LIDAR-based SLAM implementation using Kalman filter

Pawel Słowak*, Piotr Kaniewski#
Military University of Technology, ul. gen. Sylwestra Kaliskiego 2, 00-908 Warsaw, Poland

ABSTRACT

The capability to navigate accurately is one of the features, that a mobile robot should have to be able to perform tasks autonomously. In a GPS/GNSS-denied environment, for example inside buildings, localization of a mobile platform is an especially challenging problem. In such cases, to provide a robot with the ability to determine its position and to analyze its surroundings, Simultaneous Localization and Mapping (SLAM) algorithms could be implemented. In the article, we present a SLAM system that uses a Kalman filter together with data gathered by a 2D LiDAR. Our approach applies the ICP algorithm to calculate the localization and employs clustering and shape recognition technics to build the map of the environment. The article contains a detailed description of the individual elements of the proposed SLAM solution. Furthermore, it presents the results of experiments during which the system was validated.

Keywords: SLAM, LiDAR, Kalman Filter, clustering

1. INTRODUCTION

In the era of common access to devices that can determine their own location using satellite navigation receivers, there is an increasing demand for analogous capabilities in places where, for various reasons, an external signal providing data for localization algorithms is unavailable. The reason for the lack of availability is most often caused by interferences or a necessity of operating indoors. This issue is especially absorbing in terms of the analysis of mobile robots' autonomy. One way to obtain information about the location of a platform operating inside buildings is to equip it with a subsystem capable of implementing the procedure of simultaneous localization and mapping (SLAM). SLAM algorithms use data from one or several sensors and fuse it, according to known motion and observation models, in order to create the most probable hypothesis of the environment state together with the location and orientation of the robot. Among various devices that can be used to perform the task of simultaneous localization and mapping, a LiDAR is chosen frequently[1]. This sensor is capable of measuring the distance to obstacles in its surroundings, as well as determining the angle at which a given object is observed. Further processing of registered data (cloud of points) makes it possible to obtain a map of the area in which the mobile robot operates.

SLAM algorithms use a wide variety of probabilistic tools to synthesize information from sensors. The mathematical apparatus can be divided into three groups: parametric filters[2] (Kalman filter, extended Kalman filter[3], unscented Kalman filter), non-parametric filters (particle filter)[4] and optimization methods[5]. The idea of using a LiDAR as a main sensor for systems performing SLAM algorithms has been present for over two decades[6]. Among solutions based on the laser scanner, one can distinguish those in which various variants of the Kalman filter[7] are used to fuse the data, as well as those employing the particle filters[8] and the optimization methods[9]. The least complex variant of a LiDAR scan is a two-dimensional registration of the sensor's surroundings[10]. More complex systems, in which a three-dimensional point clouds are constructed to represent a map, could employ fully 3D scanners[11], sets of 2D LiDARs surveying environment in different planes[12] or a single 2D scanner that can change its orientation in space[13].

This article presents a SLAM system based on data recorded by a 2D LiDAR. To fuse information concerning the platform's location and its surroundings, a variant of Kalman filter together with an Iterated Closest Points (ICP) algorithm were applied. The appropriate map form was obtained with the use of clustering and shape identification algorithms.

*pawel.slowak@wat.edu.pl, phone +48 261 839 080; fax +48 261 837 461; www.wat.edu.pl; #piotr.kaniewski@wat.edu.pl, phone +48 261 839 080; fax +48 261 837 461; www.wat.edu.pl;

Radioelectronic Systems Conference 2019, edited by Piotr Kaniewski, Jan Matuszewski, Proc. of SPIE
Vol. 11442, 114420N · © 2020 SPIE · CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2564818

2. SLAM ALGORITHM IMPLEMENTATION

The proposed solution to the problem of simultaneous localization and mapping consists of several interrelated modules. The following chapter describes each of them, as well as characterizes the way in which the entire SLAM system was organized.

2.1 ICP algorithm

A point cloud $\mathbf{P}_i$ is a result of a single LiDAR scan. It contains a number of two-element vectors, which specify polar coordinates of points registered in the sensor's surroundings at a given time moment i. To determine the change of LiDAR's pose between two consecutive scans, the ICP algorithm was implemented[14]. Its task is to find an optimal linear transformation from $\mathbf{P}_1$ into $\mathbf{P}_2$ by minimizing the mean square error (MSE) of displacement between corresponding points in both point clouds.

