0 & -\frac{y_{\min}}{P_y} \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$P_x = \frac{x_{\max} - x_{\min}}{w}, P_y = \frac{y_{\max} - y_{\min}}{h}$$

Through the extraction process, the fusion problem can then be solved with several image-processing techniques, which can be considered as a novel approach.

B. Aligned Angle Analysis

The coordinate of the laser-based SLAM is obtained from the robot odometry, whereas the monocular camera-based SLAM defines the coordinate according to the first keyframe it generates. As a result, there exists a transformation matrix between these two system. For a 2-dimensional mapping problem, it is the in-plane rotation and translation that influences the misalignment of two coordinates.

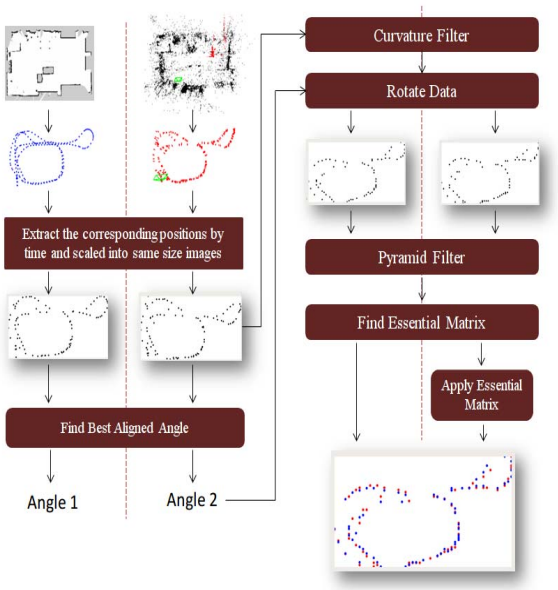


Fig. 1 The system architecture for the fusion process is shown above. We first take laser-based and monocular camera-based SLAM for demonstration. In the final result (right and down figure), blue dots denote the processed data from Gmapping (laser-based SLAM), red dots denote the processed data from ORB-SLAM2 (monocular camera-based SLAM).

This alignment problem is an optimization problem if defining the loss value as the Euclidean distance between two trajectory points at the same time stamp. We then minimize the loss value by rotating the images. After every point in a trajectory image is rotated, the maximum distance along x and y-coordinates will then be rescaled to fit the size of original trajectory images. Through the process, not only the orientation but also the translation relation of two trajectory coordinates can be calculated. The angle that leads to the optimal value is named as *aligned angle*. We will apply aligned angles to the rotational matrix as the following:

$$R_{AAA}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Algorithm 1: Aligned Angle Analysis

Input: Trajectory set of laser SLAM $P_L = \{p^L_1, p^L_2, \ldots, p^L_n\}$
Trajectory set of visual SLAM $P_V = \{p^V_1, p^V_2, \ldots, p^V_n\}$
Set R_L and R_V = Rotation angle ranges of P_L and P_V
Initialize $loss \rightarrow \infty$, $\alpha_L = 0, \alpha_V = 0$
For θ_L in R_L:
  For θ_V in R_V:
    $\tilde{P}_L$ = Rotate P_L by α_L
    $\tilde{P}_V$ = Rotate P_V by α_V
    $e = \sum_{i=1}^{n} \left\| \tilde{p}^L_i - \tilde{p}^V_i \right\|_2$
    If $e < loss$:
      $loss \leftarrow e$
      $\alpha_L \leftarrow \theta_L$
      $\alpha_V \leftarrow \theta_V$
Return α_L, α_V

C. Curvature Filter

As trajectory should be continuous, it is unlikely for robot to make a sharp turn. Supposed that the successive points inside a trajectory image at timestamp of *k-1, k, k+1* are p_{k-1}, p_k, p_{k+1}, forming a triangle. The point p_{k+1} will be filtered out if and only if the angle θ_k at p_k is specifically small. Given a threshold θ_{thres} and the coordinate points of p_{k-1}, p_k, p_{k+1}, the condition that keeps p_{k+1} can be verified through the law of cosine. That is:

$$\cos\theta_{thres} < \frac{\overline{p_{k-1}p_k}^2 + \overline{p_k p_{k+1}}^2 - \overline{p_{k-1}p_{k+1}}^2}{2\overline{p_{k-1}p_k}\,\overline{p_k p_{k+1}}}, \theta_{thres} \in [0,\pi] \quad (3)$$

where

$$\overline{p_i p_j} = \mathrm{dist}(p_i, p_j) = \|p_i - p_j\|_2$$
$$i, j \in \{k-1, k, k+1\} \wedge i \neq j$$

