LiDAR-Based Localization for Formation Control of Multi-Robot Systems

Tobias Recker, Bin Zhou, Marvin Stüde, Mark Wielitzka, Tobias Ortmaier and Annika Raatz

Abstract

Controlling the formation of several mobile robots allows for the connection of these robots to a larger virtual unit. This enables the group of mobile robots to carry out tasks that a single robot could not perform. In order to control all robots like a unit, a formation controller is required, the accuracy of which determines the performance of the group. As shown in various publications and our previous work, the accuracy and control performance of this controller depends heavily on the quality of the localization of the individual robots in the formation, which itself depends on the ability of the robots to locate themselves within a map. Other errors are caused by inaccuracies in the map. To avoid any errors related to the map or external sensors, we plan to calculate the relative positions and velocities directly from the LiDAR data. To do this, we designed an algorithm which uses the LiDAR data to detect the outline of individual robots. Based on this detection, we estimate the robots pose and combine this estimate with the odometry to improve the accuracy. Lastly, we perform a qualitative evaluation of the algorithm using a Faro laser tracker in a realistic indoor environment, showing benefits in localization accuracy for environments with a low density of landmarks.

T. Recker (✉) · B. Zhou · A. Raatz
Leibniz University Hannover, Institute of Assembly Technonolgy, An der Universität 2, 30823 Garbsen, Germany
E-mail: Recker@match.uni-hannover.de
URL: https://www.match.uni-hannover.de

M. Stüde · M. Wielitzka · T. Ortmaier
Leibniz University Hannover, Institute of Mechatronic Systems, At the Universit at 1, 30823 Garbsen, Germany

T. Schüppstuhl et al. (eds.), *Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2021*,
https://doi.org/10.1007/978-3-030-74032-0_30

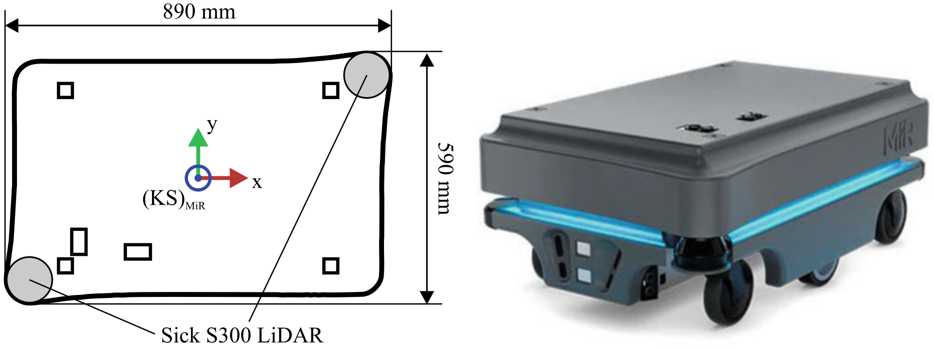
Keywords
Multi-robot systems • Formation control • Relative localization

1 Introduction

Robot formations are often used to extend a single robot's capabilities or break down complex tasks into simpler subtasks Feng et al. [1]. In this way, robot formations increase the flexibility of the overall robotic system and improve its ability to do new tasks. A group of robots is required to maintain their formation even under the influence of disturbances to function as a unit. This is done by the formation control. Depending on the control law, the formation control requires the robot's positions to be known either regarding a static map or relative to the other robots. This localization is either done using external sensors (external camera or laser tracker) Irwansyah et al. [2] or internal (on-board) sensors (internal cameras, LiDAR, ultrasonic, ultrawideband ...) Nguyen et al. [3],Li et al. [4],Güler et al. [5],Bisson et al. [6] together with knowledge of the environment (map). There are also solutions that combine both principles, for example, an external transmitter sends a signal that is then received by internal sensors. Examples of this are triangulation methods based on time of flight measurements Choi et al. [7] or the recognition of artificially placed landmarks with known positions Zhang et al. [8]. While external localization is generally more precise, it does require more equipment and limits the working area. This is why we will focus on internal sensors in this work.