First, the implemented algorithm calculates the cross-covariance matrix of point clouds $\mathbf{P}_1$ and $\mathbf{P}_2$. Then, the Singular Value Decomposition (SVD) of the matrix is performed in accordance with to the following formula:

$$Cov(\mathbf{P}_1, \mathbf{P}_2) = \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}}. \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}^{\mathrm{T}}$are orthogonal matrices and $\mathbf{W}$ is a diagonal matrix. As the polar decomposition theorem[15] states, the product of $\mathbf{U}$ and $\mathbf{V}^{\mathrm{T}}$ can be interpreted as a rotation matrix $\mathbf{R}$. $\mathbf{R}$ determines the difference in the orientation between both point clouds:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^{\mathrm{T}} \tag{2}$$

Knowing $\mathbf{R}$ and the geometrical centers $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ of each point cloud, one can determine the translation t between two consecutive LiDAR positions:

$$\mathbf{t} = \boldsymbol{\mu}_1 - \mathbf{R}\boldsymbol{\mu}_2. \tag{3}$$

The obtained $\mathbf{t}$ and $\mathbf{R}$ can be used to match both clouds and determine the change in position and orientation of the sensor between registrations. The result of 5 iterations of the implemented ICP algorithm is shown in Figure 1.

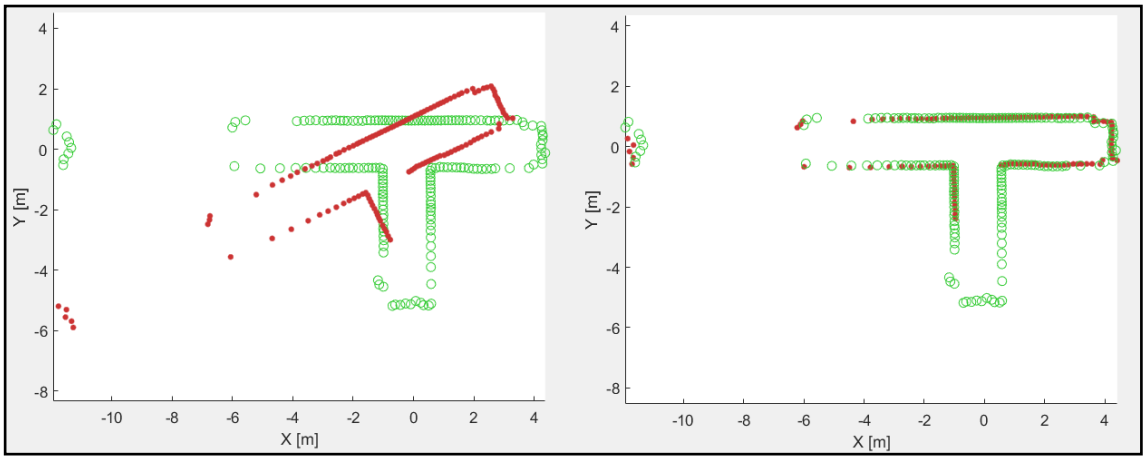

Figure 1. Left: two registered point clouds registered in consecutive scans. Right: both point clouds after ICP matching.

2.2 Clustering algorithm

To construct a map, the SLAM implementation proposed in this paper uses shapes recognition. Simple structures (lines and L-shapes) are searched for in data recorded by the laser scanner. The procedure of identifying shapes in a point cloud is facilitated by points clustering. The idea behind the operation of clustering scans is to divide registered points into disjoint sets. The conditions of belonging to a given set are defined by means of metrics, i.e. functions determining the relative distance between elements of the set[16]. This operation is used to preprocess information. Separating disjoint subgroups (clusters) enables to perform an analysis of each of them individually. Given that an appropriate metric was

implemented, resulting sets of points represent real objects, such as fragments of walls, doors, windows, etc. Hence, further data processing and map creation could be faster and less complex[16].

The basic metric used to cluster points, which are defined using cartesian coordinates system, is the Euclidean distance. If a separation between two consecutive points is smaller than a given threshold, they are assigned to the same cluster. However, this criterion is not sufficient to reliably cluster a point cloud. Spatial density of consecutive registrations depends on the angle at which a laser beam is reflected from a given surface. The more it deviates from the right angle, the greater are the relative distances between successive points and the smaller are clusters. Moreover, the range from the sensor is affecting the spatial density of a cloud as well. The range at which a measurement is taken is directly proportional to the density of points. To optimize the clustering process, a straight-line criterion was employed. As the clustering algorithm is going to work on the data collected indoors, numerous flat surfaces are expected to be scanned, i.e. walls, doors, etc. Therefore, a metric was introduced that forces the algorithm to group three consecutive points if the angle between them is close to π.

However, not all points identified as lying along a line should be grouped together. In some cases, the angle and range at which the elements of the environment are observed limits the capability to correctly identify the surfaces on which they are located. Therefore, a metric was implemented that prevents the addition of points to a given cluster if the range or angle of observation were different than the predefined accepted values.

The result of the clustering algorithm is shown in Figure 2. Each point cluster is drawn using a different color and the location of the LiDAR is marked with a red circle.

