DOI 10.2478/ama-2021-0004

EFFICIENT NON-ODOMETRY METHOD FOR ENVIRONMENT MAPPING AND LOCALISATION OF MOBILE ROBOTS

Mateusz ADAMOWICZ[*], **Leszek AMBROZIAK**[*], **Mirosław KONDRATIUK**[*]

[*]Department of Robotics and Mechatronics, Faculty of Mechanical Engineering, Bialystok University of Technology,
ul. Wiejska 45C, 15-351 Białystok, Poland

mateusz.adamowicz.1994@gmail.com, l.ambroziak@pb.edu.pl, m.kondratiuk@pb.edu.pl

received 13 October 2020, revised 14 April 2021, accepted 19 April 2021

Abstract: The paper presents the simple algorithm of simultaneous localisation and mapping (SLAM) without odometry information. The proposed algorithm is based only on scanning laser range finder. The theoretical foundations of the proposed method are presented. The most important element of the work is the experimental research. The research underlying the paper encompasses several tests, which were carried out to build the environment map to be navigated by the mobile robot in conjunction with the trajectory planning algorithm and obstacle avoidance.

Key words: mobile robot, mapping, navigation, obstacle detection, obstacle avoidance

1. INTRODUCTION

In recent years, robotics researchers have shown an increasing interest in autonomy of mobile vehicles. The autonomy of robots is mainly associated with obstacle detection and avoidance systems (Gao et al., 2019; Rulin, 2017), precise navigation systems (Romaniuk et al., 2016; Bakkali et al., 2007), new control methods allowing adaptation to failures (Ambroziak et al., 2019; Lanzon et al., 2014), cooperation of a few objects, and formation and swarm motion (Ambroziak et al., 2015; Kownacki et al., 2019). Mentioned topics relate to tasks carried out by mobile robots in indoor and outdoor conditions. In both cases the problem of navigating a robot in an unknown environment without the possibility of global positioning system (GPS), real-time localisation system (RTLS), etc. usage is a current topic. Frequently, the task of navigating a mobile robot is based on a map built during a mission, which is at the same time used to try to localise the robot. This is known as the Simultaneous Localisation and Mapping (SLAM) technique, and has been discussed in the works of Dissanayake et al. (2000) and Dissanayake et al. (1999). SLAM is one of the key enabling technologies of mobile robot autonomy and is regarded as one of most important problems in the pursuit of building fully autonomous objects. Despite a significant progress in this topic, it is still a great challenge and an open case (Chen et al., 2017). Currently available SLAM methods provide good solutions for mapping static, structured environments with limited volume–size. However, facilitating a robot to autonomously navigate a map of unstructured, dynamic and large-size environment continues to remain an emerging and open research problem. In many different SLAM techniques we can distinguish full (Guivant et al., 2002) and online (Dubbelman et al., 2015) ones. Other common distinctions are volumetric and feature-based (Li et al., 2019), topological and metric (Li et al., 2010), active and passive (Soragna et al., 2019), and single- and multi-robot (Xiaolin et al., 2012). All SLAM techniques use different methods of filtration and position estimation. The most popular one is the usage of Extended Kalman Filters (EKF SLAM). EKF SLAM was introduced in Smith et al. (1990) and Moutarlier et al. (1989); these were single state vectors used to estimate robot locations. A second set of algorithms solving the SLAM are the graph-based optimisation techniques. Graph-based approach was first described in Lu et al. (1997) and Dellaert (2005). The third group of SLAM techniques is based on particle filters, which is a kind of nonparametric statistical filtering technique. This method was introduced in Zhu et al. (2019) and mostly popular in online SLAM because of grooving availability of cheap and efficient computers on module.

SLAM techniques and all previously listed methods require the use of appropriate sensors and are using odometry information. Methods that do not use odometry are not found in the literature and we can mention a few papers such as Klecka et al. (2014), where authors propose non-odometry feature based approach to SLAM.

This paper addresses the problem of SLAM without any additional odometry measurement sensor such as inertial measurement unit (IMU) or encoders for movements' calculations. The theoretical foundations of the proposed method are presented. The most important element of the work is the experimental research: a number of tests were carried out to build the environment map to be navigated by the mobile robot in conjunction with the trajectory planning algorithm and obstacle avoidance.