When calculating two robots' relative positions using internal sensors, it is common to localize both robots within the map and then subtract the absolute positions. The main drawback of this method is the introduction of several sources of error. Firstly, the localization within a given (perfect) map contains a rather significant error, which alone amounts to an inaccuracy of several mm Chan et al. [9],Gang et al. [10]. Secondly, the map does not provide a perfect representation of the real environment, which reduces accuracy even further. Also, individual robots in a robot formation can obstruct the line of sight of other robots to important features in the map, making localization more difficult. To counteract this type of error, researchers have tried to estimate the relative position of the other robots directly Wasik et al. [11],Franchi et al. [12],Teixido et al. [13],Huang et al. [14],Rashid and Abdulrazaaq [15]. For example, this is done by detecting the outline of other robots in the laser scanner data Wasik et al. [11]. For this purpose, the robot's contour is approximated by simple shapes such as rectangles or circles. These shapes are then searched for in the measurement data of the laser scanners Teixido et al. [13]. If a sufficient number of points can be detected on the robot's outline, the position of the robot can be determined. If not, the position can be estimated based on the last known position by using an extended Kalman filter (EKF) taking the robot's velocity either from odometry or derivation of the previous positions Huang et al. [14].

While there has been much research in this area, we could not find a solution perfectly fitted to our needs. While researchers like Huang et al. Huang et al. [14] were able to localize robots relative to each other, the pose estimation performance is insufficient. In general, either the frequency of the estimation Wasik et al. [11] or the accuracy Huang et al. [14],Franchi et al. [12],Koch et al. [16] was insufficient.[1] In other cases, the algorithms used require features (markers) Teixido et al. [13], a special robot shape Howard et al. [18] or additional sensors Bisson et al. [6] not present on our hardware.



(a) MiR 200 - schematic view (b) MiR 200 - manufacturer photo

Fig. 1 Dimensions of the MiR 200 [mobile-industrial-robots.com]

2 System Overview

For our use case, we want to determine the position and orientation (pose) of several MiR 200 robots. The MiR 200 is a rectangular mobile robot with a footprint of 890 mm by 580 mm (see Fig. 1). It is equipped with two sick s300 laser scanners on opposing corners of the robot, giving it a 360-degree detection radius without any blind spots. In this publication, we want to use both of these scanners to directly estimate the position of all robots in our formation without the need to subtract error-prone absolute positions. For this, we are proposing the algorithm shown in Fig. 2. In the first step, we fuse the sensor data from the front and rear laser scanner. We then use the map to remove every data point caused by an obstacle. Afterward, we use Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to detect the potential robots in the remaining data. The next step is to check if

[1] In another publication Recker et al. [17] we have already investigated the theoretical accuracy of our formation control and developed mechanical compensation units to compensate for the lateral errors within the formation. Due to the mechanical design of the compensation units, they can only compensate errors of individual robots up to 100 mm.

the potential robots are one of our MiR 200 robots. We do this by checking if the shape of the respective point-cloud matches the shape of one of our robots. The last step is to compute the position and orientation and validate it.

