Fast Active SLAM for Accurate and Complete Coverage Mapping of Unknown Environments

Kruno Lenac[1], Andrej Kitanov[1], Ivan Maurović[1], Marija Đakulović[1], and Ivan Petrović[1]

Faculty of Electrical Engineering and Computing
Unska 3, Zagreb
University of Zagreb, Croatia

Abstract. In this paper we present an active SLAM solution with an active loop closing component which is independent on exploration component and at the same time allows high accuracy robot's pose estimation and complete environment mapping. Inputs to our SLAM algorithm are RGBD image from the Kinect sensor and odometry estimates obtained from inertial measurement unit and wheel encoders. SLAM is based on the Exactly Sparse Delayed State Filter for real-time estimation of robot's trajectory, vision based pose registration and loop closing. The active component ensures that localization remains accurate over a long period of time by sending the robot to close loops if a criterion function satisfies the predefined value. Our criterion function depends on the number of states predicted without an update between predictions, information gained from loop closing and the sheer distance between the loop closing state location and the current robot location. Once a state in which a loop closure should occur is reached and an update is performed, the robot returns to its previous goals. Since the active component is independent on the exploration part, the SLAM solution described in this paper can easily be merged with any existing exploration algorithm and the only requirement is that the exploration algorithm is able to stop exploration at any time and continue the exploration after the loop closing was accomplished. In this paper, we propose an active SLAM integration with the 2D laser range finder based exploration algorithm that ensures the complete coverage of a polygonal environment and therefore a detailed mapping. The developed Active SLAM solution was verified through experiments which demonstrated its capability to work in real-time and to consistently map polygonal environments.

1 Introduction

There are three main tasks that autonomous mobile robot must be able to accomplish in order to successfully complete the given tasks. It must be able to explore the environment, to build its map and to localize itself in that map. The tasks of localization and mapping must be done simultaneously since the map landmarks can not be created without knowing their location and localization can not be performed without a map to localize in. Therefore, these two tasks are always considered as one problem known as Simultaneous Localization And Mapping (SLAM). Most solutions to the SLAM problem belong in one of the two main groups. The first group includes SLAM algorithms who have the motion model and the measurement model defined in the state space with added white Gaussian noise (e.g. [1–3]). The second group of SLAM algorithms has models representation in a form of a set of particles with general non Gaussian distribution (e.g. [4,5]). The distinction in representation also exists dividing them to feature based and pose based SLAM systems, where the later uses a pose graph and a set of independent local maps while the former marginalize them out in order to obtain the robot's current pose and map.

SLAM is mostly treated as a passive system which means that it does not send any commands to the robot - it only acquires sensor data and uses that data to build a map of surrounding environment and to localize the robot in that map, i.e. it does not decide where the robot must go. Controlling the robot is crucial in order to autonomously map the complete environment and to ensure the system observability [6] and mapping accuracy. That is the reason why an active SLAM with exploration algorithm is required. The exploration algorithm scans robot's environment and decides where it must go in order to efficiently explore the environment. An active SLAM ensures that uncertainty of SLAM localization and mapping remains within desired boundaries by changing the robots trajectory. In [7] *"localization metric"* is introduced which provides the uniform basis for measuring localization quality. The localization quality over a trajectory is combined with the navigation cost of the trajectory and information gained from the environment by following the trajectory in one single criterion. That criterion is used for determining on which trajectory should the robot travel on. Numerical method that uses a non-linear Model Predictive Control (MPC) for estimating the pose and the map error, that will occur by following one trajectory, is introduced in [8]. In [9] relative entropy is used as a measure for the information gain. Environment is discretized into grids and optimal trajectories (according to information gain criterion) are planned on the global scale thus minimizing unnecessary loop closures and noise while ensuring more precise maps. In [10] information gain is also used as a criterion for choosing the trajectory. The difference is that Rao-Blackwellized particle filter is used for SLAM and entropy calculation. In [11] global planning is avoided by using attractors in combination with a local planning strategies. The attractor is placed in the environment according to the current robot goal (explore, improve map or improve localization). The attractor then influences information gain computed by a local planner which uses MPC. One different approach to minimize localization and mapping errors is used in [12], where reinforcement learning is used to generate a robot's motion in such a way that it minimizes error generation in the mapping process. This approach enables the usage of a simple exploration strategy while maintaining the location and map accuracy. In [13], the Fast SLAM is used who is independent on the exploration algorithm and planing. When localization uncertainty reaches the predefined level the exploration task is stopped and possible previously visited states are considered for loop closing. The state with the