2. SENSOR SYSTEM USED IN STUDIES

Different SLAM methods use different types of sensors such as CCD cameras, rangefinders, lidars, and RGBD sensors. In this work, scanning laser range finder UTM30-LX (Fig. 1) was used. This kind of sensor emits a pulse every quarter, half or one degree over an angular range of 270° and returns a planar cross-section of the observed scene in polar coordinate form. The use of a specific type of sensor requires checking its measurement quality. The quality of the measurement depends on the following stages.

Fig. 1. Laser range finder UTM30-LX

Laser rangefinder UTM30-LX allows to measure distances in the range of 0.1–30.0 m. An important parameter is the standard deviation of the measurement from the actual value, because it determines the resolution of the environment map (it is not possible to make an environment map with an accuracy of 1 mm if the standard deviation is, for example, 10 mm).

Fig. 2 shows standard deviation of the scanning laser rangefinder measurements at different angles of surface, which reflected the beam of radiation coming from the sensor. There is a relatively large increase in the standard deviation for position of the reflecting surface other than perpendicular to the laser beam, but these are still quite small values (Nakamura et al., 2014).

On the basis of the results shown in the Fig. 2, it can be stated that it is possible to make the environment map with the accuracy of 1 cm with using scanning laser rangefinder.

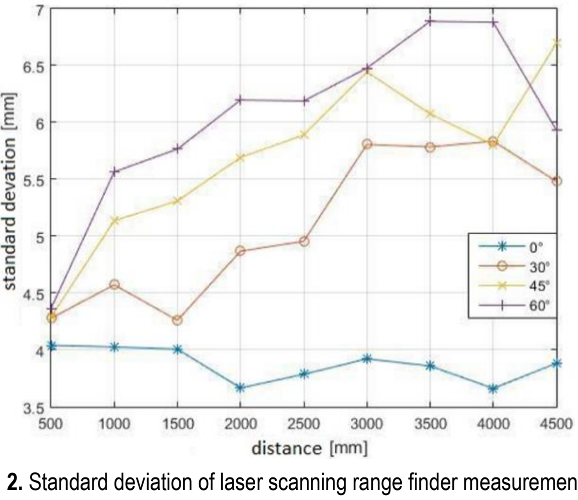

Fig. 2. Standard deviation of laser scanning range finder measurements

3. DESCRIPTION OF THE LOCALISATION AND MAPPING ALGORITHM

The main idea of the algorithm is the same as in the article (Janah et al., 2018). The localisation method is based on applying scan (cloud of points from one measurement taken with scanning laser rangefinder) to the environment map. Applying a scan consists in performing a rotation and displacement of it such that the greatest number of points overlaps the map of the environment – the more points imposed, the better the scan fits and thus more accurate the position and orientation relative to the surroundings.

The difference between the presented approach and the approach from work (Janah et al., 2018) is the way of moving and rotating the scan in order to match it with the map. In Janah et al. (2018) the Particle Swarm Optimisation (PSO) algorithm was used, while in this work a brute force algorithm was chosen (description shown below).

Firstly, measurements from one scan are collected and stored. If the first scan is done, the collected points are saved on the map. In case of other scans, calculations are made to determine the location. If the location has changed, the points collected in this scan are placed on the map. After mapping, if the scan was the last one, the program ends. In any other case, the next measuring cycle starts. Fig. 3 presents the flowchart of the proposed localisation and mapping method.

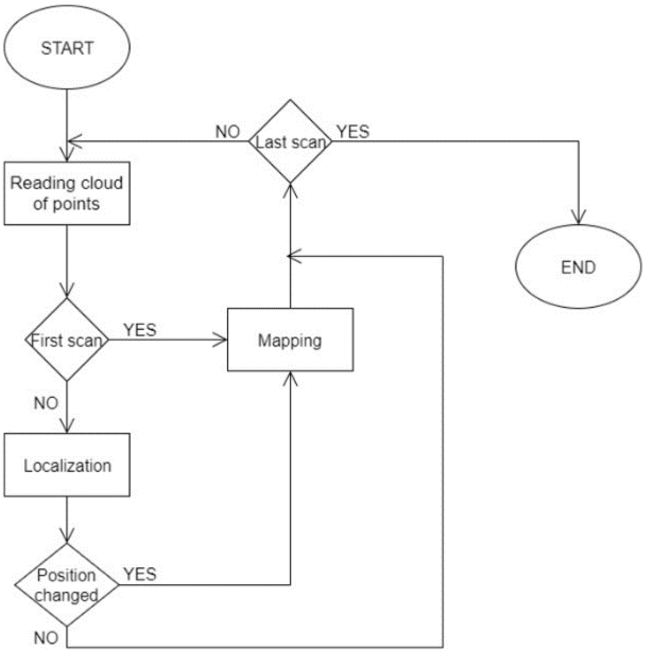

Fig. 3. Flowchart of localisation and mapping method

Fig. 4 shows the location and orientation of the laser range finder relative to the global coordinate system. The map of environment shows the points at which obstacles in the global coordinate system were detected. The transfer of measured points to the global coordinate system is based on the formula:

$$\begin{bmatrix} {}^{m}X_i \\ {}^{m}Y_i \end{bmatrix} = \begin{bmatrix} \cos(\varphi_r) & -\sin(\varphi_r) \\ \sin(\varphi_r) & \cos(\varphi_r) \end{bmatrix} \begin{bmatrix} {}^{s}X_i \\ {}^{s}Y_i \end{bmatrix} + \begin{bmatrix} x_r \\ y_r \end{bmatrix} \tag{1}$$

where x_r, y_r, φ_r – are coordinates informing about the linear and angular position of the laser range finder relative to the global coordinate system; P_i – i-th point of the scan; ${}^{s}X_i$, ${}^{s}Y_i$ – the coordinates of the i-th point in the coordinate system associated with the laser range finder; and ${}^{m}X_i$, ${}^{m}Y_i$ – the coordinate of the i-th point in the global coordinate system.

The location system searches for relevant shifting and rotation changes to minimise the function *f(z)*, which is shown below:

$$f(z) = 1 - \frac{Data_{fit}(z)}{Data_{valid}} \tag{2}$$

where $z = [\delta_x, \delta_y, \delta_\varphi]^T$ is shift and rotation of the sensor coordinate system relative to the coordinate system in the previous measurement cycle, which is elucidated in Nakamura et al. (2014) and Janah et al. (2018).

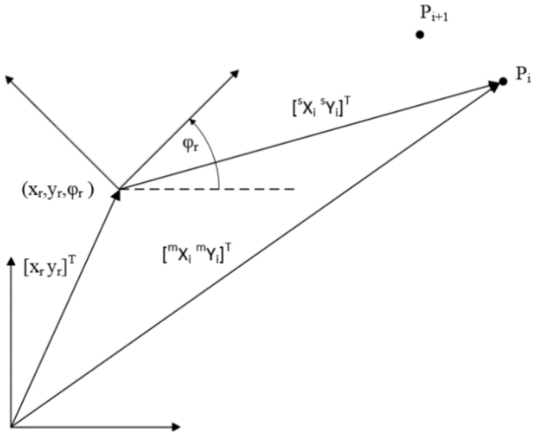

Fig. 4. Coordinate system

Data_{valid} is the number of points collected in the current scan that are suitable for determining the location. Points are rejected whose distance from laser range finder is less than 15 cm, and those that are the result of the mixed pixels effect.

$$Data_{fit}(z) = \sum_{i=1}^{N} g(i) \tag{3}$$

$Data_{fit}(z)$ is the number of points that were taken into account when calculating *$Data_{valid}$* and coincide with the map. The *g(i)* function assumes a value of 0 if the i-th point is not reflected on the map and 1 in the other cases. The affiliation of a given point to the map is checked after the earlier appropriate rotation and moving this point to the global coordinate system.

Minimising the *f(z)* function consists in searching for such z, for which this function assumes the smallest value. Ideally, this means finding a relative shift and rotation for which *f(z)* assumes a value of 0. This would mean that all points taken into account are reflected in the environment map.

Fig. 5 shows how the search algorithm for relative displacement and rotation works. The relative displacement in the coordinate axes is limited on the range from −5 cm to 5 cm every 1 cm (map resolution) and relative rotation from −5° to 5° every 0.2° (appropriate relative displacement is shown as gray cells, current position of the sensor as a black cell). From the given area, the z for which the *f(z)* function has the smallest value is selected.

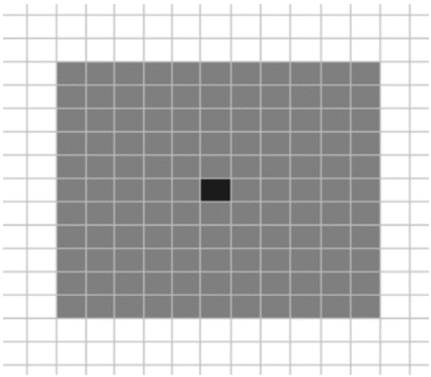

Fig. 5. Search area around the sensor position in the previous cycle

4. OBSTACLES DETECTION

In order for the robot to get from the start point to the target point, it must detect objects that stand in its way and prevent it from travelling between these points in a straight line. To this end, the so-called occupancy matrix determines the places where, due to the existence of obstacles, the robot is not able to physically find itself (as the robot is understood as the centre of the sensor in the plan view).