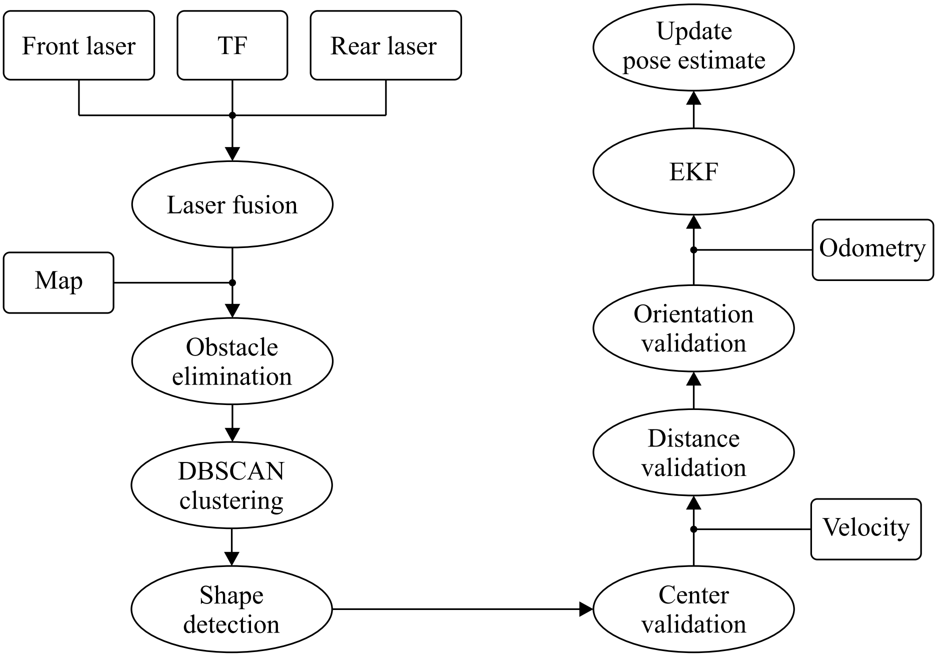


Fig. 2 Flowchart of the proposed algorithm

3 Algorithm

As described above, our algorithm starts by removing known objects (obstacles) from the measurement data. For this, we use a standard adaptive Monte Carlo Localization (AMCL) algorithm to estimate the pose of every robot within our map. Next, we remove every data point within close proximity (0.1 m) of a known obstacle within the map.[2] This process is shown in Fig. 3. Note that this step is not mandatory for our algorithm to work and can be skipped if no map is available or if localization in the map is impossible.

The next step is to detect clusters of points, which could be a robot. For this, we use an implementation of DBSCAN clustering (based on Ester et al. [19]) to assign every point

[2] The safety PLC of the MiR 200 will trigger an emergency stop if it detects an object within 0.1 m. Accordingly, we except our robots to be at least 0.1 m away from every object.

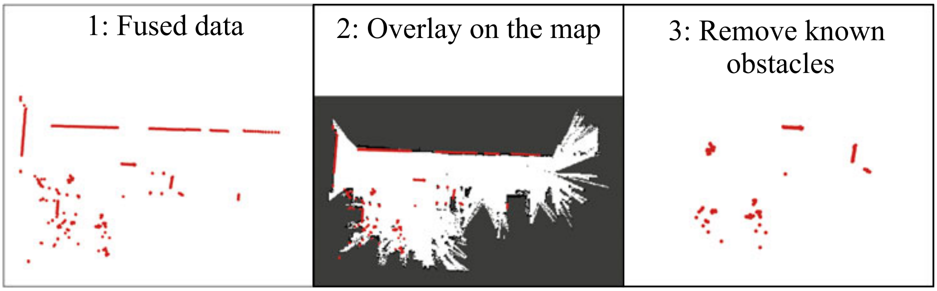


Fig. 3 Eliminating obstacles from the data based on the map

to a group of points and remove noise from the data. As our laser scanner has an angular resolution of about 0.4 degrees and the smallest side of our robot is 0.58 m long, we expect the scanners to detect at least 16 points on our robot - given the robot is less than 5 m away. Accordingly, every cluster consisting of less than 16 points can be disregarded. The step of clustering is shown in Fig. 4.

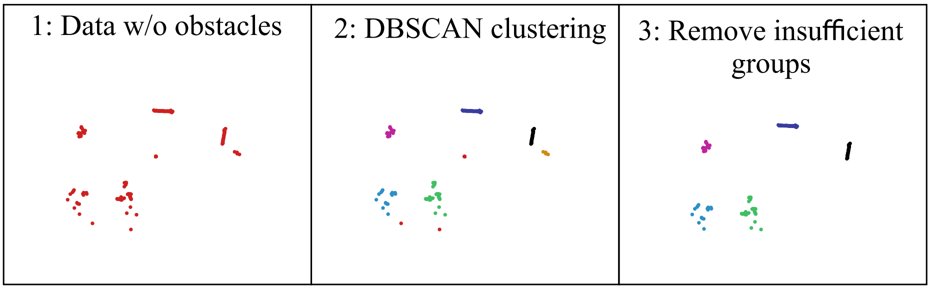


Fig. 4 DCSCAN clustering to remove scattered points

The third step is to remove clusters that do not fit the shape or size of a robot (see Fig. 5). As our robot is rectangular, only straight lines or 90-degree corners are valid shapes. In addition, we require the length of the point clusters to match the length of one of our robots sides with a maximum deviation of 20%. To do this detection, we expand on work from Zhang et al. Zhang et al. [20] by adjusting the geometric parameters to our robot and changing the orientation estimation to a multi-stage process. Instead of trying all possible orientations between 0 and 90 degrees with a fixed step size, we make several passes with decreasing step size. For the next stage, we only reduce the step size for the search space between the two angles, which previously yielded the highest correspondence between measured values and robot shape.