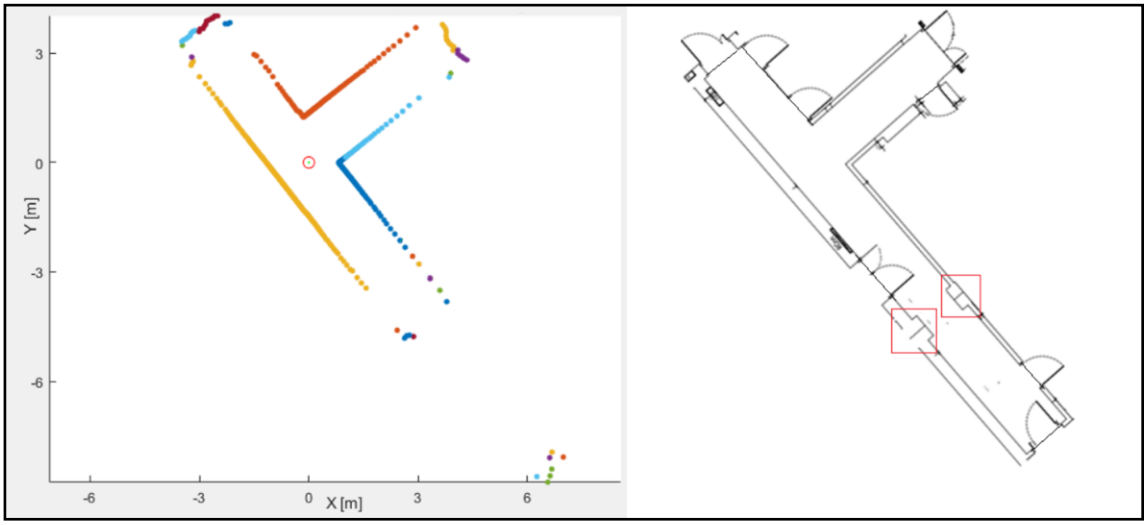

Figure 2. Left: The result of the clustering algorithm single run. Right: a plan view of a scanned corridor.

The right side of the Figure 2 shows a plan view of the scanned corridor. The wall protrusions that are not visible when analyzing the point cloud due to the small incidence angle (and the resulting insufficient density of measuring points) are marked in red.

2.3 Shape identification algorithm

After properly performed point cloud clustering, the SLAM algorithm proceeds to identification of shapes among obtained clusters. The algorithm detects straight lines and corners (L-shapes). If a given cluster cannot be assigned to any of these two categories, it is left unprocessed for further analysis. During latter algorithm iterations, it is combined with newly recorded points.

Shapes are identified in the following order. First, an analysis of the covariance matrix of coordinates of points in a cluster is performed. Provided that the value of one of the diagonal elements of the matrix is significantly larger than the value of the other element, the cluster is identified either as a horizontal line (variance in the x direction is relatively large), or a vertical line (variance in the y direction is relatively large). For clusters that do not meet the conditions described above,

principal component analysis (PCA) is performed. If one of the eigenvalues of the covariance matrix is significantly larger than the other, then the cluster is identified as a line as well. The operation of the algorithm described above is shown in Figure 3, where all the lines were correctly identified among the clusters in Figure 2.

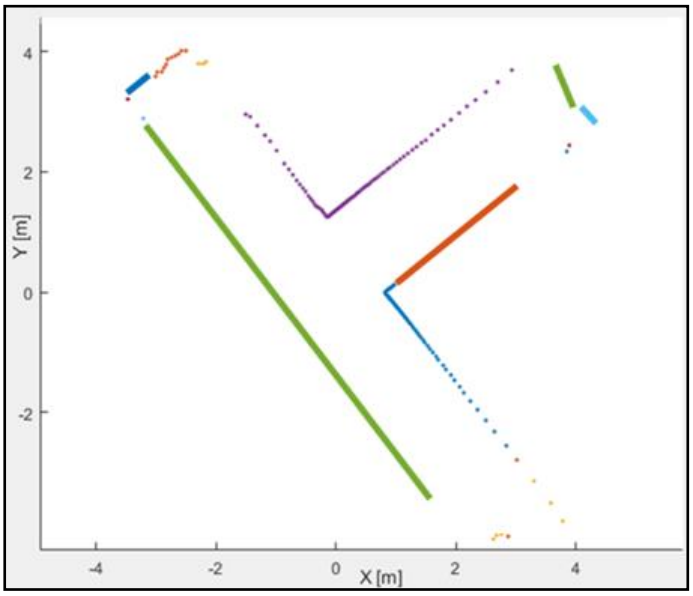

Figure 3. Identification of lines among point clusters.