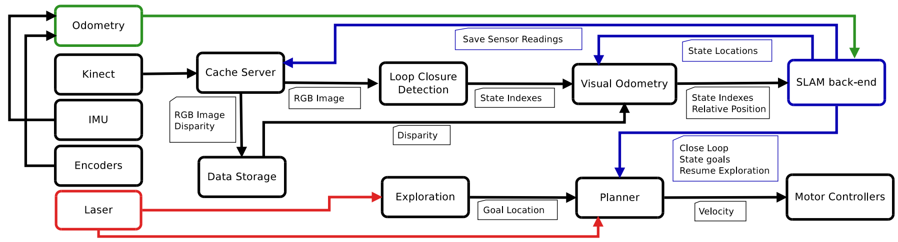

Fig. 1: System integration: Active SLAM with exploration

highest information gain and the easiest reachability is chosen. When the robot reaches that state it follows its previously traversed path until the uncertainty drops below a desired level and then the exploration is continued.

The active SLAM solution presented in this paper is similar to [13]. The main difference is that our SLAM is based on the Exactly Sparse Delayed State Filter (ESDSF) [14] which makes state choosing for loop closing much easier because of a more suitable state representation in an information filter opposed to a particle filter. The particle filter maintains a set of trajectories as particles. Selection of the most likely movement allowing the robot to follow the previous path requires the additional processing. The other more important reason to use information filter is a possibility to reduce trajectory uncertainty almost to the zero because there will not exist representational pdf loss problem (particle depletion) [15] that leads to filter divergence. However, we will not use uncertainty as a fixed measure for cancelling the exploration since it can lead to a longer and repetitive loop closures that are not necessarily required for the exploration tasks. In contrast to the local optimal planning strategies, our solution is much simpler and faster and optimizes trajectory indirectly by searching only for global goals in which loop closures appear since these are the most informative ones and can provide a required accuracy even with locally non optimal path planner.

Our exploration cancelling criterion will be dependent on the number of predicted states without an update, information gain and on a distance to the state for possible loop closure. The loop will be followed only until the first large enough loop closure update has been accomplished and then the exploration will be continued. We have combined this active SLAM with an exploration algorithm which is used for high detailed 3D thermal mapping [16] and which provides a capability for complete coverage mapping in polygonal environments. Our active SLAM maintains high localization accuracy that enables thermal mapping to be accomplished in fewest possible scans of the environment for the entire area to be mapped. Developed integrated system is shown in Fig. 1 whose main modules are described in the rest of the paper.

2 SLAM

The pose based SLAM systems rely on a fact that trajectory estimation and map building are conditionally independent. This enables the separation of the active SLAM and exploration algorithms as the local maps are independent when conditioned on a specific trajectory. The SLAM system employed in this work is a visual 3D pose based SLAM that estimates the robot's trajectory as a set of discrete states with their associated local maps. Local maps are consisted out of RGBD measurements, i.e. planar features extracted from them.

In Fig. 1 it can be seen that our SLAM system consists out of four main parts (Server, Image matching, Visual Odometry and SLAM back-end). A server component is responsible for storing and synchronizing sensor measurements. Whenever a new state is augmented to the SLAM state vector, a server stores the current RGBD image. RGB image is sent to an image matching component which then compares the RGB image of the current state with RGB images of all other states. If the match is found, a disparity measurements of matched states are retrieved by Visual odometry. Visual odometry then computes relative position between those states and sends it to a SLAM back-end. The SLAM back-end is responsible for trajectory optimization. The filter prediction is done whenever a robot's position and a position of the last augmented state differ more than predefined value while the update occurs whenever visual odometry sends a relative position between the two states.

