Hybrid SLAM-based Exploration of a Mobile Robot for 3D Scenario Reconstruction and Autonomous Navigation

Thi Thoa Mac[1*], Chyi-Yeu Lin[2], Nguyen Gia Huan[1], Luong Duc Nhat[1], Pham Cong Hoang[1], Hoang hong Hai[1]

[1]School of Mechanical Engineering, Hanoi University of Science and Technology, No. 1 Dai Co Viet, 100000 Vietnam; thoa.macthi@hust.edu.vn, huan.ng141888@sis.hust.edu.vn, hoang.pc141805@sis.hust.edu.vn, nhat.ld143312@sis.hust.edu.vn, hai.hoanghong@hust.edu.vn

[2]Department of Mechanical Engineering, Taiwan Building Technology Center, Center for Cyber-Physical System, National Taiwan University of Science and Technology, 10607 Taipei, Taiwan, jerrylin@mail.ntust.edu.tw

[*]Corresponding author: Thi Thoa Mac

Abstract: SLAM can be categorized into two groups: laser-based SLAM and visual-based SLAM. They are used to identify surrounded objects of a robot. This paper proposes a combination of visual-based SLAM algorithm and laser-based SLAM. The purpose is to reduce effort but still provides the high quality 3D-reconstructed map. First, this paper presents visual-based SLAM and laser-based SLAM separately. Then, two techniques are integrated into one system. In addition, bi-direction RRT path planning algorithm is developed to create a feasible and optimal trajectory. A self-tuning Fuzzy-PID controller also is introduced for driving the robot to follow the trajectory precisely. The simulations and real experiments are conducted in order to illustrate the superiority of the proposed approach.*

Keywords: SLAM; mapping; 3D point cloud; sensor fusion; autonomous robot

1 Introduction

Simultaneous localization and mapping (SLAM) is one of the most important technique for localization and autonomous navigation of mobile robot [1]. The essential principle of SLAM is to provide information of the surrounding environment based on its sensor system and to construct the map of the working space while estimate the robot localization and orientation. Recently, LiDAR-SLAM (Light Detection and Ranging) and Visual-SLAM are two popular practical approaches to build maps in 2D and 3D for the intelligent autonomous applications [2].

LiDAR is preferred to use to construct a grid map and to detect the obstacles [3]. Extended Kalman filter (EKF) is implemented to obtain the position and orientation of the robot [4]. However, this approach is very difficult to apply in real nonlinear systems as it has accumulated errors which may cause to inaccurate positioning and mapping. In [5], 2D LiDAR scanner is used for in-row robot navigation in orchards. A Particle Filter (PF) with a laser beam model and Kalman Filter (KF) are implemented for localization and a line-detection algorithm, respectively. Self Adaptive Monte Carlo Localization (SA-MCL) is implemented in [6] for autonomous navigation with 2D and 3D LiDARs. The advantage of this method is solving the kidnapping sub-problems. Cartographer methodology is proposed by applying the laser loop closing to both sub-map and global map. As a consequence, the accumulative error is smaller. As LiDAR emits infrared light, the objects that do not reflect infrared light such as matte-black, glasses, degrade the performance of the laser-based SLAM packages. In addition, long corridors, square-shaped rooms and open wide areas where no obstacle information can be acquired make the laser-based SLAM algorithms non-operational.

Visual-based SLAM stirs up both scholar and commercial interests because of its effectiveness in the last decade. Compared to LiDAR-based SLAM, Visual-based SLAM is preferred as cameras have become much cheaper and also provide texture rich information about robot working environment. A survey of visual SLAM and Structure from Motion (SfM) in dynamic environments is introduced in [7]. This paper mentioned that Dynamic-SLAM is a robust visual SLAM. A study on 3D scenario reconstruction based on Growing Neural Gas (GNG) is investigated in [8]. The advantage of this method is accelerating the learning speed and reducing the noise from the capture system. In [9], a multi-level RANdom SAmple Consensus (RANSAC) approach is applied to segment and track moving objects. The problem of SLAM in a dynamic environment is studied in [10]. A Single Shot Detector (SSD) based on deep learning is constructed to detect dynamic objects. To improve the recall rate of detection, a proposed missed detection compensation algorithm is used. Then, the feature based visual SLAM system is produced using the feature points of dynamic objects to eliminate the pose estimation’s error. In [11], a fast Semi-direct monocular Visual Odometry (SVO) is implemented to integrate the feature point and direct tracking optical flow method. Other approaches such as DSO (Direct Sparse Odometry) [12], VINS-Mono (Monocular Visual-Inertial System) [13] are introduced to save computing resources in tracking and matching. The disadvantage is the insensitivity to features.