Inequality (3) can be modified as :

$$\frac{\overline{p_k p_{k+1}}}{\overline{p_{k-1}p_k}} > \left[\left(\frac{\overline{p_{k-1}p_{k+1}}}{\overline{p_k p_{k+1}}}\right)^2 + 1 - 2\frac{\overline{p_{k-1}p_{k+1}}}{\overline{p_k p_{k+1}}}\cos\theta_{thres}\right]^{\frac{1}{2}} \quad (4)$$

After curvature filtering, we again apply the translation and scaling processes discussed in *sub-section A* to fit the trajectory images. This additional process will bring about another two transformation matrices T^L_{curv} and T^V_{curv}. The detailed pseudocode is as the following:

Algorithm 2: Curvature Filter

Input: Trajectory set of laser SLAM $P_L = \{p^L_1, p^L_2, \ldots, p^L_n\}$
Trajectory set of visual SLAM $P_V = \{p^V_1, p^V_2, \ldots, p^V_n\}$
Sort P_L and P_V with the same timestamp
Set θ_k = filter angle threshold
Initialize storages Q_L and Q_V for processed data
For all $\{p^L_{k-1}, p^L_k, p^L_{k+1}\}$ and $\{p^V_{k-1}, p^V_k, p^V_{k+1}\}$:
  $D^L_1 = dist(p^L_{k-1}, p^L_k)$
  $D^L_2 = dist(p^L_k, p^L_{k+1}) / D^L_1$
  $D^L_3 = dist(p^L_{k+1}, p^L_{k-1}) / D^L_1$

  $D^V_1 = dist(p^V_{k-1}, p^V_k)$
  $D^V_2 = dist(p^V_k, p^V_{k+1}) / D^V_1$
  $D^V_3 = dist(p^V_{k+1}, p^V_{k-1}) / D^V_1$
  If $D^L_2 > \sqrt{D^L_3 + 1 - 2D^L_3\cos\theta_k}$ and $D^V_2 > \sqrt{D^V_3 + 1 - 2D^V_3\cos\theta_k}$:
    Append p^L_{k+1} , p^V_{k+1},to Q_L and Q_V respectively
Return Q_L, Q_V

D. Pyramid Filter

The idea of this procedure is inspired by the Pyramid technique in the field of image-processing [26], as trajectory map can be viewed as an image. Given a scaling ratio *r*, the coordinate position of certain trajectory points in the original image size will be resized by 1/*r* and rounded down to integers, which serves as a downsampling process. Then, the numbers will be multiplied by *r* in order to get back to the original scale, which, on the other hand, can be thought of as upsampling. Through this process, not only can the high frequency noises be excluded, but also smoothens the trajectory, as some updated positions merge among one another and therefore filter out the perturbations from measurements.

Algorithm 3: Pyramid Filter

Input: Curvature-filtered set of laser SLAM $Q_L = \{q^L_1, q^L_2, \ldots, q^L_n\}$
Curvature-filtered set of visual SLAM $Q_V = \{q^V_1, q^V_2, \ldots, q^V_n\}$
Set: scale ratio *r*
$\tilde{Q}_L = round(Q_L/r)\cdot r$, $\tilde{Q}_V = round(Q_V/r)\cdot r$
Return $\tilde{Q}_L, \tilde{Q}_V$

E. Finding Essential Matrix

With the purpose of increasing the fusion accuracy, the essential matrix [27] between two trajectory images is applied. As setting that *y* and *y'* are normalized image coordinates on monocular camera-based SLAM and laser-based SLAM respectively, the essential matrix E follows the formula:

$$(y'^T)Ey = 0 \quad (5)$$

in case y and y' correspond to the same timestamp. In addition, the rotation matrix R_{ess} and the translation vector t_{ess} between two SLAM coordinates can be found through SVD[28] or other methods[27]. The transformation matrix which will be applied in the final section is then constructed as:

$$T_{ess} = \begin{bmatrix} R_{ess} & t_{ess} \\ O & 1 \end{bmatrix} \quad (6)$$

F. Integrating

The final mapping process is to calculate a series of matrix multiplication combining results from the previous sections. Given trajectory points p^L and p^V from laser and visual SLAM, the *mapping matrix* $\boldsymbol{G_{LV}}$ between two coordinates can be obtained by:

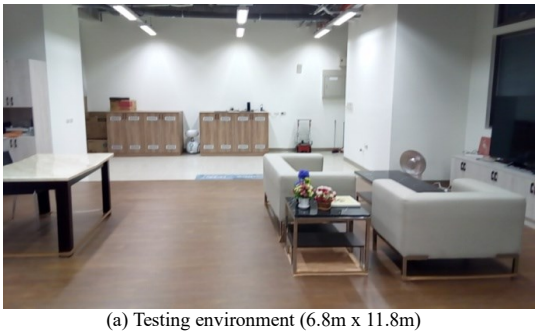
$$p^L = G_{LV}p^V = \left(T^L_{traj}\right)^{-1}\left(R^L_{AAA}\right)^{-1}\left(T^L_{traj}\right)^{-1}T_{ess}T^V_{traj}R^V_{AAA}T^V_{traj}p^V \quad (7)$$

where $R^L_{AAA} = R_{AAA}(\alpha_L)$ and $R^V_{AAA} = R_{AAA}(\alpha_V)$, recalling from *sub-section B*. Through the mapping matrix, the scale of monocular camera-based SLAM can be calibrated to that of the laser-based SLAM.

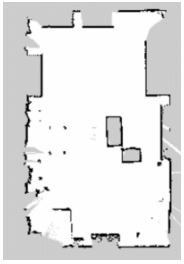
IV. Experimental Result

We implement Gmapping algorithm [2] and ORB-SLAM2 algorithm [29] as the choice of laser-based SLAM and visual SLAM. The map-building and fusion process are running under the laptop with Intel® Core™ i7-8550U (1.80 GHz x 8) CPU, without the usage of GPU. The whole system is combined under Robot Operating System. The experiment is conducted in an daily household indoor environment with the size 6.8m x 11.8m (Fig. 2 a). For the robot to localize based on the map built from Gmapping, we select Adaptive Monte Carlo Localization (AMCL) package implemented under ROS.

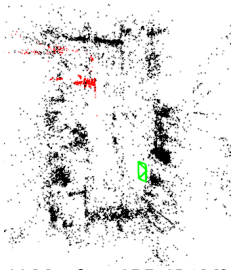
The first part of our experiment uses Pioneer 3-DX equipped with well-performed webcam and laser rangefinder (Fig.3(c)) to provides the quantitative localization result of the proposed architecture as the equipment mounted on the robot has higher quality. On the other hand, the second part of experiment uses Pepper as the robot platform (Fig. 4). As a social interaction robot, Pepper does not equip sensors as



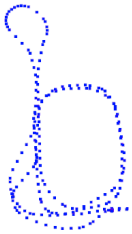
(a) Testing environment (6.8m x 11.8m)



(b) Map from Gmapping SLAM



(c) Map from ORB-SLAM2



(d) Trajectory of Gmapping SLAM

(e) Trajectory of ORB-SLAM2

Fig. 2 The photo of testing environment (a) and the maps built by Gmapping SLAM (b) and ORB-SLAM2 (c) respectively. Figure (d) and (e) are the trajectories extracted from both SLAM algorithms.

powerful as Hokuyo laser and webcam. Its laser range finder has only 45 laser beams within 180 degrees. Besides, due to the data transmission between Pepper and computer, the frame rate for visual image is merely 4fps.

A. Experiment using Pioneer 3-DX

The equipment installed on Pioneer 3-DX are a monocular camera with resolution of 640x480 and a Hokuyo URG-04LX-UG01 laser rangefinder with angular resolution 0.352 degrees. The map-building and fusion processes will be operated at the initial state as a preliminary so that we can evaluate the accuracy and the performance of the fusion. Without depth information, the scale of map from ORB-SLAM2 varies from the camera type, the trajectory while building the map, and the environment. We show that in our system, we don't need to pay effort to this problem as the scaling information has been included in mapping matrix $\boldsymbol{G_{LV}}$.

As the operation time becomes longer, the odometry will drift worse gradually, leaving more or less impact to AMCL. Furthermore, due to the fact that there exist places with similar geometry inside the indoor environment, AMCL may converge to the wrong position. Last but not least, AMCL may diverge to multiple positions if the dynamic obstacles present. Through the experiment, we show that our fusion architecture is capable of solving these kinds of problems.