Applying all the previous steps, we are left with two similar groups of points. As booth groups form a line, there are four theoretical robot positions for each line (see Fig. 6). Out of

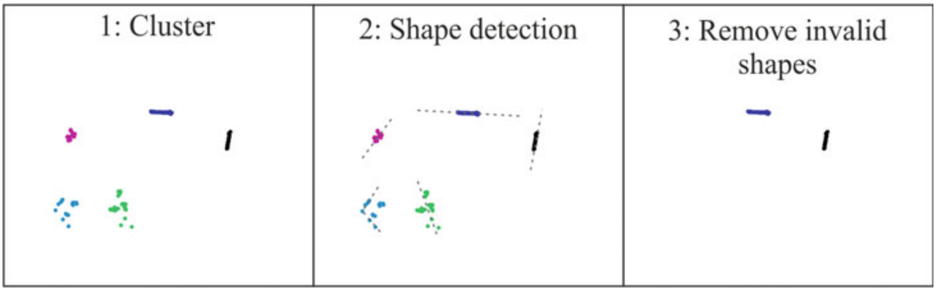


Fig. 5 Shape detection to remove (dynamic) obstacles with the wrong shape

these four positions, we eliminate the two with the closest distance to the measuring robot. If a robot were at one of those two positions, the structure of the robot would obstruct the line of sight to the detected line. Next, we compare the length of the point cluster to the length of our robot's front and side. This step is quite prone to errors. Because of the noise of the sensors, and because of the clustering, the length of the visible side cannot always be determined correctly. Therefore, the last step is to compare the new position estimate with the previous one. By comparing the two estimates and the velocity readings from the odometry, the most likely new position can be determined. In this way, other objects with a similar shape to the robot can also be excluded.

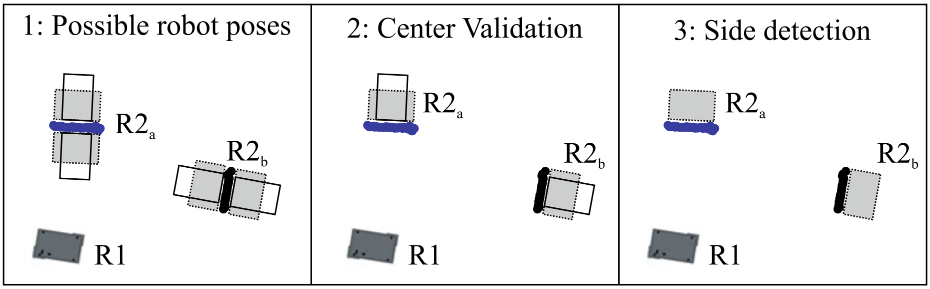


Fig. 6 Center validation to eliminate impossible robot positions

Remark: The last step shown here necessarily requires the exchange of data between the individual participants of the formation. However, in many cases, this exchange is not possible or desired. Given that situation, it would also be possible to get the velocities of the other robots by differentiating their last known positions (we also validated this in our experiments). Since in this case, the accuracy will most likely decrease in some situations (also see Sec. 4), we, therefore, recommend using the real velocity data if available.

Since the measurement frequency of LiDAR systems is usually lower than the measurement frequency of the wheel encoders (in our 12 Hz vs. 50 Hz), we suggest a sensor

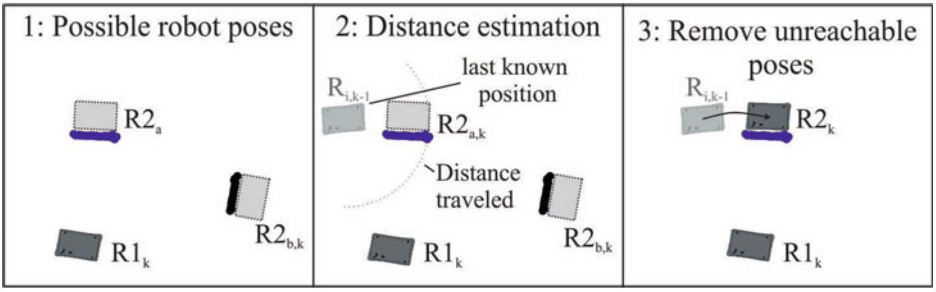


Fig. 7 Travel distance estimation to eliminate poses too far apart

fusion using an EKF. Our implementation is based on the `robot_pose_ekf`[3] package using a plugin made for GPS signals to update the prediction with the localization from our algorithm. As sensor data fusion with an EKF has already been covered extensively in the literature Khatib et al. [21]Eman and Ramdane [22]Huang et al. [14], we do not want to elaborate on it any further in this publication.