If a given cluster fails to meet the previously described conditions, the algorithm proceeds to a more complex procedure. It checks whether a group of points can be classified as a corner (L-shape). In the first step, the shape of a rectangle is fitted into the cluster according to the SBRF (Search Based Rectangle Fitting) algorithm[17], which minimizes the expression:

$$\min_{0<\theta<\frac{\pi}{2},c_1,c_2} \sum_{i\in\mathbf{P}}(x_i \cos\theta + y_i \sin\theta - c_1)^2 + \sum_{i\in\mathbf{P}}(-x_i \sin\theta + y_i \cos\theta - c_2)^2 \quad (4)$$

where θ is the angle of rotation of the rectangle (for the value 0 one of the edges is parallel to the abscissa), x_i and y_i are the coordinates of subsequent points from the cluster, and c_1, c_2 are the coefficients that satisfy the equation of two perpendicular lines $x \cos\theta + y \sin\theta = c_1$ and $-x \sin\theta + y \cos\theta = c_2$. These lines are sides of the rectangle.

In the next step of the algorithm, three vertices of the rectangle, that describe the cluster most accurately, are selected. Then, four values are calculated:

- the area of the potential rectangle,
- the number of points in the analyzed cluster that lie within the triangle defined by the selected vertices of the rectangle,
- mean square error of the distance of the cluster points from two-line segments defined by the vertices of the rectangle,
- length of the shorter side of the rectangle.

If all the calculated values are within the experimentally determined limits, the shape is classified as a corner. If only the last condition is not met, the cluster can be identified as a corner or as a straight line. The final decision is made by comparing MSE of distance between a point cloud fragment and a closest line and corner.

If any of the first three values does not fall within the specified limits, the analyzed cluster cannot be identified as a line or L-shape. It is then further stored as a set of points. The results of operation of the above described algorithm are shown in Figure 4.

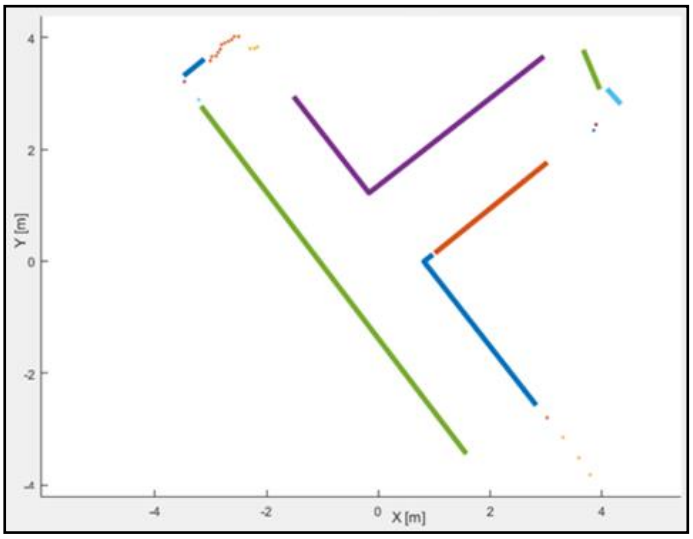

Figure 4. Identification of lines and L-shapes among point clusters

2.4 Algorithm implementation

A Kalman filter (KF) is an element of the implemented SLAM system that allows for indirect fusion of recorded data after each measurement. The first step of the KF is a prediction of the sensor state vector (position, orientation and speed). The adopted motion model assumes a constant velocity of the laser scanner between measurements. The relationships describing this process can be characterized by using the following formulas:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{\Phi}\hat{\mathbf{x}}_{k|k} \tag{5}$$

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}\mathbf{P}_{k|k}\mathbf{\Phi}^T + \mathbf{Q}_k \tag{6}$$

where $\hat{\mathbf{x}}_{k+1|k}$ is the predicted state vector, containing estimates of the platform's kinematic parameters, $\hat{\mathbf{x}}_{k|k}$ is the state vector estimated by the filter in the previous time step k, $\mathbf{\Phi}$ is the transition matrix, while the notation $k+1|k$ means that the prediction relates to the state at the moment $k+1$, based on the state from the moment k. The matrices $\mathbf{P}_{k+1|k}$ and $\mathbf{P}_{k|k}$ represent the covariance matrix of prediction and filtration errors respectively, whereas $\mathbf{Q}_k$ is the covariance matrix of the process disturbances, modeling random effects acting on the robot's motion.

A predicted sensor displacement, calculated as a difference between the currently predicted and previously estimated state vectors $\hat{\mathbf{x}}_{k+1|k} - \hat{\mathbf{x}}_{k|k}$, is used as the initial condition for the ICP algorithm. This is an important step as the ICP algorithm inherent property is to seek a local minimum of the displacement MSE. If the sensor position between consecutive registrations differs significantly, the algorithm may erroneously indicate the pose transformation, what could lead to an incorrect fusion of subsequent scans.