(a) Pioneer 3-DX

(b) Monocular camera (resolution of 640x480)

(c) Hokuyo URG-04LX-UG01

Fig. 3 The equipment for first part of experiment

Fig. 4 The social interaction robot Pepper for the second part of our experiment. Only the forehead monocular camera (resolution of 640x480) and laser range finder in the bottom (45 laser detection beams) are used.

1) Verifying localization scaling accuracy

To verify the localization ability of the modified visual SLAM, six linear paths are selected randomly in the scene, with vertices labeled from A to F (Fig. 5). Given the coordinates of vertices, we will calculate the Euclidean distances of each path in the ORB-SLAM2 system with mapping matrix. As for the need of comparison between calculation and the real-world measurement, we apply relative error formulated by equation (9) with D_G and D_r representing the distance of ORB-SLAM2 calibrated with mapping matrix and the real world measurement. As shown in TABLE I, the relative errors among all paths are less than 5%, which has shown quite appealing performance of the mapping system.

$$relative\ error\ =\ \left|D_G - D_r\right| / D_r \tag{8}$$

Another impressive part of this experiment is that the distance ratio between ORB-SLAM2 with mapping matrix and the original one varies in different paths. In other words, the scaling ratios of ORB-SLAM2 are not the same between the x axis and y axis. Nevertheless, this problem can be ignored in our system.

TABLE I. Result of verifying localization scaling accuracy

Path	Real distance (m)	Distance in ORB-SLAM2 map (unknown unit)	Distance after applying G_{LV} to the position in ORB-SLAM2 map (m)	The distance ratio with and without applying G_{LV} on ORB-SLAM2	Relative error (%)
AB	1.53	0.1985	1.48	7.46	3.14
AD	2.92	0.3739	2.86	7.65	2.04
AE	6.71	0.8824	6.82	7.73	1.64
BF	4.19	0.5244	4.00	7.62	4.61
CD	2.63	0.5314	2.63	4.95	0.00
FD	4.00	0.5146	4.15	8.07	3.84

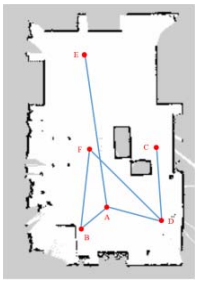
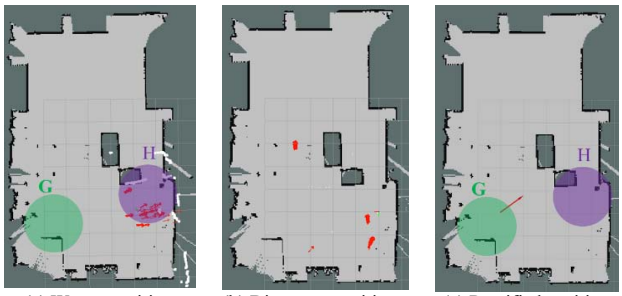


Fig. 5 The randomly selected paths (marked as blue lines) with vertices labeled from A to F (marked as red spots).



(a) Wrong position (b) Divergent positions (c) Rectified position

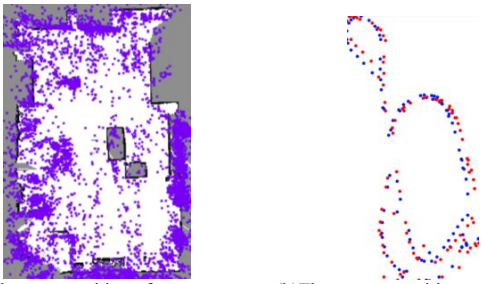
Fig. 6 Shows the experiment result of section *A. 2)* and *A. 3)*. (a) shows that AMCL converges to the wrong position with regions of similar geometry and (b) is the case that multiple positions exist when people moving. Both of the problems are solved through our system as (c) being the correct result.

2) Correcting wrong positions

Similar geometric regions in an environment may mislead the convergence of particles in AMCL. For example, the geometry of our indoor environment looks alike in region G and H (Fig. 6). As a consequence, even though the robot is placed at G, AMCL converges its location to H as Fig. 6(a). Through our system, the position of robot can be calibrated to the right position , as shown in Fig. 6(c).