For ease of use, the area where the robot will move will be limited to the surface of a square with 2 m × 2 m sides. The occupation matrix will take the form of a 200 × 200 array, each element corresponding to a square with sides of 1 cm × 1 cm. The element assumes a value of 1 if the robot cannot be found in the given square with sides of 1 cm × 1 cm. Fig. 6 shows how the cells that the surface is divided into are assigned to the occupancy matrix *T*.

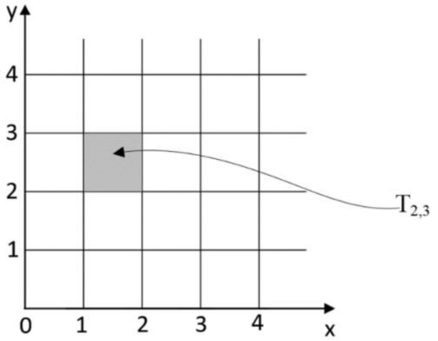

Fig. 6. Division of operational environment into elementary cells

The obstacle detection and avoidance method assume a safe area around each obstacle (larger than the nominal dimensions of the object). Fig. 7 presents the way of representing obstacles and the obstacle-free area. Obstacles are shown in black colour. The gray area indicates the area where the robot cannot be found due to its dimensions – safety area (zone). The area in which the robot can move is marked in white.

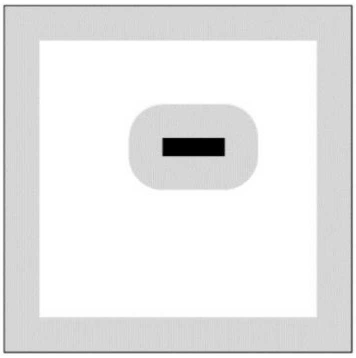

Fig. 7. Obstacle and safety zone scheme

5. PATH PLANNING

Robot path planning is an important element of the mobile robot navigation process. In this case, Rapidly Exploring Random Trees (RRT) algorithm was used due to the fact that it can be used as an online planning method and does not need large computing resources. Furthermore, RRT method is quite simple approach to planning paths through an obstacle field from a start node to an end node.

Fig. 8 shows the principle and the result of track search using the RRT algorithm. A graph is created to which nodes and edges are added randomly. A random q_{rand} point is created. The graph is searched for the node closest to the q_{rand} point. This node is called q_{near}. An episode with q_{rand} and q_{near} ends is created. On this segment, q_{new} is placed at a distance equal to the length of the step (if the segment length is less than the length of the step, q_{rand} becomes the point of q_{new}) from the point q_{near}. A new node is added to the graph, which is the point q_{new} and a new edge, i.e. a segment with ends at points q_{near} and q_{new}.

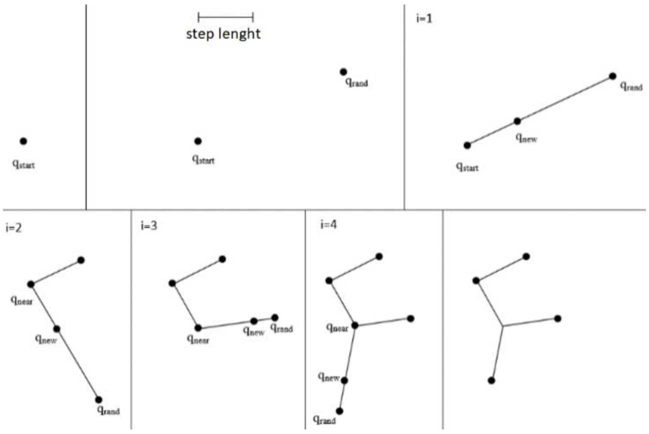

Fig. 8. Obstacle and safety zone scheme

6. MOBILE PLATFORM

A mobile robot platform specially prepared for this purpose was used to conduct the experimental tests of mapping, localisation, obstacle detection, avoidance and path planning. It is a four-wheeled robot with each wheel separately driven by a DC motor (Fig. 9). Beagle Bone Black computer on module (COM) was used as a main computer responsible for all navigational tasks. COM collects data from scanning laser rangefinder, which was mounted on the top of the robot. The UTM-30LX laser scanner communicates with the computer via a USB port. This causes many difficulties related to the implementation of communication. The robot is using the Linux Debian 7.5 operating system. Libraries responsible for map building, location, obstacle avoidance and trajectory planning were developed.