2.1 SLAM Back-end

The SLAM back-end algorithm is based upon Exactly Sparse Delayed State Filter (ESDSF), developed in [17]. ESDSF is used for the estimation of a Gaussian robot's trajectory x which consists of n pose samples $x_i, i = 1, \dots, n$:

$$x = \begin{pmatrix} x_0 \\ x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix}, \; x \sim \; N(\mu, \sum) = N^{-1}(\eta, \Lambda)$$

where the relation between μ and η is: $\eta = \Lambda\mu$, and between the covariance $\sum$ and an information matrix Λ: $\Lambda = \sum^{-1}$. As elaborated in [17], an information matrix Λ of this system has sparse structure which makes SLAM computationally and memory efficient when implemented to use the sparse algebraic system solvers and therefore significantly gains an execution speed.

2.2 Image Matching

Image matching is essential for SLAM algorithm since it is used for a loop closing detection. In our SLAM system the image matching is done using FabMAP2 algorithm [18]. FabMAP2 algorithm compares images by converting them to a bag of words model and by using the Chow Liu tree [19]. From every received image features are extracted using e.g. SURF [20], SIFT [21] and those features are converted to a bag of words. Those words are then compared with the words contained in a previously learned dictionary. Images are then classified by the number and type of the words they contain, and that are in dictionary. After the classification, images can easily be compared based on the classification result. Additionally to comparing the images using a bag of words FabMAP2 uses Chow Liu tree to determine whether or not the images are a match.

2.3 Vision Odometry

Vision odometry is used to determine the relative position between two states that FabMAP2 has found a match for. Visual odometry algorithm used in our SLAM system is the robot vision library (RVL) developed and presented in [22]. RVL first generates 2.5D triangle mesh grid from disparity recorded by the RGBD sensor. Planar surfaces are then extracted using the Region merging and growing algorithm [23] for both views. Finally, modified RANSAC [24] algorithm is used to determine the relative position between states by registration of extracted planar features.

3 Active SLAM

In our active SLAM solution SLAM is completely separated from the route planning and exploration. Only when SLAM detects that possible loop closing is nearby and that too many states have been predicted without an update it begins to rank all possible loop closing states. If the state for possible loop closure has a high enough information gain then the SLAM sends command to exploration algorithm to stop exploring and sends the robot to close the loop. For a loop closure to be considered, two conditions have to be met:

- **condition 1:** more than N_s states have been predicted without an update or an update occurred between states x_i and x_j and $|i-j| < N_{ij}$, where N_{ij} is predefined number of states;
- **condition 2:** Euclidean distance between the current position and position of the possible state for loop closing has to be lower than a defined value d_m.

The first condition ensures that a loop closing is required in order to maintain precision of localization and map and the second condition ensures that a robot will not have to travel long to achieve the loop closing. If both conditions are met then all the states that satisfy second condition are possible candidates for loop closing. Additional condition that a state has to satisfy in order to be chosen for the loop closing is calculated from the topological map of states.

Topological map of states is represented by a $n \times n$ matrix G, where n is a current number of states. Matrix G is a binary matrix (elements are 0 and 1). Matrix element (i,j) is 1 if a state i is connected with the state j. States are connected if they were predicted one after another or if an update occurred between them. Topological matrix is used to determine how much would the update between two states impact the overall map and localization quality. In general update will impact overall quality more if the two states (i and j) are farther away from each other. The problem is that the sum of Euclidean distances between all states from state i to state j would be a wrong measure as it is illustrated in Fig. 2. Although the sum of Euclidean distances between all states from C to S is high, since an update occurred between the states 2 and 10, the overall information gained from an update between states C and S is small. This is because a lot of information gained from loop closing between C and S was already gained by loop closing between states 2 and 10. This is why a third condition is necessary for detecting the good states to initiate a loop closing:

$$d_M(x_i, x_j) > d_t \tag{1}$$

where d_M is topological distance between the states i and j and d_t is a predefined topological distance threshold. Topological distance is computed from the graph G_f generated from the matrix G. Nodes are represented by the states and connections between nodes i and j exist if the element (i,j) in matrix G is 1. Weight of connection is Euclidean distance between the states i and j. Topological distance of states (i,j) is calculated as the shortest path from the node i to the node j in the graph G_f. The shortest path is calculated using the A* algorithm since A* is efficient when no replanning is required. For example, topological distance from the node C to the node S (Fig. 2) would be the sum of Euclidean distances between the states (S,2),(2,10),(10,11),(11,12) and (12,C). If topological distance is high enough the state is chosen for a loop closing. When a robot arrives to that state and an update does not occur a robot continues to follow the loop until a good enough update occurs. The update is considered as good enough if it occurred between the states which indexes (i,j) satisfy the condition $|i - j| > N_{\text{loop}}$, where N_{loop} is a predefined number of states. This condition ensures that the update really occurred between current location and the state that was augmented when a robot first traversed the loop. If an update passes this condition the control is transferred to exploration algorithm and a robot continues to position that was selected by the exploration algorithm before a loop closing was initiated.

4 Exploration

The aim of the exploration algorithm is to find a minimal number of positions from where to take scans in order to build a detailed map of the environment. The algorithm needs to guarantee a complete exploration of the environment within a finite number of measurements. We used the exploration strategy described in [16] and [25], which is an extension of Ekman's exploration algorithm [26] by removing the rigid constraints on the range sensor and a robot localization. In [26] was shown that the exploration of polygonal environments guarantees a complete coverage considering no positional uncertainty and an ideal range sensor. Under the assumption that in our SLAM the uncertainty is lower each time when active SLAM closes a loop, our exploration strategy completely explores the environment.

The exploration starts with no *a-priori* information on the environment and after the first laser scan in 2D an initial environment model is generated. The model consists out of lines vectorized from the initial scan. Based on the initial map and information available from the first scan, the next best robot position is calculated. The next candidate scan positions are defined 1m in front of the lines which separate explored and unexplored area, so-called jump edges. The jump edges are generated by connecting the two adjacent points in one scan if the distance between points is above some threshold value, i.e. connecting discontinuities in a range data. For details see [16].

Figure 3 shows generated candidate positions p_1 and p_2 from the current position R. The jump edges are marked with red color. Among all jump edges

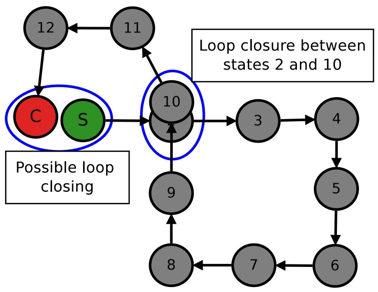

Fig. 2: Topological distance

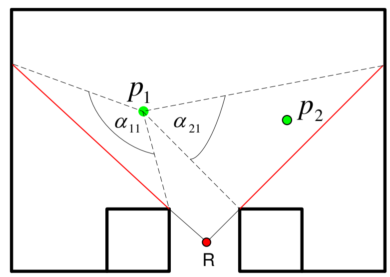

Fig. 3: Jump edges from the current robot position R

the next best scan position is chosen according to a criterion that maximizes the area explored in the next scan and minimizes a distance from the current robot position to the jump edge. Estimation of the explored area in the next step is done by the angle between potential candidate position p and two jump edge ends as shown on Fig. 3. Also the angles according to the other jump edges are taken into account what leads to the following criterion for candidate position p_j:

$$I_j = k_1 \frac{1}{d_j} + k_2 \sum_{i=1}^{N} \alpha_{ij}, \tag{2}$$

where d_j represents a path length from the current robot position to the candidate position, while α_{ij} refers to the angle between candidate position P_i and P_j, and k_1 and k_2 are tuning parameters used to treat angles and path distance equally. In each step the exploration node communicates with a path planning module, receiving the distance to the all candidates position, i.e. path length. The best candidate scan position is sent to the planner module which drives the robot to it to take the scan. The whole procedure is then repeated. As already explained the exploration activity with a planner has to stop when the loop closing is active and continues when the loop has been closed.

The final aim of the exploration is to have a dense and precise map. In a case when a localization is not perfect the environment could be completely explored but the final map could differ from the real environment edges generating a non precise final map. The number of scan positions needed to cover the whole environment could also vary depending on the localization quality. Including an active SLAM the number of scans can be lower and the map precisely represents the environment.