In our previous study, we developed a robust six Degree of freedom (Dof) SLAM algorithm using an RGB-D (Depth Sensor) graph-based approach [14]. The RGB-D camera-based SLAM of indoor environments is developed using plane features [15]. The STING-PE (Statistical Information Grid - Plane Extraction) and PAG-

PM (Plane Association Graph based Plane Matching) have been integrated. The camera pose is calculated based on the matched plane feature. In [16], a solution to an active SLAM is applied within an MPC (Model Predict control) framework. In addition, a sub-map joining method is implemented to archive the effectiveness of the proposed method and improve the computation time.

In [17], a deep CNN (Convolutional Neural Network) model is applied for terrain segmentation in wild environments. In similar approaches, RGB-D SLAMIDE (SLAM In Dynamic Environments) is investigated in [18-20]. The results are impressive by integrating SLAM framework with deep learning network. Many studies of SLAMIDE focused on the LiDAR SLAM and RGB-D SLAM as both information of the depth and surrounding environment are provided. In [21], a Learned Action SLAM, which combines path planning with SLAM is introduced. In this approach, heterogeneous robots are able to share their learnt knowledge through Learning Classifier Systems (LCS). A sensor fusion-based indoor exploration approach is introduced in [22] to simultaneously optimize the map quality and the exploration speed.

Unlike the existing approaches, in our work, a combination of visual-based SLAM algorithm and laser-based SLAM are proposed for autonomous navigation. In which, laser-based SLAM algorithm used a 3600 Laser Distance Sensor Rplidar-A1 and visual-based SLAM is implemented using a RGB-D camera, Intel RealSense. The combination of a 2D Occupancy Grid Map and 3D Point Cloud Map on Robot Operating System (ROS), is proposed to increase the accuracy. In addition, a RRT* (Rapidly Exploring Random Tree) path planning algorithm is also investigated to create a feasible and optimized trajectory for the mobile robot. A Self-tuning Fuzzy PID Controller also is proposed for driving the robot to track the trajectory accurately.

This paper is organized as follows. The visual-based SLAM, laser-based SLAM and integrated algorithms are briefly outlined in Section II. Section III presents the RRT* path planning. Section IV introduces Fuzzy-PID controller. Section V demonstrates simulations and experiment results for our research. Lastly, the conclusions and future works are given in section VI.

2 SLAM Implementation

In this section, we introduce LiDAR-based SLAM, Visual-based SLAM and the integrated approach. The aim focuses on the following three objectives and contributions: 1) developing a 3D-reconstructed mapped point cloud using LIiDAR sensor and RGB-D camera, 2) reducing effort and time of the point cloud data collection and registration process for ensuring construction quality and safety, and 3) providing high resolution registered RGB-mapped point cloud.

2.1 LiDAR-based SLAM

LiDAR-based SLAM can create fast two dimensions working space from a LiDAR with low computation resources. It has proven to generate very low-drift localization while mapping in real-world autonomous navigation scenarios. However, LiDAR-based SLAM is not exactly a full SLAM approach as it does not detect loop closures, and thus the map cannot be corrected when visiting back a previous localization. Loop closure detection can be implemented by combining the data of LiDAR and addition sensor (e.g. encoder, IMU, camera, …).

The laser scans can be used to build a map by employing a probabilistic approach. For a given robot's pose, each range measurement determines the coordinates of a cell. Cells that are behind the detected obstacles are registered as unknown cells whereas the cells that are between the sensor and the detected obstacles are registered as obstacle-free cells. The robot should be able to obtain the distance value from certain objects. The map (in Fig. 1) is the result of LiDAR-based SLAM experiment when the robot moved around a room. As mentioned above, LiDAR-based SLAM does not detect loop closure, so the bigger the environment the larger the error. To minimize this error, in this experiment, the new potion of the map can be updated and overwrote the map constantly. As a result, the largest linear error is 3 cm and the largest angular error is 5 degree.