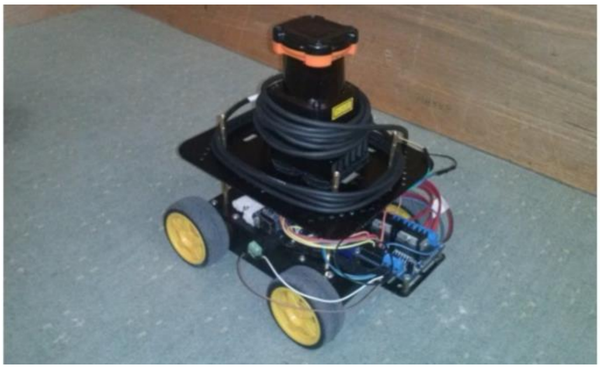

Fig. 9. Mobile platform used in the experimental tests

7. TEST'S RESULTS

The mobile robot's navigation task was divided into three parts: mapping and localisation, obstacle detection and path planning. A special, structured test environment has been developed for the experimental verification of the algorithms (Fig. 10). The prepared environment was specially configured to be simplified and in the first stage of the algorithm, sufficient verification was carried out to check all parts of the navigation system. Scheme of the objects' placement located in the experimental environment is presented in the Fig. 11. The scheme shows the environment in which the localisation algorithm was tested. The black point at the bottom of the picture indicates the place where the sensor was placed when the program implementing this location method started.

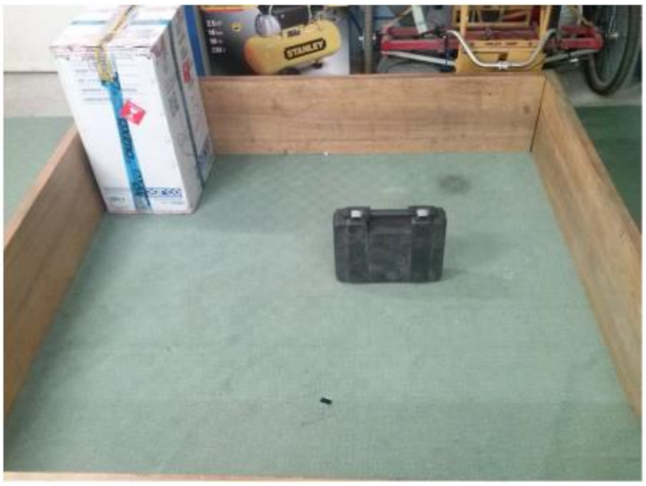

Fig. 10. Experimental environment

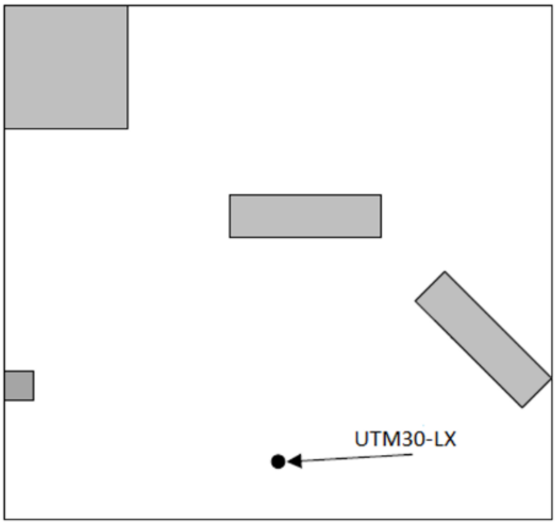

Fig. 11. Placement of objects in experimental environment

Figs 12–14 show how important it is to process data from the sensor. Using the data visible on Fig. 12, ascertainment of the correct location is possible but the map would be updated with incorrect points, which with each step would have an increasing impact on the correctness of location determination and would completely prevent the correct operation of the other systems, such as setting a path to avoid obstacles.