4 Evaluation

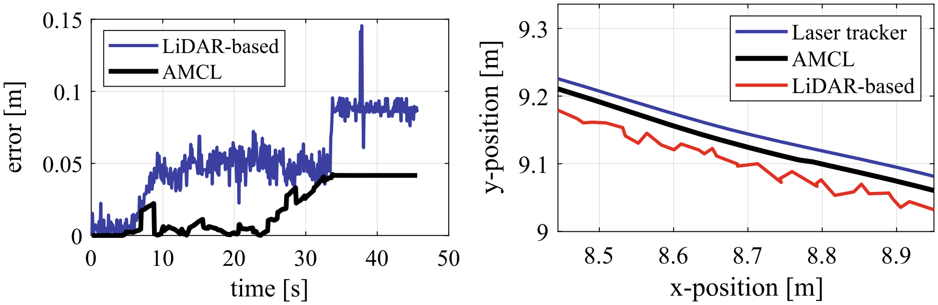
As mentioned before, we are using the MiR 200 industrial robot for the evaluation of our algorithm. In the first experiment (scenario I) we used two of these robots in our laboratory. To get an impression of the achievable accuracy, in this experiment, we only move one of the two robots and measure its position via the LiDAR of the other robot. We then compare the position estimated from our algorithm with a standard AMCL implementation. For AMCL, we have used the existing implementation in the ROS navigation stack and parameterized it manually to the best of our knowledge. In this scenario, we will not transfer any data between the robots, so the velocity of the robot has to be derived from previous position data. For reference, we use an external laser tracker to get the “true” relative position and distance. Our Faro Vantage laser tracker provides 3-DOF position measurements with an accuracy of under 0.1 mm and a frequency of 1000 Hz.

For the second experiment, we perform a simulation. This is necessary because we want to investigate the behavior of the algorithm compared to AMCL in an environment with fewer features. The simulation environment Gazebo also allows us to measure the true position of all robots in real-time, which would not be possible in reality (due to a lack of sensors). In the simulation, we will therefore be able to move both robots simultaneously and still get their true positions. Besides, we now transfer the odometry data between both robots. For the S300 laser scanners we are assuming a standard deviation of the noise of 29 mm for distances under 3 m and 1% of the distance above 3 m (according to the datasheet). The

[3] https://github.com/ros-planning/robot_pose_ekf.

encoders are modeled with a noise equal to 1% of the current velocity. Again we will try to estimate the pose of both robots using AMCL and the direct LiDAR-based detection we described in this article.

When comparing the measured trajectories in Fig. 9 (a), it becomes apparent that for the scenario I conventional distance estimation via AMCL is superior to the direct distance measurement. Figure 8 also confirms that in this case, both the average and maximum deviation are lower, and the signal is less noisy. We assume that due to the very good map and the high density of recognizable features in the surrounding area, localization via AMCL is better suited to determine the relative position in this and similar cases. By combining odometry and LiDAR, the AMCL can also compensate most of the noise for an individual robot. Since, in this scenario, we did not transfer the odometry data between robots, we can not use the EKF to reduce the noise in the LiDAR data for our algorithm. As a result, we recognize that applying the presented algorithm does not make sense under the condition of a good map and many features in the environment. Therefore, in the following, we test our algorithm under the assumption that the map is known poorly, or only a few features are available for localization via AMCL.