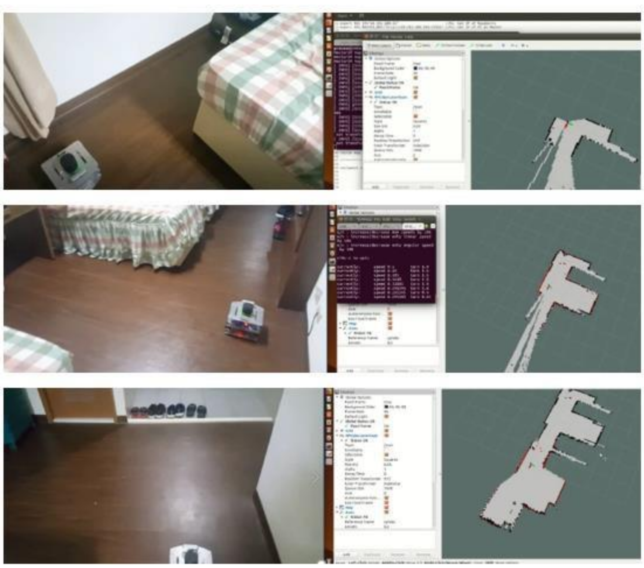


Figure 1

LiDAR-based SLAM results

2.2 Visual-based SLAM

In this study, RTAB-Map (Real-Time Appearance-Based Mapping) based on an incremental appearance-based loop closure detector is implemented. It consists of three stages: sensor measurement, frontend, and backend stages. In the frontend stage, the sensor data is processed and the geometric constraints between the successive RGB-D frames are extracted. The backend stage is focused on solving the accumulated drift problem and detecting the loop closure detection. To avoid the dead-reckoning problem, in our previous study, Explicit Loop Closing Heuristic (ELCH) [14] is implemented. This method updates the accumulated errors of the new frame's constraint. The error is distributed to all previous frames with proper weights. Using the Intel Realsense D435 sensor, the generated 3D map has proper quality. In addition, the *rtabrviz* interface gives several information while making 3D map as shown in Fig. 2. Window 1 is the RGB image that camera received. Window 2 shows the loop closure detection. Window 3 gives the image after applying SIFT algorithm and highlight points that will be used for feature matching. Window 4 shows 3D-point cloud map.

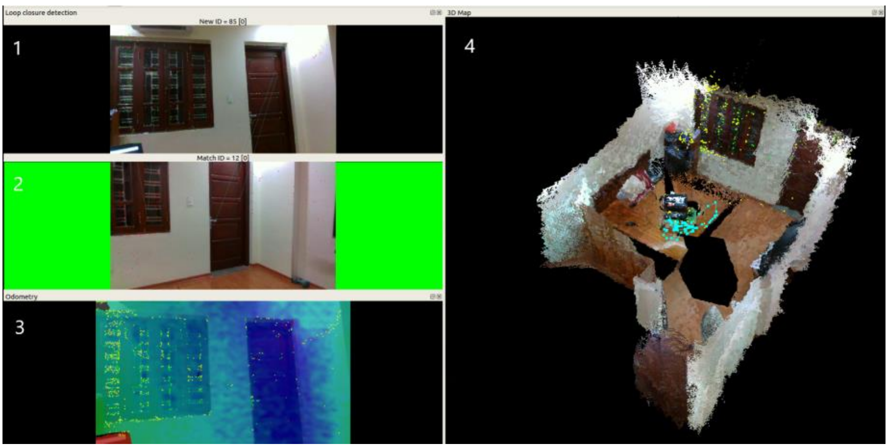


Figure 2
Visual- based SLAM result

An advantage feature of Visual-based SLAM over LiDAR-based SLAM is that the time of relocalization is significantly smaller. However, Visual-based SLAM works poorly or in some case fail in featureless environment. As shown in Fig. 3, the input data is a corner of the room which has no distinctive features for the detection algorithm. As a consequence, the system fails to conduct a reliable odometry.

Figure 3
Visual-based SLAM in featureless environment

2.3 Integrated LiDAR Visual-based SLAM

In this study, the sensors are configured competitively to improve the output odometry. Visual-based SLAM provides loop closure detection to increase localization accuracy. LiDAR-based SLAM provides wider range of data to increase the field of view (FOV) of the system to overcome featureless environment. Both sensors provide point cloud data with the information of a surrounding environment. Then, feature-matching algorithm will be applied to 3D point cloud to update the map and 2D point cloud will be used to deduce the odometry information. The pose information of the robot is obtained based on integrated information. The sensor fusion flow chart is presented in Fig. 4.

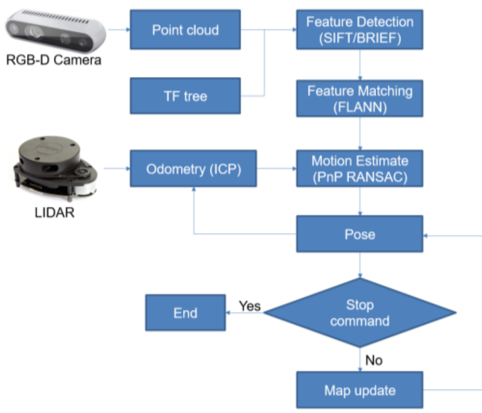
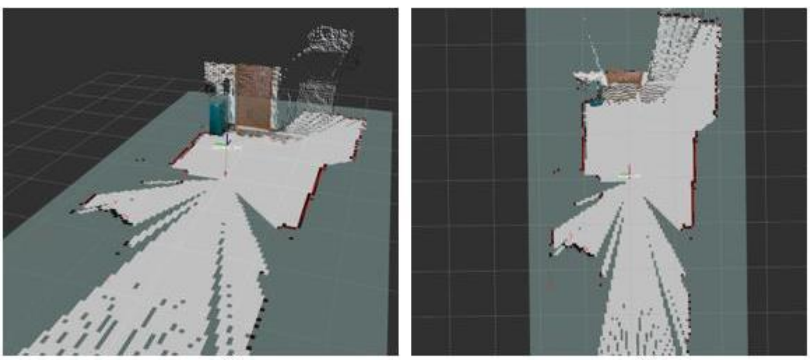


Figure 4
Sensor fusion flow chart

When the robot starts its program, both RGB-D camera and LiDAR sensor provide point cloud: one is 3D, the other is 2D. The transformation tree is predefined by user and provides coordinate information. When 3D point cloud is received, it will be scanned to detect some key features. The feature detection method used in this study is SIFT/ BRIEF. The SIFT (the Scale Invariant Feature Transform) is used to transform image data into scale-invariant coordinates relative to local features. It generates large numbers of features that densely cover the image over the full range of scales and locations.