The translation and rotation calculated by the ICP algorithm are treated as a measurement vector in the update step of the Kalman filter. Update equations are presented below:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left[\mathbf{z}_{k+1} - h(\hat{\mathbf{x}}_{k+1|k})\right] \tag{7}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T \tag{8}$$

where $\mathbf{K}_{k+1}$ is the Kalman gain matrix, $\mathbf{z}_{k+1}$ is the measurement vector, h is the non-linear observation function, and $\mathbf{S}_{k+1}$ is the innovation covariance matrix.

After estimating the state vector, the map is updated. Firstly, the algorithm determines which points in the cloud were registered close to the already identified shapes. If the distance is below the threshold value, points and shapes are merged. Then, the procedures of clustering and shape identification is conducted for the remaining points. The last step of a map

update is to check which of the shapes can be merged with each other. For example, if two straight line segments lay close enough and their directions are similar – they can be represented as a single line segment.

3. RESULTS

During tests, several experiments were carried out. All data registrations were made using a RPLidar A2M8 scanner. It scans its surroundings in the 360º sector at the maximum range of 12 m. An omni-directional scan is produced by rotating the ranging core, where the laser beam source is mounted. The resulting measurement consists of a set of two-dimensional vectors $\mathbf{p}_i = [\theta, \rho]$ (point cloud), where θ is the azimuth angle expressed in degrees and ρ is the range.

3.1 Trajectory tracking

The first test was carried out to check the accuracy of a trajectory tracking. It was done by performing static, omni-directional laser scans at specific locations in a corridor. The subsequent positions of the scanner during the measurements corresponded to a rectilinear motion at a speed of approximately 5 m/s, with a scanning frequency of 10 Hz. As a result, point clouds were registered about every 0,5 m.

Figure 5 shows the result of the implemented algorithm without an active shape recognition module. The platform was moving in the positive direction of the *X*-axis. It can be seen that despite relatively large distances between subsequent registrations, the Kalman filter predicted the platform's kinematic parameters accurately enough, so that the ICP algorithm was able to calculate proper approximations of displacements. The analysis of the resulting map and the motion of the sensor shows that shapes and dimensions inside the corridor are correctly reproduced and the trajectory has been correctly determined as well.

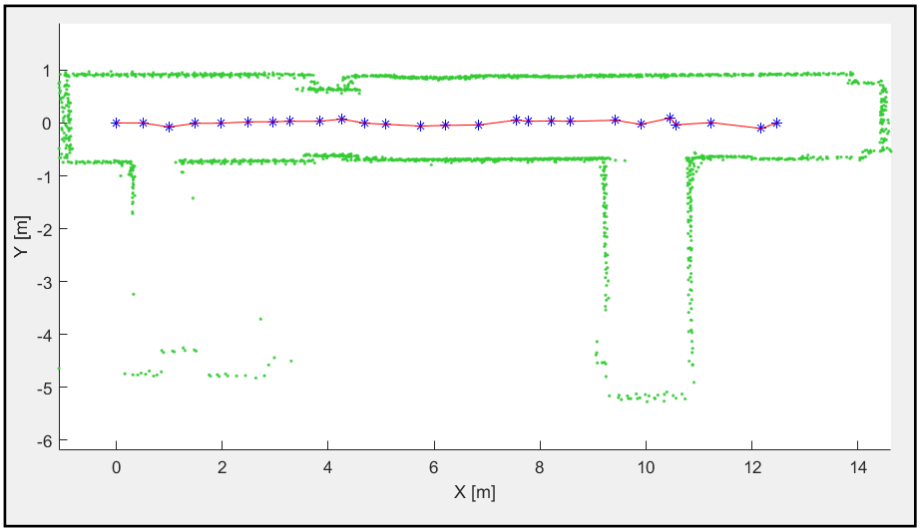

Figure 5. Map fused from 25 consecutive scans. Trajectory is marked with red, and the subsequent poses are marked with blue stars.

An appropriate adjustment of various parameters of different system components has a significant impact on the accuracy of the process of simultaneous location and mapping. For example, a forced reduction of cloud density is often desired to minimize the amount of registered data or limit the unevenness in the spatial distribution of points. Figure 6 illustrates how the modifications in scan data preprocessing scheme could affect the mapping accuracy. After changing the desired minimal distance between points form 0.01m to 0.02m, the algorithm was longer able to achieve expected accuracy. The analysis of the map and sensor trajectory shows, that with consecutive measurements errors increased. The last part of the trajectory does not indicate movement in a straight line, and the location of the walls becomes ambiguous.