3) Rectifying divergent positions

AMCL may diverges if dynamic obstacles are added. To be more specific, a robot may exist multiple positions under the condition that the surroundings become too crowded. During the experiment, several people are walking with robot being placed at the same position as *sub-section 2)*, causing divergent positions in AMCL as Fig. 6(b) shows. Nevertheless, with the assistance of the ORB-SLAM2 and mapping matrix G_{LV}, the position can be rectified to the correct position as Fig. 6(c).



(a) The superposition of two maps (b)The superposition of the extracted trajectories.

Fig. 7 The final result of fusion. (a) The visualization of maps fusion, which shows the fusion result can not only apply on the region of the trajectory but also for the whole map. (b) shows the superposition image of the extracted trajectories. Blue dots denote the trajectory extracted from Gmapping algorithm, while red dots denote the trajectory extracted from ORB-SLAM2 algorithm.

4) Final map fusion

Through visualizing the projected point cloud of ORB-SLAM2 with mapping matrix, it is obvious that this fusion system can be applied on not only the region of the trajectory but also the whole map. The final result of the map fusion is as Fig. 7, where (a) and (b) showing the superposition of the map and the trajectories respectively.

B. Experiment using Pepper (social interaction robot)

Regarded as an social companion robot, sensors mounted on Pepper is not as powerful as webcam or Hokuyo laser rangefinder. On one hand, there are merely 45 beams within 180 degrees (angular resolution of 4 degrees) inside its laser rangefinder. Thus, Pepper is highly to get lost when using AMCL algorithm as the only localization resource. On the other hand, due to the delay from image transmission, about 4 fps, and the limitation of monocular camera-based SLAM localization, Pepper may not obtain continuous localization information as from ORB-SLAM2, leading to unsafety in the indoor environment.

With the use of proposed fusion system, localization can be more robust. While AMCL localization provides continuous localization, the modified ORB-SLAM2 provides absolute location which helps AMCL to find the location back when the algorithm gets lost in the environment. Fig. 8 shows the trajectories extracted from Gmapping and ORB-SLAM2, it turns out the system does correct the localization from ORB-SLAM2 to the correct scale; moreover, the localization after fusion is better than the result from purely Gmapping SLAM.

V. Conclusion

In this paper, a novel approach of fusing laser-based and monocular camera-based SLAM to achieve robust indoor localization is proposed. Instead of using feature matching, trajectory matching method is applied, which therefore becomes general architecture for any kinds of SLAM fusion. Moreover, experiments show our fusion result is accurate on not only the matched trajectory, but also the whole map. What's more, two SLAM algorithms can complement drawbacks from each other given weak sensors. That is, even the laser rangefinder is sparse and the visual image has 0.2 second time delay, our system generates accurate localization that is robust enough for navigation.

VI. Acknowledgment

MOST (Ministry of Science and Technology of Taiwan) Joint Research Center for AI Technology and All Vista Healthcare under the MOST Grant (MOST 107-2634-F-002-018)

References

[1] Dellaert, F.; Kaess, M. Square root SAM: Simultaneous localization and mapping via square root information smoothing. Int. J. Robot. Res. 2006, 25, 1181–1203.

[2] Grisetti, G.; Stachniss, C.; Burgard, W. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, 18–22 April 2005; pp. 2432–2437.

[3] Gouveia, B.D.; Portugal, D.; Marques, L. Speeding up Rao-Blackwellized particle filter SLAM with a multithreaded architecture. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, USA, 14–18 September 2014; pp. 1583–1588.

[4] Tang, J.; Chen, Y.; Jaakkola, A.; Liu, J.; Hyyppä, J.; Hyyppä, H. NAVIS—An UGV indoor positioning system using laser scan matching for large-area real-time applications. Sensors 2014, 14, 11805–11824

[5] Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. IEEE Trans. Pattern Anal. 2007, 29, 1052–1067.

[6] Strasdat, H.; Montiel, J.; Davison, A.J. Scale drift-aware large scale monocular SLAM. In Proceedings of the Robotics: Science and Systems (RSS), Zaragoza, Spain, 27–30 June 2010.

[7] Ramos, F.T.; Fox, D.; Durrant-Whyte, H.F. CRF-matching: Conditional random fields for feature-based scan matching. In Proceedings of the Robotics: Science and Systems (RSS), Atlanta, GA, USA, 27–30 June 2007.