DOI 10.2478/ama-2021-0004

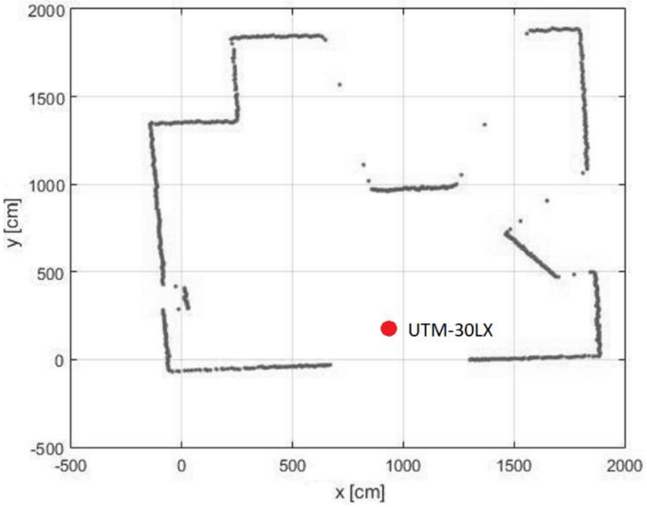

Fig. 12. Cloud of points from the raw data (Gao et al., 2019)

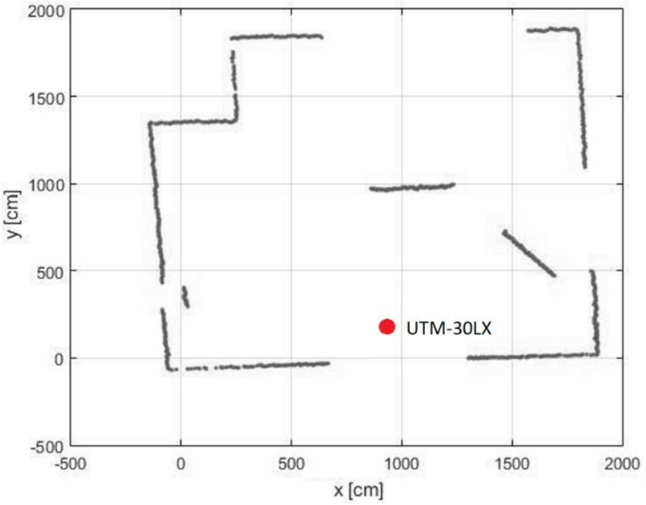

Fig. 13. Cloud of points after data filtration (Gao et al., 2019)

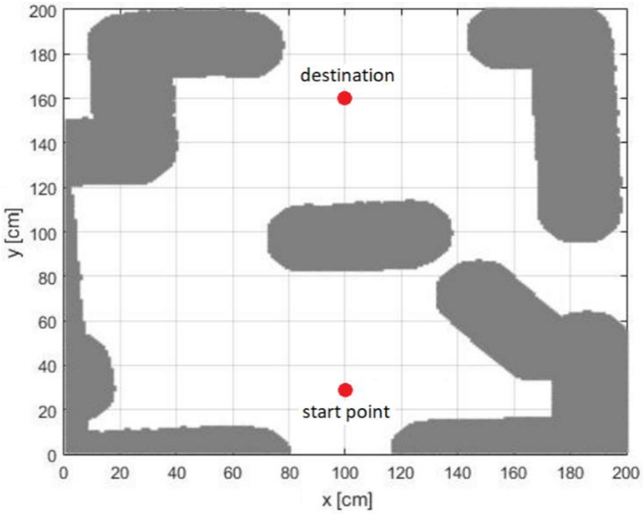

Fig. 14. Occupancy grid generated with using prepared map of the environment

After removing the error points (filtration was presented in Fig. 13), an occupancy matrix was created. Fig. 15 presents this matrix in a graphic form. The area in which the robot cannot be found is highlighted in gray. The robot can move freely in the white area.

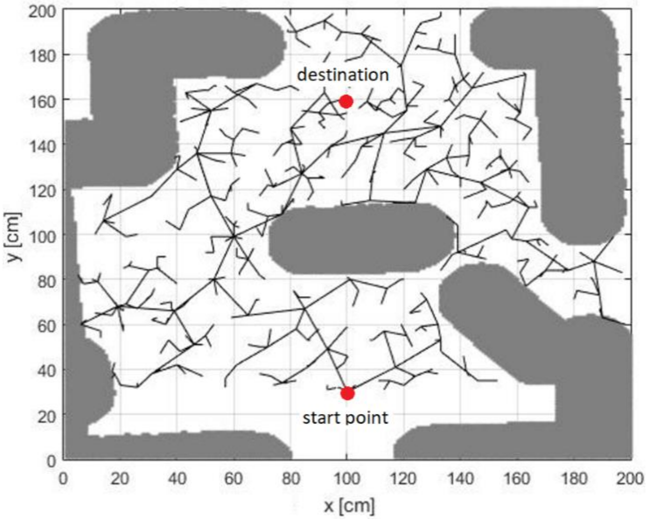

Fig. 15. Generated graph with RRT path planning method. RRT - rapidly exploring random trees