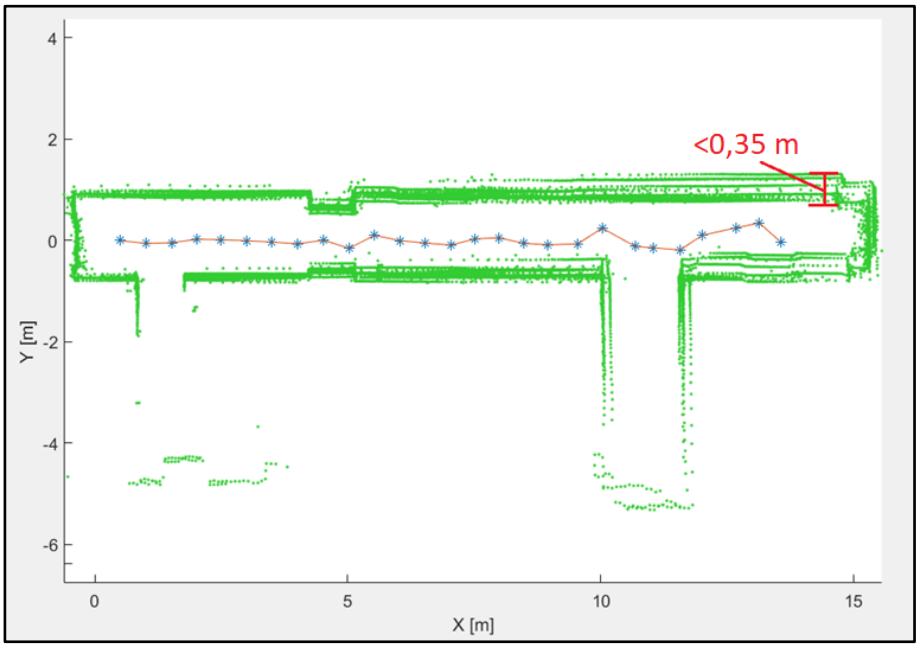

Figure 6. The map after changes in scan data preprocessing scheme.

Another experiment was aimed at verifying the importance of using the KF. The SLAM algorithm was run without the displacement prediction step. This way, the operation of ICP algorithm information about an initial guess was examined. The results of the experiment are illustrated in Figure 7.

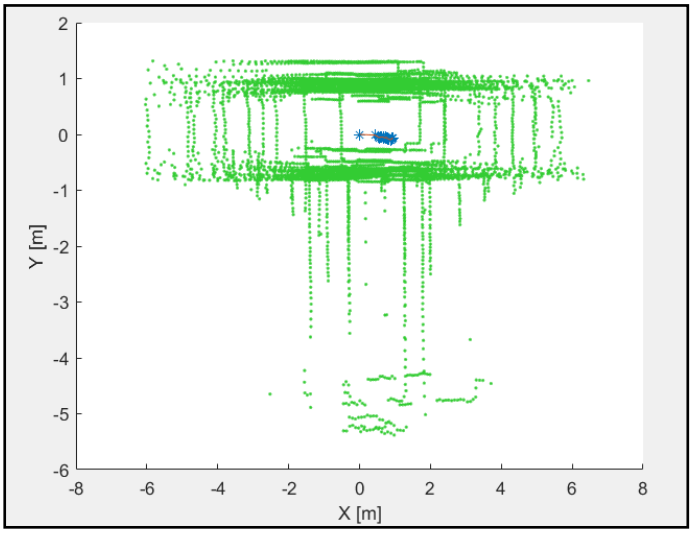

Figure 7. The resulting map without KF displacement prediction.

It seems conspicuous that the lack of displacement prediction precludes correct operation of the SLAM algorithm and that using the KF is necessary.

3.2 ICP and map fusion

The next experiment was conducted to test the clustering and shapes identification algorithms as well as the efficiency of the map update process after subsequent registrations. In the figures below, various stages of a map synthesis are presented.

Figure 8 shows how adjacent points are grouped into clusters and then turned into simple shapes. To increase a flexibility of the system the maximum size of point clusters was limited. Hence, there are visible differences of the adjusted clustering algorithm when comparing to the way the point cloud was processed in Figure 2.