The SIFT algorithm has four operations. Firstly, it estimates a scale space extreme based on the Difference of Gaussian (DoG). Secondly, it finds the key point localization by eliminating the low contrast points. Thirdly, a key point orientations are obtained based on local image gradient. Finally, it computes a descriptor for the local image region. For more detail, please refer to [24]. Binary Robust Independent Elementary Features (BRIEF) is another alternative method, which is applied in this study as requests less complexity than SIFT with similar matching performance. Feature Matching algorithm is implemented using Fast Approximate Nearest Neighbor Search (FLANN) [25]. Then, PnP (perspective-n-point) and RANSAC (Random Sample Consensus) are applied to enhance motion estimation [26]. Those features will be used to compare the older frame with newer frame to deduce the robot position and update the map. Iterationately, the closest neighbor of each point in the source is found by using a search algorithm and the rigid body transformation between the target points and their closest neighbors. The entered target point cloud is then transformed using the rigid body transformation estimation and a new closest neighbor search is performed. This process is iterated until convergence. The 2D point cloud also gives information of the robot position using ICP (Iterative Closest Point) to minimize the difference between two point-clouds. In ICP algorithm, the target is fixed while the source is transformed. The data from LiDAR and 3D camera are configured competitively to improve the odometry.



a) b)

Figure 5

Combining 3D point cloud with 2D grid map. a) 3D view; b) top view

The mapping process using ROS provides a graphical user interface named as *rtabmapviz*, which visualizes visual odometry, output of the loop closure detector, and a point cloud that is a 3D dense map. The reconstructed map is shown in Figure 5. In this experiment, the largest linear error/angular error of the combination method is smaller than sole methods.

3 Path Planning

The robot path planning problem is divided into classical methods and heuristics methods [28, 29]. Planning methods based on sampling-based motion planning (SBP) algorithms have applied on robot systems because of their capability in complex and/or time-consuming. SBP includes probabilistic roadmap (PRM) and rapidly-exploring random trees (RRT) [30]. Basicly, the path is generated by connecting points sampled randomly. This method is able to archive a feasible robot path relatively quickly, even in high-dimensional space [31-32].

In this paper, Bi-directional RRT* is proposed and implemented. Essentially, Bi-directional RRT* is variant RRT* algorithm in which the tree grows from both the starting point and the ending point. In other world, there will be two trees grow in the space. When two trees' nodes meet or close enough, a path is generated. Figure 6 compares the time-consuming of three path planning algorithms RRT*, extended RRT* and Bi-directional RRT* and the number of sampling nodes they need to generate. As can be seen from this figure, the Bi-directional RRT* only needs fewer than 150 sampling nodes to find the path in many trials while the RRT* need approximately 600 sampling nodes and the extend RRT* took over 800 nodes.

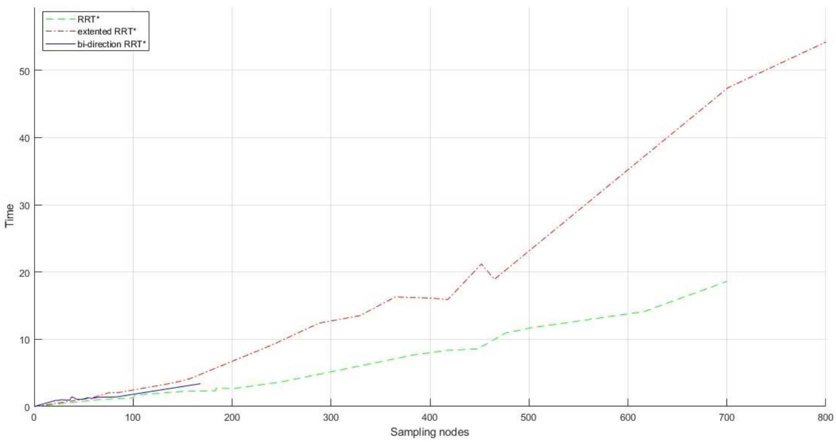


Figure 6

Comparing each variation of RRT* algorithm

At first, when the sampling node number is smaller than 180 nodes, three algorithms have an insignificant difference in time. The Bi-directional RRT*, however, always find a path within 5 seconds meanwhile the RRT* could need up to 20 seconds and the time for extended RRT* could over 1 minute with more than 1000 nodes generated. The actual execution time by the technique reveals that Bi-directional RRT* is dramatically fast in generating path, leading to a decrease in computational burden. Therefore, it can state that, the Bi-directional RRT* algorithm is reliable and satisfactory for our autonomous navigation application.