[8] G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR), 2007. 1, 2.2

[9] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: Large-scale direct ¨ monocular SLAM," in Proc. Eur. Conf. Comput. Vision, Zurich, Switzerland, Sep. 2014, pp. 834–849.

[10] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in Proc. IEEE Int. Conf. Robot. Autom., Hong Kong,Jun. 2014, pp. 15–22.

[11] R. Mur-Artal, J. Montiel, and J. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in Proc. IEEE Int. Conf. Comput. Vision, Barcelona, Spain, Nov. 2011, pp. 2564–2571.

[13] May, S.; Droeschel, D.; Holz, D.; Fuchs, S.; Malis, E.; Nüchter, A.; Hertzberg, J. Three-dimensional mapping with time-of-flight cameras. J. Field Robot. 2009, 26, 934–965.

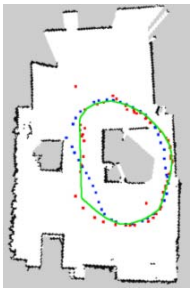


Fig. 8 The result for second part of experiment. The blue dots denote the trajectory extracted from Gmapping algorithm. The red dots denote the trajectory from the proposed algorithm. The green loop denotes the ground truth of the trajectory which is preset before the experiment.

[14] Zhang, X.; Rad, A.B.; Wong, Y.K. Sensor fusion of monocular cameras and laser rangefinders for line-based simultaneous localization and mapping (SLAM) tasks in autonomous mobile robots. Sensors 2012, 12, 429–452

[15] Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. Int. J. Robot. Res. 2012, 31, 647–663.

[16] Lee, D.; Kim, H.; Myung, H. GPU-based real-time RGB-D 3D SLAM. In Proceedings of the International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Daejeon, Korea, 26–29 November 2012; pp. 46–48.

[17] Lee, D.; Kim, H.; Myung, H. 2D image feature-based real-time RGB-D 3D SLAM. In Proceedings of the Robot Intelligence Technology and Applications (RiTA), Gwangju, Korea, 16–18 December 2012; pp. 485–492.

[18] Lee, D.; Myung, H. Solution to the SLAM problem in low dynamic environments using a pose graph and an RGB-D sensor. Sensors 2014, 14, 12467–12496.

[19] Segura, M.J.; Auat Cheein, F.A.; Toibero, J.M.; Mut, V.; Carelli, R. Ultra wide-band localization and SLAM: A comparative study for mobile robot navigation. Sensors 2011, 11, 2035–2055

[20] He, B.; Zhang, S.; Yan, T.; Zhang, T.; Liang, Y.; Zhang, H. A novel combined SLAM based on RBPF-SLAM and EIF-SLAM for mobile system sensing in a large scale environment. Sensors 2011, 11, 10197–10219

[21] Shi, Y.; Ji, S.; Shi, Z.; Duan, Y.; Shibasaki, R. GPS-supported visual SLAM with a rigorous sensor model for a panoramic camera in outdoor environments. Sensors 2012, 12, 119–136.

[22] Munguía, R.; Castillo-Toledo, B.; Grau, A. A robust approach for a filter-based monocular simultaneous localization and mapping (SLAM) system. Sensors 2013, 13, 8501–8522

[23] Guerra, E.; Munguia, R.; Grau, A. Monocular SLAM for autonomous robots with enhanced features initialization. Sensors 2014, 14, 6317–6337.

[24] Lee, S.M.; Jung, J.; Kim, S.; Kim, I.J.; Myung, H. DV-SLAM (dual-sensor-based vector-field SLAM) and observability analysis. IEEE Trans. Ind. Electron. 2014, 62, 1101–1112

[25] Jung, J.; Oh, T.; Myung, H. Magnetic field constraints and sequence-based matching for indoor pose graph SLAM. Robot Auton. Syst. 2015, 70, 92–105.

[26] E.H. Andelson and C.H. Anderson and J.R. Bergen and P.J. Burt and J.M. Ogden. "Pyramid methods in image processing". 1984.

[27] H. Christopher Longuet-Higgins (September 1981). "A computer algorithm for reconstructing a scene from two projections". Nature. 293 (5828): 133–135.

[28] Richard Hartley and Andrew Zisserman (2003). Multiple View Geometry in computer vision. Cambridge University Press.

[29] R. Mur-Artal, J. D. Tardos, "ORB-SLAM2: an Open-Source SLAM system for monocular stereo and RGB-D cameras", 2016.