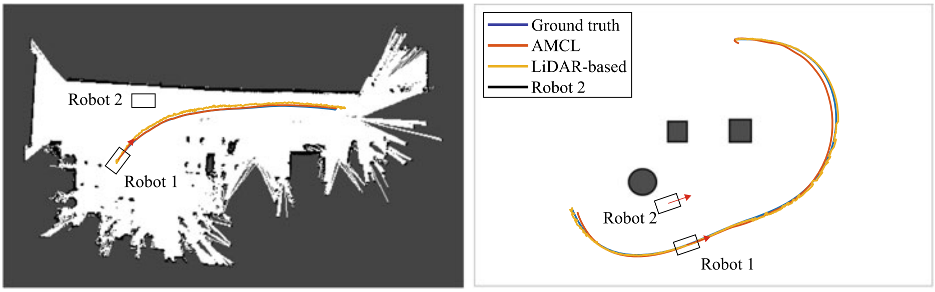
(a) Error in the relative distance

(b) Section of the global path

Fig. 8 Scenario I: One real robot moving - one real robot recording

In the second scenario, our robots are starting close to three landmarks in an otherwise empty environment. The robots then move along an oval trajectory around those landmarks. You can see this in Fig. 9 (b). Since the AMCL has significantly fewer features available, the localization and thus the determination of the relative position is less accurate. This is especially apparent in Fig. 10 (a), where starting at 30 s, two of the landmarks are partially covered by the second robot. Since the localization via AMCL is based exclusively on odometry without a line-of-sight connection to known features, the error temporarily doubles.

However, this experiment also shows a weakness in our proposed method: If the angle between the two robots is a multiple of 90 degrees (e.g., the robots are moving side by side), only one side of the other robot is detected. This is also visible in Fig. 10 (18 sec to 25 sec).

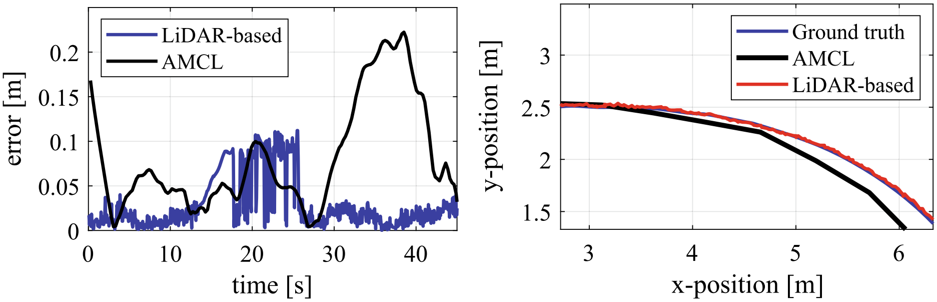


(a) Scenario I: Trajectory + Map

(b) Scenario II: Trajectory + Environment

Fig. 9 Global trajectories for scenario I and II

Since no corner can be identified in the contour, the determination of the start- and endpoint of the robot is subject to greater uncertainty. Also, in some cases, there is a slight delay between measured and real position, which also has a negative influence on the error. This delay occurs when the computer of our robot is busy with another process, and the necessary calculations are not completed in time. In general, we can still calculate a new position with every new LiDAR measurement, which averages to about 12,8 Hz on an Intel Core i7 8700k with 16 Gb of RAM using a Preempt-RT Linux Kernel (5.9.1-rt20).[4] Overall our method does provide some benefits, especially in environments with only a few landmarks.



(a) Error in the relative distance

(b) Section of the global path

Fig. 10 Scenario II: Both simulated robots are moving - robot one measures robot 2

[4] In normal operation the processor is not fully utilized by the algorithm. Only in the case that a real time process occupies the processor the computing time cannot be kept.

5 Conclusion and Future Work

In this paper, we introduced an improved method for the estimation of relative position for multi-robot formation control. We then compared the performance and applicability of this method to another standard method for relative robot localization. Although our algorithm does not always achieve better results, there is a clear area of application. Additionally, in all cases we studied, we did not require any form of communication or map for the algorithm to function. In the future, we plan on improving the algorithm regarding computing time. This could be done, for example, by first estimating the new robot position and, based on this estimation, removing distant objects from the measured data. Also, we will analyse more edge cases like two robots getting extremely close to each other. Another interesting aspect is the combination/fusion of different localization methods. As our method shows advantages in certain scenarios and AMCL in others, we plan to combine both algorithms to generate a more robust localization.