4 PID-Fuzzy Controller

Compared with advanced algorithms, Fuzzy-PID method is relatively easy implemented in the practical applications. Therefore, in this research, Fuzzy controller is applied to find optimal parameters of the PID controller. In which, the proportional parameter and integral parameter are continuously tuned by Fuzzy logic, based on feedback signal as shown in Figure 7.

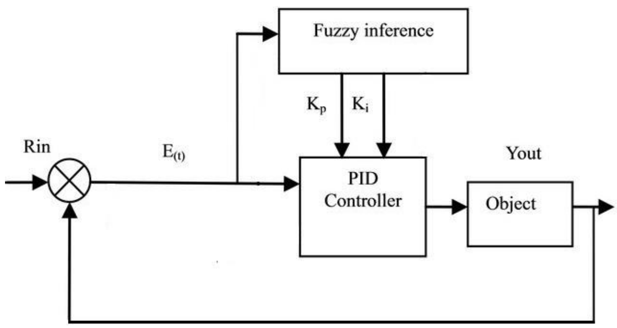


Figure 7
Self-tuning Fuzzy PID Controller

Figures 8-11 present the comparison of Left/Right wheel's velocities and Left/Right wheel's errors respectively among PID, Fuzzy and Fuzzy-PID controllers. The gain K_u and ultimate period P_u then create two separately controller by Ziegler-Nichols method - basic type and non-overshoot (no OS) type as in Table 1.

Table 1
The optimal parameters of the PID controller

Specification	K_p	K_i	K_d
Basic	$0.60 \times K_u$	$2 \times K_p / P_u$	$K_p \times P_u /8$
Non overshoot	$0.2 \times K_u$	$2 \times K_p / P_u$	$K_p \times P_u /3$

For basic-PID controller, with the calculated K_u = 1.35 and P_u = 0.083, the overshoot is low (about 3%). Settling time is around 0.25 second, peek time is about 0.2 second. In the same parameters, the non-overshoot PID controller has no overshoot, however, peek time - nearly 0.5 second is two-times slower than other controllers. The fuzzy controller has smaller overshoot than PID controller. In addition, the peak time and settling time that is faster than PID's ones. For Fuzzy-PID controller, the overshoot is smaller (about 2%), the peak time and settling time are slightly decrease comparing with Fuzzy controller. The wind-up problem is also minimized, and the system working process is smoothly. In conclude, a self-tuning Fuzzy-PID controller has better performance compared with sole Fuzzy and PID controllers. Therefore, this controller is applied for the mobile robot in this project.

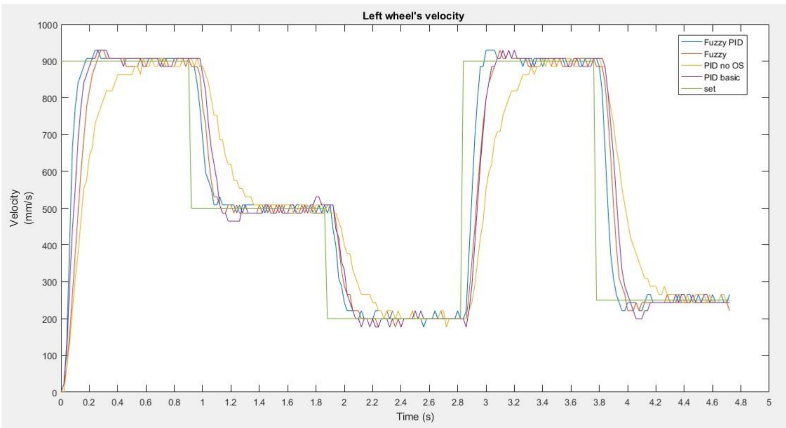


Figure 8
The comparision of PID and Fuzzy/PID controllers for Left wheel's velocities