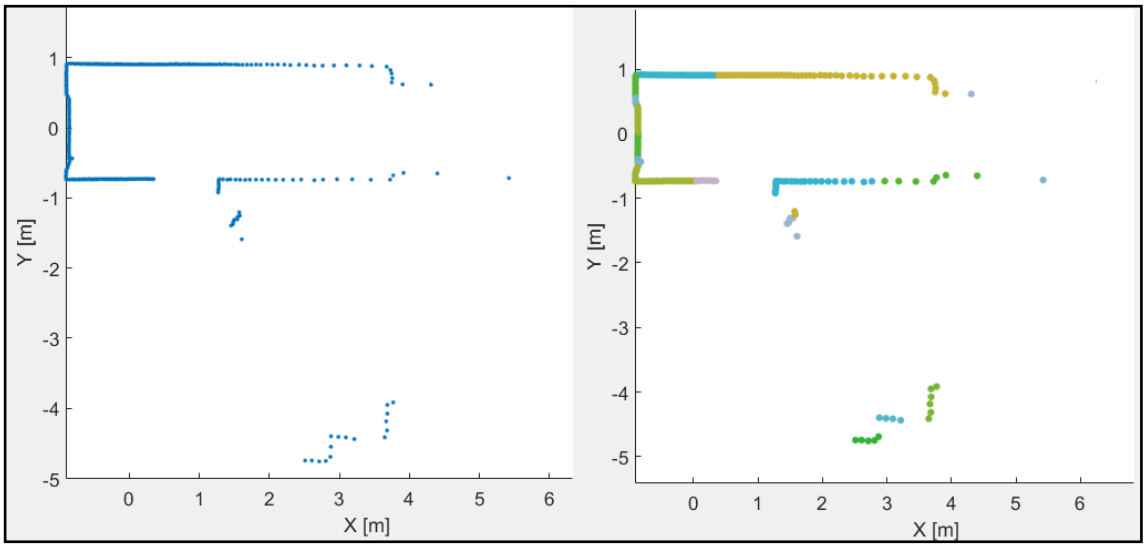

Figure 8. Points clustering with limited cluster size.

A potential merging into larger structures takes place after the shape identification step. In Figure 9, adjacent shapes are checked for collinearity and position on the plane. If conditions for the fusion are met, line segments are merged with line segments or L-shapes.

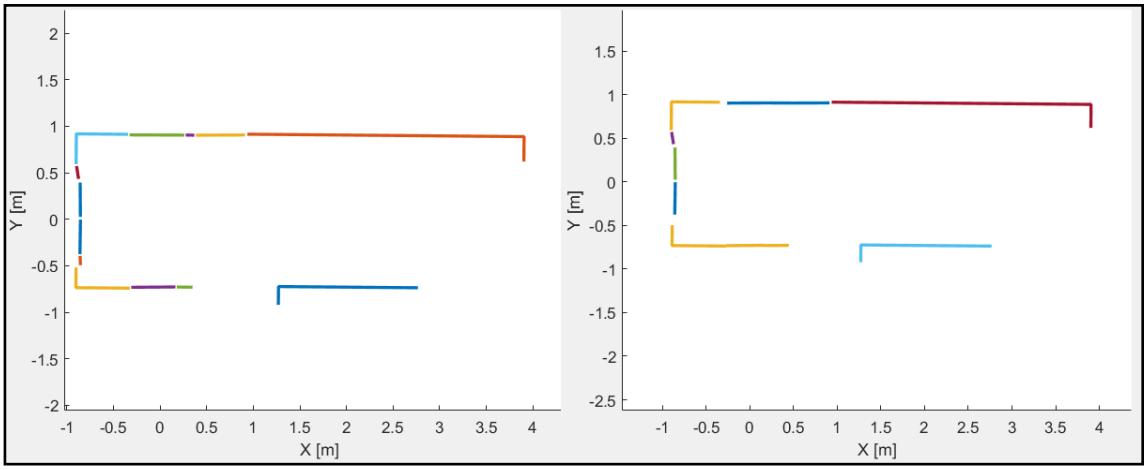

Figure 9. Shapes merging into larger structures.

In Figure 9, it can be noticed that not all of the small shapes were a part of the merging process. This is due to the fact, that the criteria which manage this process must minimize the risk of mismatching the shapes as it may result in erroneous decisions about navigation and mapping.

The result of mapping of the entire corridor during the dynamic experiment, where the sensor was registering consecutive scans while moving, is shown in Figure 10. It can be seen that only the last stage, i.e. scanning the right part of the corridor, has visible errors in the data association. For most of the trajectory, the algorithm for simultaneous location and mapping

works correctly. The shape of the building interior and the distances between the walls correspond to the real dimensions of the building.