5 Experimental Results

All experiments were done using the equipment shown in Fig. 4. Sensors were mounted on Clearpath A200 (Husky) all terrain mobile platform. Laser scanner used for obstacle avoidance and exploration was SICK LMS100-1000, which has a maximum range of 50m. Since Husky has differential drive with four wheels,

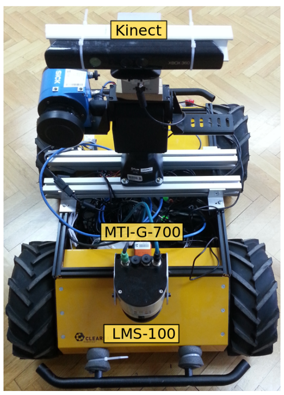

Fig. 4: Equipment used in experiments

Name	Value	Description
N_s N_s	18 20	Minimal number of predicted states after which loop closing starts if no update occurred or update occurred between states with index difference smaller than N_{ij}
d_m	8m	Maximum Euclidean distance between the robot and the state for the state to be considered for a loop closing
d_t	25m	Minimum topological distance between the robot and the state for the state to be considered for a loop closing
N_{loop}	10	Minimum difference between indexes of the states that update occurred between required for a loop closing to end.

Table 1: Description of used parameters

a rotation angle estimation using only encoders would produce significant errors due to the slippage. This would require very frequent loop closures in order for SLAM to produce a reliable localization. That is why we used encoders only for estimating robot's movements in forward/backward direction and why Inertial Measurement Unit (IMU) was used for estimating the angle of rotation. IMU sensor used in experiments was Xsense MTi-G-700 GPS/INS. For depth registration and RGB image acquisition we used Microsoft Kinect. All experiments were done using Lenovo Thinkpad E531 portable computer with Intel Core i7 4th generation mobile processor and 4GB of RAM.

To test the exploration algorithm in conjunction with an active SLAM, the robot was put in an environment without any prior knowledge about it. The main task was to build a map of entire environment. In order to test an active SLAM performance two experiments were completed: the first one with an active SLAM turned off and the second one with an active SLAM turned on. In both experiments parameter values were the same and are listed in table 1.
In both experiments the map was created using the union of point clouds recorded by a laser at each time step during motion and taken from locations estimated by the SLAM. The map generated in this way shows localization error because as error increases the point clouds that should represent a scan of same location overlap less and less thus creating distortions on the map.

Figure 5 shows the generated map and the robot trajectory when an active SLAM was turned off. Although the trajectory in the first (left) room robot explored seems like loop closing should be detected, it was not. The robot's angle of rotation was too different and the RGB images did not match. This is a clear example where an active SLAM would help. Since this large loop closing was missed, odometry errors that accumulated over time were not corrected. As a result we have a moderate localization error making distortions of the map of the first room explored and even more distortions in the second room that was explored. Figure 6 shows a map of the same area but generated with an active

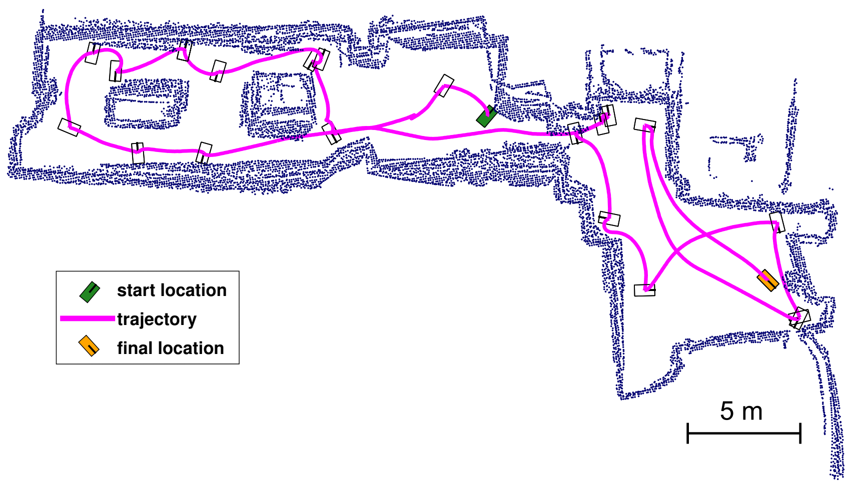

Fig. 5: Generated map and trajectory without active SLAM (black rectangles represent the robot footprint at all goals it was sent to).