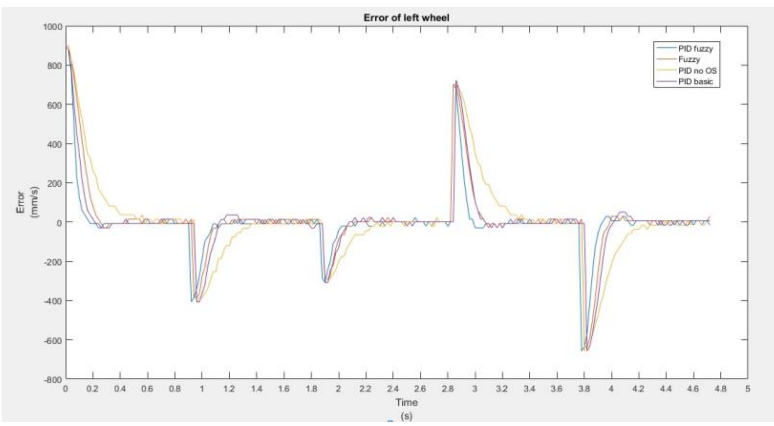


Figure 9
The comparision of PID and Fuzzy/PID errors Left wheel's error

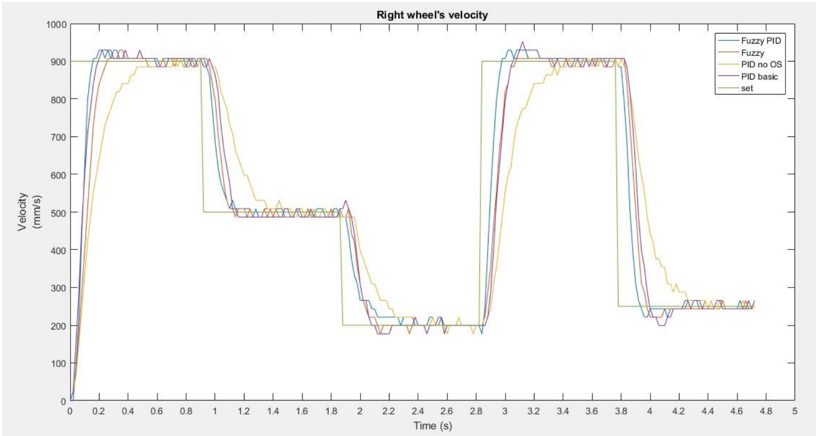


Figure 10
The comparision of PID and Fuzzy/PID controllers for right wheel's velocities

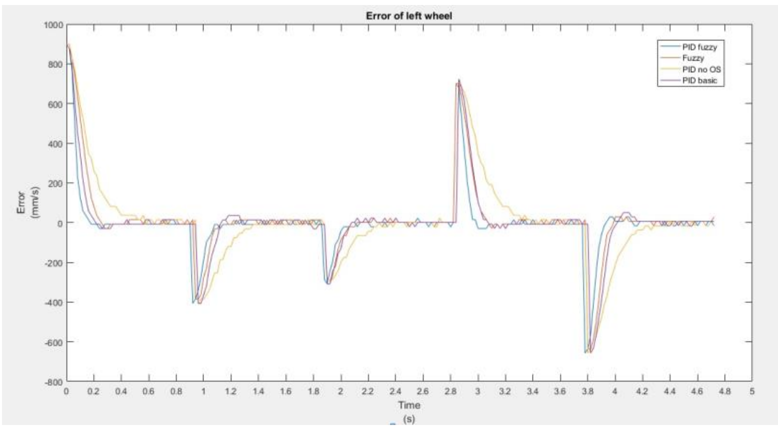


Figure 11
The comparision of PID and Fuzzy/PID errors right wheel's error

5 Experiment Results

We used Robot Operating System (ROS) for simulation environment. The experimental setup consists a mobile robot equipped with a Rplidar-A1 sensor and an Intel Realsense D435 sensor. The RPLIDAR A1 operates clockwise to generate an outline map for the robot working environment within 12 meters. The D435 sensor provides 3D real-time information precisely. The test cases is indoor environment with the approximate area 30 m^2. The robot traveled with a speed of 2 m/s. Figures 12 shows the robot mapping results using LiDAR-based SLAM and RRT* algorithm to avoid obstacle and reach the goal. By using LiDAR-based SLAM, the robot is able to conduct 2D map, however, the information of the robot working environment is quite simple.

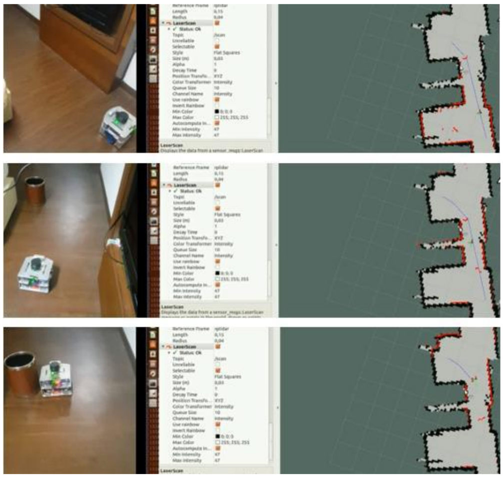


Figure 12
Robot navigation by RRT* algorithm and Fuzzy-PID controller

The results for the robot autonomous navigation using visual-based SLAM, and the integrated method are shown in Figures 13, 14, respectively. Compare to visual-based SLAM, integrated SLAM provides high resolution registered RGB-mapped point cloud. Furthermore, this methodology is able to reduce effort and time of the point cloud data collection. Based on the generated map, the robot performs its autonomous navigation with bi-directional RRT* and Fuzzy-PID controller and find the optimal path.