It is very clearly visible in Fig. 14 that some areas within which the robot cannot move are not included in the occupancy matrix. This is due to the fact that the sensor does not collect distance from objects behind other objects. Therefore, when the mobile robot performs the obstacle avoidance task, this matrix must be updated, and the RRT graph and path mapping ought to be created again.

To create the path that the robot should follow so that it can get from the start point to the target point, the RRT algorithm described was used. Fig. 15 shows the results of this algorithm for 500 iterations. The inaccessible area is marked in gray. The graph is black. The area of graph filling was limited from below by the coordinate of the position of the sensor. The graph fills the white area with the edges without crossing the gray area. The edges are therefore sections on which the robot can move without interfering with other objects.

After searching the graph and applying the smoothing and optimisation algorithm, the desired robot path was obtained and presented in Fig. 16.

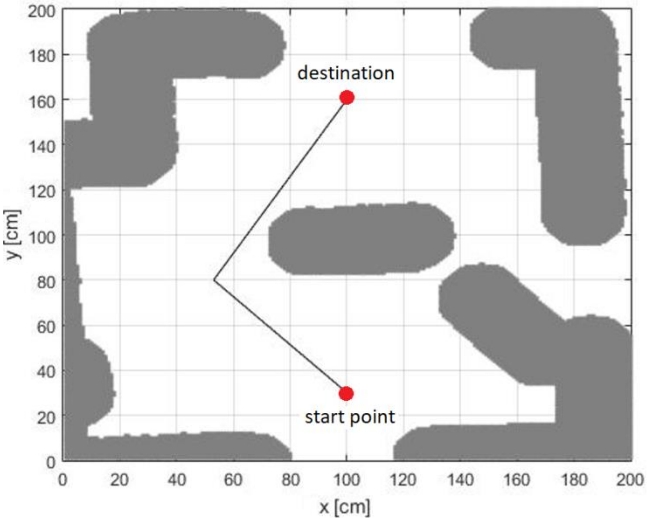

Fig. 16. Found and optimised path

In the original implementation, the sensor location was determined every few seconds. Modification of the method was based on using every 20-th point of the scan to determine the location and updating the map with all points. This modification caused a significant increase in the speed of algorithm (200 iterations of the algorithm were made in 36.7 s).

8. CONCLUSIONS

The paper addresses the problem of SLAM without any additional odometry measurements sensor for movements' calculations. Only laser scanner rangefinder data were used to build the map and locate the robot. Whole navigation system of the robot includes mapping and localisation subsystem, obstacle detection and avoidance system based on a prepared map and path planning algorithm using RRT method. The next step should be testing of the system in an unstructured environment, which is crowded with more objects.