SLAM turned on. Robot's trajectory is divided in three main parts. The first part (marked with magenta color) represents a robot's motion before an active SLAM cancelled exploration. Parameter N_s from the section 3 was set to 18. Since there were no major loop closures up till the state 18 was augmented, an active SLAM started searching for the possible loop closures immediately after augmentation. The maximum Euclidean distance (d_m), for the second condition, was set to 8m. We can see that this condition was also met for several states during the robot's motion from state 18 to a location marked with red 'x'. The only condition left to be satisfied, in order for the loop closing process to start, was minimal topological distance d_t. Parameter d_t was set to 25m. When a robot arrived at the position marked with a red 'x', the state 2 satisfied this condition. In that moment the exploration was cancelled and the robot was sent to the state 2. When it arrived at the state 2 it continued to follow the previously traversed path up till the state 29 was augmented. This part of the robot's motion is shown in Fig. 6 as the green trajectory. When the state 29 was augmented, a loop closure was detected between the state 10 and the state 29. Since this was a large loop closure, the robot exited loop closing process and started going to the goal previously set by the exploration (marked with purple icon in Fig. 6). As

it can be seen from the last part of the robot's trajectory, no other location was suitable for an active SLAM to initiate the loop closing and the robot continued exploring until the whole area was explored.
The generated map is much better compared to a map from experiment without an active SLAM. Both rooms have very little deformations. The passage between the two rooms is clear, scans of objects and walls in the second room that robot visited are almost completely overlapped and the angle between the two rooms is correct. These two experiments show that there is a considerable improvement

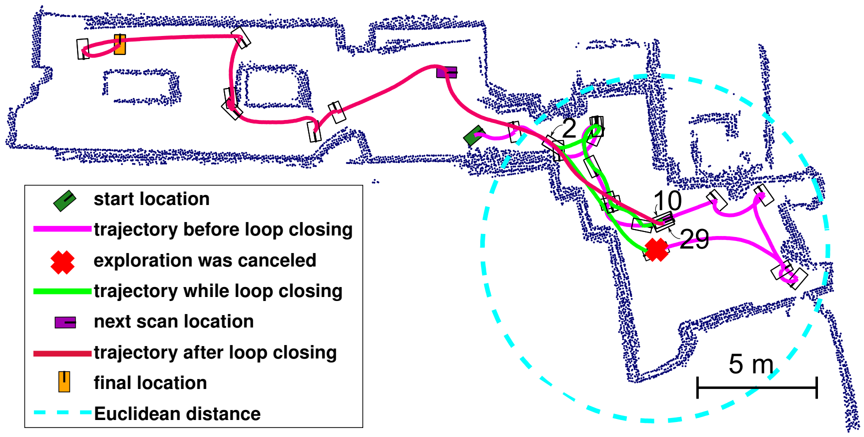

Fig. 6: Generated map and trajectory with active SLAM (black rectangles represent all goals robot was sent to, with the exception of rectangle labelled with number 29, it represents augmented SLAM state).

in the map accuracy when the active SLAM is used. The overall results would be even better if we had a possibility to use 3D laser sensor instead of Kinect. This is because a Kinect has a maximum range for depth images of about 4m. This is good enough in the small areas but in a larger open space Kinect depth image is almost useless to RVL since no planes can be extracted. That is a reason why only just few updates between neighbouring states were detected and that is why a robot needed to close the loop for a longer time in order to find an adequate state for a loop closing. If we had a 3D laser on our disposal we are confident that the overall results would be much better and this is our primary goal in a future research.

6 Conclusion

In this paper we presented a complete solution for exploring the unknown environments while maintaining the localization and map accuracy. Our exploration algorithm ensures a complete area mapping in finite number of scans while our active SLAM algorithm ensures that localization remains accurate by sending the robot to close loop if criterion function satisfies a predefined value.