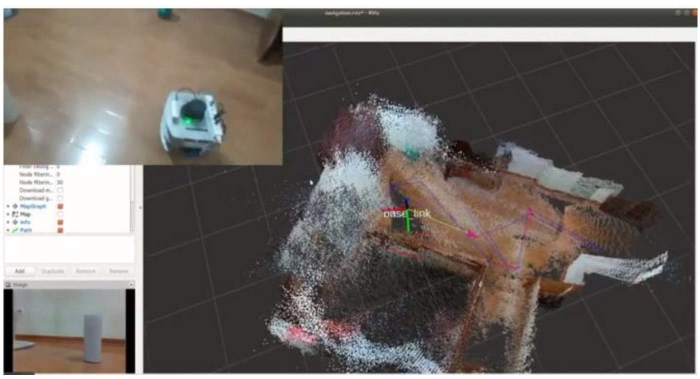


Figure 13
Visual-based SLAM autonomous navigation

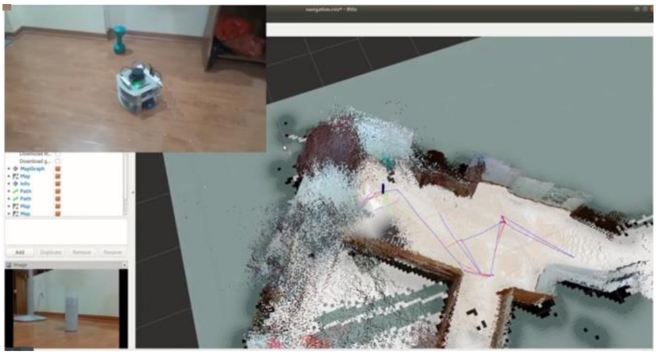


Figure 14
Hybrid LiDAR, Visual-based SLAM autonomous navigation

Conclusion and Future Work

In this paper, we have presented a full solution for integrated SLAM with LiDAR sensor and RGB-D camera. The solution is a combination of LiDAR, RGB-D camera data. LiDAR-based SLAM, can generate 3D point cloud with a little computational burden. However, in some scenarios such as long corridors, square-shaped rooms and open wide areas where no obstacle information can be acquired make the laser-based SLAM algorithms non-operational. On the other hand, visual SLAM with *Rtabmap* package comes with feature-matching and a 3D map that provide richer information on the surrounding environment. However, the computation time is large. The integrated SLAM reduces effort and time of the point cloud data collection, but still provides the high quality 3D-reconstructed map. In addition, we have implemented bi-direction RRT* path planning and Fuzzy-PID controller for the autonomous navigation purpose. Future work includes the development of the collaborations multiple mobile robot. The robot swarms can share information about their working environments for others. The proposed approach can be implemented in many real-life applications such as service robots in buildings, surveillance operations, agricultural robots, space exploration missions.

Acknowledgement

This work is granted by the National Foundation for Science and Technology Development of Vietnam NAFOSTED, Vietnam under project number 107.01-2018.331

References

[1] D. Schleicher, L. M. Bergasa, M. Oca˜na, R. Barea, E. L´opez, Real-time hierarchical stereo Visual SLAM in large-scale environments, Robotics and Autonomous Systems, Vol. 58 (2010), pp. 991-1002

[2] A. Gil, O. Reinoso, M. Ballesta, M. Juli´a, Multi-robot visual SLAM using a Rao-Blackwellized particle filter, Robotics and Autonomous Systems, Vol. 58 (2010), pp. 68-80

[3] G. Jiang; L. Yin; S. Jin; C. Tian; X. Ma; Y. Ou, A Simultaneous Localizationand Mapping (SLAM) Framework for 2.5D Map Building Based on Low-Cost LiDAR and Vision Fusion, Appl. Sci., Vol. 9, 2105, 2019

[4] H. Lee, H. Chae, K. Yi, A Geometric Model based 2D LiDAR/Radar Sensor Fusion for Tracking Surrounding Vehicles, IFAC-PapersOnLine, Vol. 52 (8), pp. 130-135, 2019

[5] P. M. Bloka, K. Boheemen, F. K. Evert, J. Isselmuiden, G. Kim, Robot navigation in orchards with localization based on Particle filter and Kalman filter, Computers and Electronics in Agriculture, Vol. 157, Feb 2019, pp. 261-269

[6] A. Yilmaz, H. Temeltas, Self-adaptive Monte Carlo method for indoor localization of smart AGVs using LIDAR data, Robotics and Autonomous Systems, Vol. 122 (2019), 10328, pp. 1-19

[7] M. R. U. Saputra, A. Markham, N. Trigoni, Visual SLAM and Structure from Motion in Dynamic Environments: A Survey, ACM Computing Surveys, 2018, Vol. 51(2), pp. 1-36

[8] D. Viejo, J. Rodriguez, M. Cazorla, A study of a soft computing based method for 3D scenario reconstruction, Applied Soft Computing, Vol. 12 (10), 2012, pp. 3158-3164