References

1. Feng, Z., et al.: An overview of collaborative robotic manipulation in multi-robot systems. Annual Reviews in Control - **49**, 113–127 (2020)
2. Irwansyah, A., Ibraheem, O.W., Hagemeyer, J., et al.: FPGA-based multi-robot tracking. Journal of Parallel and Distributed Computing **107**, 146–161 (2017)
3. Nguyen, T. H., Kim, D. H., Lee C. H. et al.: Mobile Robot Localization and Path Planning in a Picking Robot System Using Kinect Camera in Partially Known Environment. In: International Conference on Advanced Engineering Theory and Applications, pp 686–701 (2016)
4. Li, X., Du, S. , Li, G. et al.: Integrate Point-Cloud Segmentation with 3D LiDAR Scan-Matching for Mobile Robot Localization and Mapping, Sensors 2020 20(1), 1–23 (2020)
5. Güler, S., Abdelkader, M., Shamma, J. S.: Infrastructure-free Multi-robot Localization with Ultrawideband Sensors. In: American Control Conference (ACC), pp. 13–18 (2019)
6. Bisson, J., Michaud, F., Létourneau, D.: Relative positioning of mobile robots using ultrasounds. Intelligent Robots and Systems **1783–1788**, (2003)
7. Choi, W., Li, Y., Park, J. et al: Efficient localization of multiple robots in a wide space. In International Conference on Information and Automation, pp. 83–86 (2010)
8. Zhang, H., Zhang, L., Dai, J.: Landmark-Based Localization for Indoor Mobile Robots with Stereo Vision. International Conference on Intelligent System Design and Engineering Application, pp. 700–702 (2012)
9. Chan S., Wu P., Fu L.: Robust 2D Indoor Localization Through Laser SLAM and Visual SLAM Fusion. IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1263–1268 (2018)
10. Gang, P. et al.: An Improved AMCL Algorithm Based on Laser Scanning Match in a Complex and Unstructured Environment. Complexity, pp. 1–11 (2018)
11. Wasik, A. et al.: Lidar-Based Relative Position Estimation and Tracking for Multi-Robot Systems. In: Robot 2015: Second Iberian Robotics Conference, pp. 3–16 (2015)
12. Franchi, A., Oriolo, G., Stegagno, P.: Mutual localization in a multi-robot system with anonymous relative position measures, In; IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3974–3980 (2009)

13. Teixido, M., Pallejà, T., Font, D., et al.: Two-Dimensional Radial Laser Scanning for Circular Marker Detection and External Mobile Robot Tracking. Sensors **12**, 16482–97 (2012)
14. Huang, G.P., Trawny, N., Mourikis, A.I., et al.: Observability-based consistent EKF estimators for multi-robot cooperative localization. Auton Robot **30**, 99–122 (2011)
15. Rashid, A., Abdulrazaaq, B.: RP Lidar Sensor for Multi-Robot Localization using Leader Follower Algorithm. The Iraqi Journal of Electrical and Electronic Engineering, pp. 21–32 (2019)
16. Koch, P., May, S., Schmidpeter, M., et al.: Multi-Robot Localization and Mapping Based on Signed Distance Functions. J Intell Robot Syst **83**, 409–428 (2016)
17. Recker T.,Heinrich, M., Raatz, A.: A Comparison of Different Approaches for Formation Control of Nonholonomic Mobile Robots regarding Object Transport, CIRPe – 8th CIRP Global Web Conference on Flexible Mass Customisation, (in publication) (2020)
18. Howard, A., Mataric, M.J., Sukhatme, G. S.: Putting the 'I' in 'team': an ego-centric approach to cooperative localization. In: IEEE International Conference on Robotics and Automation, pp. 868–874 (2003)
19. Ester, M., Kriegel, H.P., Sander, J. et al.:A density-based algorithm for discovering clusters in large spatial databases with noise, Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
20. Zhang, X., Xu, W., Dong, C., et al.: Efficient L-shape fitting for vehicle detection using laser scanners. IEEE Intelligent Vehicles Symposium (IV) **54–59**, (2017)
21. Khatib, E. I. Al., Jaradat, M. A., Abdel-Hafez M. et al.: Multiple sensor fusion for mobile robot localization and navigation using the Extended Kalman Filter. In: 10th International Symposium on Mechatronics and its Applications (ISMA), pp. 1–5 (2015)
22. Eman, A., Ramdane, H.: Mobile Robot Localization Using Extended Kalman Filter. In: 3rd International Conference on Computer Applications & Information Security (ICCAIS), pp. 1–5 (2020)