Our active SLAM system is based on the Exactly Sparse Delayed State Filter which is an appropriate state representation for motion planning to follow previous path and fast enough for a real-time execution. The other and more important reason to use information filter is a possibility to reduce the trajectory uncertainty almost to zero because no representational pdf loss problem exists which leads to the filter divergence. However, we do not use uncertainty as a fixed measure for cancelling the exploration since it can lead to a longer and repetitive loop closures that are not necessary required for the exploration tasks.

In experiments we have shown that our solution works efficiently and that indeed an active SLAM is required in order to maintain the localization and map accuracy but we also encountered some problems. Although the active SLAM realization, as a module separated from the exploration and planner, benefits from allowing the simple connection with many different planning and exploration algorithms, it has also a disadvantage because an active SLAM is cut out from the trajectory planning process. Our future research will be concentrated on solving that problem. We will include an active SLAM in the exploration goals selection and path planning in order to minimize the requirement for a loop closing.

7 Acknowledgements

This work has been supported by the European Community Seventh Framework Programme under grant No. 285939 (ACROSS).

References

1. J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel, "1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010.
2. M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1365–1378, 2008.
3. U. Frese, "Efficient 6-dof slam with treemap as a generic backend," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 4814–4819.
4. G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 34–46, 2007.
5. M. Montemerlo, S. Thrun, and B. Siciliano, *FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics*, ser. Springer Tracts in Advanced Robotics. Springer, 2007, vol. 27.
6. A. Kitanov and I. Petrović, "Generalization of 2d slam observability condition," in *5th European Conference on Mobile Robots (ECMR2011)*, 2011.
7. A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *IEEE / RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 534–539.
8. S. Huang, N. Kwok, G. Dissanayake, Q. Ha, and G. Fang, "Multi-step look-ahead trajectory planning in slam: Possibility and necessity," in *IEEE / RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5026–5031.

9. R. Sim and N. Roy, "Global a-optimal robot exploration in slam," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 661–665.
10. C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters," in *Proceedings of Robotics: Science and Systems*, 2005.
11. C. Leung, S. Huang, and G. Dissanayake, "Active slam using model predictive control and attractor based exploration," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 1091–1096.
12. T. Kollar and N. Roy, "Using reinforcement learning to improve exploration trajectories for error minimization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 3338–3343.
13. C. Stachniss, D. Hahnel, and W. Burgard, "Exploration with active loop-closing for fastslam," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.
14. R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based slam," *Robotics, IEEE Transactions on*, vol. 22, no. 6, pp. 1100–1114, 2006.
15. R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," in *NIPS*, 2000, pp. 584–590.
16. D. Borrmann, A. Nüchter, M. Đakulović, I. Maurović, I. Petrović, D. Osmanković, and J. Velagić, "The project thermalmapper – thermal 3d mapping of indoor environments for saving energy," in *Proceedings of the 10th International IFAC Symposium on Robot Control (SYROCO '12)*, 2012.
17. A. Kitanov and I. Petrović, "Exactly sparse delayed state filter based robust slam with stereo vision," in *The joint conference of the 41st International Symposium on Robotics (ISR 2010) and the 6th German Conference on Robotics (ROBOTIK 2010)*, 2010.
18. M. Cummins and P. Newman, "Highly scalable appearance-only slam - fab-map 2.0." in *Robotics: Science and Systems*. The MIT Press, 2009.
19. C. K. Chow and C. N. Liu, "Approximating Discrete Probability Distributions With Dependence Trees," *IEEE Transactions on Information Theory*, vol. IT-14, pp. 462–467, 1968.
20. H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.
21. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
22. R. Cupec, E. K. Nyarko, D. Filko, and I. Petrović, "Fast pose tracking based on ranked 3D planar patch correspondences," in *10th IFAC Symposium on Robot Control SYROCO*, vol. 10, no. 1, 2012, pp. 108–113.
23. F. Schmitt and X. Chen, "Fast segmentation of range images into planar regions," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. IEEE, 1991, pp. 710–711.
24. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography." *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
25. M. Đakulović, Š. Ileš, and I. Petrović, "Exploration and mapping of unknown polygonal environments based on uncertain range data," *Automatika*, vol. 52, no. 2, pp. 118–131, 2011.
26. A. Ekman, A. Torne, and D. Stromberg, "Exploration of polygonal environments using range data." *Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 2, pp. 250–255, 1997.