[9] M. S. Bahraini, M. Bozorg, A. B. Rad, SLAM In dynamic environments via ML-RANSAC, Mechatronics, Vol. 49 (2018) 105-118

[10] L. Xiao, J. Wang, X. Qiu, Z. Rong, X. Zou, Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment, Robotics and Autonomous Systems, Vol. 117 (2019), pp. 1-16

[11] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, D. Scaramuzza, SVO: Semidirect visual odometry for monocular and multicamera systems, IEEE Trans. Robot, Vol. 33 (2) (2017), pp. 249-265

[12] J. Engel, V. Koltun, D. Cremers, Direct sparse odometry, IEEE Trans. Pattern Anal. Mach. Intell, Vol. 40 (3) (2018), pp. 611-625

[13] T. Qin, P. Li, S. Shen, VINS-Mono: A robust and versatile monocular visual-inertial state estimator, IEEE Trans. Robot (2018) pp. 1-17

[14] H. Chen; C. Y. Lin, RGB-D Sensor Based Real-time 6DoF-SLAM, 2014 International Conference on Advanced Robotics and Intelligent Systems (ARIS), pp. 61-65

[15] Q. Sun, J. Yang, X. Zhang, F. Sun, RGB-D SLAM in Indoor Environments With STING-Based Plane Feature Extraction, IEEE/ASME Transactions on Mechatronics, Vol. 23(3), 2018, 1071-1082

[16] Y. Chen, S. Huang, R. Fithch, Active SLAM for Mobile Robots with Area Coverage and Obstacle Avoidance, IEEE/ASME Transactions on Mechatronics, 01 January 2020, doi: 10.1109/TMECH.2019.2963439

[17] W. Zhang, Q. Chen, W. Zhang, X. He, Long-range terrain perception using convolutional neural networks, Neurocomputing, Vol. 275 (2018), pp. 781-787

[18] Y. Sun, M. Liu, M. Q. H. Meng, Motion removal for reliable RGB-D SLAM in dynamic environments, Robot. Auton. Syst, Vol. 108 (2018) pp. 115-128

[19] C. Yu, et al., DS-SLAM: A semantic visual SLAM towards dynamic environments, in: presented at the 2018 IEEE International Conference on Intelligent Robots and Systems (IROS), 2018

[20] R. Scona, M. Jaimez, Y. Petillot, M. Fallon, D. Cremers, Staticfusion: background reconstruction for dense rgb-d slam in dynamic environments, in: presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018

[21] H. Williams, W. N. Browne, D. A. Carnegie, Learned Action SLAM: Sharing SLAM through learned path planning information between heterogeneous robotic platforms, Applied Soft Computing, Vol. 50 (2017), pp. 313-326

[22] J. Park, J. Song, Sensor fusion-based exploration in home environments usinginformation, driving and localization gains, Applied Soft Computing, Vol. 36 (2015) pp. 70-86

[23] H. Hu, H. Su, P. Ye, Q. Jia, X. Gao, Multiple Maps for the Feature-based Monocular SLAM System, Journal of Intelligent & Robotic Systems, Vol. 94 (2), 2019, pp. 389-404

[24] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, Int. J. Comput. Vision, Vol. 60(2), Nov. 2004, pp. 1-28

[25] M. Muja, David G. Lowe, Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, VISAPP International Conference on Computer Vision Theory and Application, 2009, pp. 331-340

[26] Y. Zheng, Y. Kuang, S. Sugimoto, K. A° strom, M. Okutomi, Revisiting the PnP Problem: A Fast, General and Optimal Solution, ICCV 2013

[27] M. Jordan, A. Perez, Optimal Bidirectional Rapidly-Exploring Random Trees, Tech. Rep. MIT-CSAIL-TR-2013-021, Computer Science and Artificial Intelligence

[28] T. T. Mac, C. Copot, D. T. Tran, R. De Keyser, Heuristic approaches in robot path planning: A survey, Robotics and Autonomous Systems 86, pp. 13-28, 2016

[29] T. T. Mac, C. Copot, R. De Keyser, M. Ionescu, The Development of an Autonomous Navigation System with Optimal Control of an UAV in Partly Unknown Indoor Environment, Mechatronics, Vol. 49, pp. 187-196, 2018

[30] J. Lee, O. Kwon, L. Zhang, S. E. Yoon, A selective retraction-based RRT planner for various environments, IEEE Trans. Robot. 30 (4) (2014) 1002-1011

[31] K. Qiana, Y. Liu, L. Tian, J. Bao, Robot path planning optimization method based on heuristic multi-directional rapidly-exploring tree, Computers & Electrical Engineering, Vol. 85, 2020, 106688

[32] C. Yuan, G. Liu, W. Zhang, X. Pan, An efficient RRT cache method in dynamic environments for path planning, Robotics and Autonomous Systems, Volume 131, September 2020, 103595