REFERENCES

1. **Ambroziak L., Gosiewski Z.** (2015), Two stage switching control for autonomous formation flight of unmanned aerial vehicles, *Aerospace Science and Technology*, Vol. 46, 221–226.
2. **Ambroziak L., Simha A., Pawluszewicz E., Kotta Ü., Bożko A., Kondratiuk M.** (2019), Motor Failure Tolerant Control System With Self Diagnostics for Unmanned Multirotors, *24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Międzyzdroje, Poland, 422–427.
3. **Bekkali A., Sanson H., Matsumoto M.** (2007), RFID indoor positioning based on probabilistic RFID map and Kalman filtering, *Proc. of the 3rd IEEE International Conference on Wireless and Mobile Computing Networking and Communications*, 21.
4. **Chen X, Zhang H, Lu H., Xiao J., Qiu Q., Li Y.** (2017), Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue, *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, Shanghai, 41–47.
5. **Dellaert F.** (2005): *Square root SAM, Robotics Sci. Syst.*, ed. by S. Thrun, G. Sukhatme, S. Schaal, O. Brock, MIT Press, Cambridge.
6. **Dissanayake G., Durrant-Whyte H., Bailey T.** (2000), A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem, *Proc. of the IEEE International Conference on Robotics and Automation (ICRA) - Millennium Conference, Symposia Proceedings (Cat. No.00CH37065)*, Vol. 2, San Francisco, CA, USA, 1009–1014.
7. **Dissanayake G., Newman P., Durrant-Whyte H. F., Clark S., Csobra M.** (1999), An experimental and theoretical investigation into simultaneous localisation and map building (SLAM), *Proc. of the 6th International Symposium on Experimental Robotics*, March, 171–180.
8. **Dubbelman G., Browning B.** (2015), COP-SLAM: Closed-Form Online Pose-Chain Optimization for Visual SLAM, *IEEE Transactions on Robotics*, Vol. 31, No. 5, Oct., 1194–1213.
9. **Gao M., Tang J., Yang Y., He Z., Zeng Y.** (2019) An Obstacle Detection and Avoidance System for Mobile Robot with a Laser Radar, *16th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Banff, AB, Canada, 63–68.
10. **Guivant J., Nebot E.** (2002), Improving computational and memory requirements of simultaneous localization and map building algorithms, *Proc. of the IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Vol. 3, Washington, DC, USA, 2731–2736.
11. **Janah M., Fujimoto Y.** (2018), Performance Analysis of an Indoor Localization and Mapping System Using 2D Laser Range Finder Sensor, *44th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 5463–5468.
12. **Klecka J., Horak K., Novacek P., Davidek D.** (2016), Non-odometry SLAM and Effect of Feature Space Parametrization on its Covariance Convergence, *IFAC-PapersOnLine*, Vol. 49, Issue 25, 139–144.
13. **Kownacki, C., Ambroziak, L.** (2019), Adaptation Mechanism of Asymmetrical Potential Field Improving Precision of Position Tracking in the Case of Nonholonomic UAVs, *Robotica* (DOI: 10.1017/S0263574719000286), Vol. 37, No.10, 1823–1834.
14. **Lanzon A., Freddi A., Longhi S.** (2014), Flight control of a quadrotor vehicle subsequent rotor failure, Jo*urnal of Guidance Control and Dynamics*, Vol. 37, No. 2, 580–591.
15. **Li H., Chen Q.** (2010), Towards a non-probabilistic approach to hybrid geometry-topological SLAM, *8th World Congress on Intelligent Control and Automation*, Jinan, 1045–1050.
16. **Li P., Ke Z.** (2019), Feature-based SLAM for Dense Mapping, International *Conference on Advanced Mechatronic Systems (ICAMechS)*, Kusatsu, Shiga, Japan, 372–377.
17. **Lu F., Milios E.** (1997), Globally consistent range scan alignment for environmental mapping, *Auton. Robots* 4, 333–349.
18. **Moutarlier P., Chatila R.** (1989), Stochastic multisensory data fusion for mobile robot location and environment modeling, *5th Int. Symp. Robotics Res. (ISRR)*, 207–216.
19. **Nakamura Y., Fujimoto Y.** (2014), Validation of SLAM without odometry in outdoor environment, *13th IEEE International Workshop on Advanced Motion Control (AMC)*, Yokohama, 278–283.
20. **Romaniuk S., Ambroziak L., Gosiewski Z., Isto P.** (2016), Real time localization system with Extended Kalman Filter for indoor applications, *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland, 42–47.
21. **Rulin H.** (2017), *Research on Key Technologies of Dynamic Obstacle Avoidance in Driverless Vehicles*, University of Science and Technology of China.
22. **Smith R., Self M., Cheeseman P.** (1990), *Estimating uncertain spatial relationships in robotics, Autonomous Robot Vehicles*, ed. by I.J. Cox, G.T. Wilfong, Springer Verlag, Berlin, Heidelberg, 167–193.
23. **Soragna A., Baldini M., Joho D., Kümmerle R., Grisetti G.** (2019), Active SLAM using Connectivity Graphs as Priors, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 340–346.
24. **Xiaolin W., Jing Y., Fengchi S., Huan C., Shuzi H.** (2012), An approach to multi-robot cooperative SLAM, *Proc. of the 31st Chinese Control Conference*, Hefei, 4904–4909.
25. **Zhu D., Sun X., Wang L., Liu B., Ji K.** (2019), Mobile Robot SLAM Algorithm Based on Improved Firefly Particle Filter, *International Conference on Robots & Intelligent System (ICRIS)*, Haikou, China, 35–38.

Acknowledgements: The research work has been carried out within the framework of work No. WZ/WM-IIM/1/2019 and financed from the funds allocated for science by the Polish Ministry of Science and Higher Education.