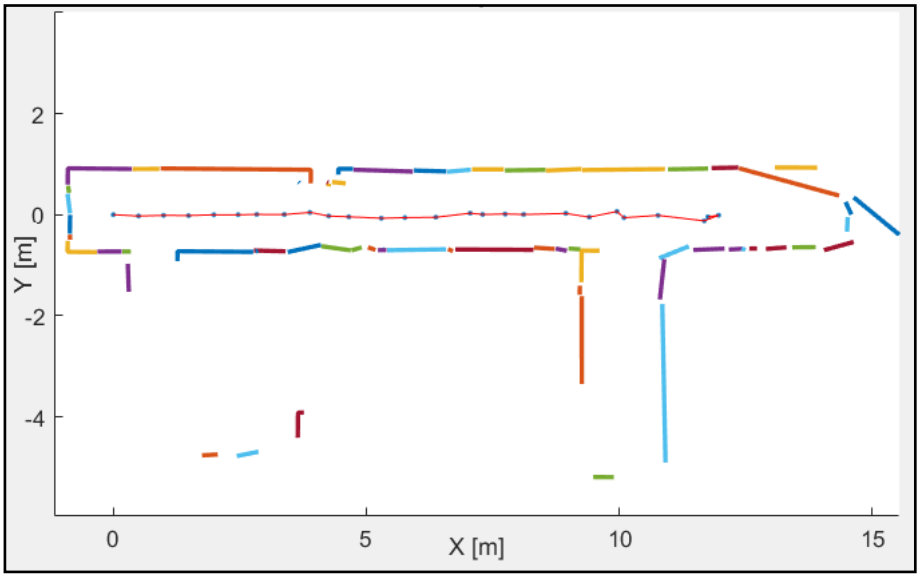

Figure 10. Map of the corridor and the trajectory of the platform.

4. CONCLUSIONS

The article presents a system performing the procedure of simultaneous location and mapping. Despite some visible inaccuracies, the presented results indicate that the adopted concept of solving the SLAM problem is correct. The degree of complexity of the system is significant, and the analysis of every algorithm step proves that robust and accurate operations require the experimental adjustment of many factors and threshold values.

The evaluation of the results and experience gained during the implementation of the adopted concept allow to enlist potential directions of the system development:

- modification of the ICP algorithm, e.g. by using the elements of the constructed map, which would increase the accuracy of estimating position and orientation changes,
- developing a concept for identifying and removing environmentally incorrectly added elements to the map, which would have a positive effect on the system's flexibility and increase its ability to operate without interruption,
- using information about occlusions, which would improve the accuracy of data association from subsequent measurements.

REFERENCES

[1] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. and Leonard, J. J., “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” IEEE Trans. Robot. 32(6), 1309–1332 (2016).

[2] Konatowski, S., “The development of nonlinear filtering algorithms,” Przeglad Elektrotechniczny, Vol. 86, Issue 9, 272-277 (2010).

[3] Solà, J., Vidal-Calleja, T., Civera, J. and Montiel, J. M. M., “Impact of landmark parametrization on monocular EKF-SLAM with points and lines,” Int. J. Comput. Vis. 97(3), 339–368 (2012).

[4] Kaniewski, P. and Słowak, P., “Simulation and Analysis of Particle Filter Based Slam System,” Annu. Navig. 25(1), 137–153 (2019).

[5] Mur-Artal, R., Montiel, J. M. M. and Tardos, J. D., “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” IEEE Trans. Robot. 31(5), 1147–1163 (2015).

[6] Castellanos, J. A., Martinez, J. M., Neira, J. and Tardos, J. D., “Simultaneous map building and localization for mobile robots: a multisensor fusion approach,” Proceedings. 1998 IEEE Int. Conf. Robot. Autom. (Cat. No.98CH36146) 2, 1244–1249, IEEE.

[7] Diosi, A. and Kleeman, L., “Laser scan matching in polar coordinates with application to SLAM,” 2005 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS, 1439–1444 (2005).

[8] Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J. and Nebot, E., “Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association,” J. Mach. Learn. Res. 4(3), 380–407 (2004).

[9] Droeschel, D. and Behnke, S., “Efficient continuous-time SLAM for 3D lidar-based online mapping,” Proc. - IEEE Int. Conf. Robot. Autom., 5000–5007 (2018).

[10] Durrant-Whyte, H. and Bailey, T., “Simultaneous localization and mapping: Part I,” IEEE Robot. Autom. Mag. 13(2), 99–108 (2006).

[11] Dube, R., Gawel, A., Sommer, H., Nieto, J., Siegwart, R. and Cadena, C., “An online multi-robot SLAM system for 3D LiDARs,” IEEE Int. Conf. Intell. Robot. Syst. 2017-Septe, 1004–1011 (2017).

[12] Vosselman, G., “Design of an indoor mapping system using three 2D laser scanners and 6 DOF SLAM,” ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. II–3(September), 173–179 (2014).

[13] Chong, Z. J., Qin, B., Bandyopadhyay, T., Ang, M. H., Frazzoli, E. and Rus, D., “Synthetic 2D LIDAR for precise vehicle localization in 3D urban environment,” Proc. - IEEE Int. Conf. Robot. Autom., 1554–1559 (2013).

[14] Segal, A. V, Haehnel, D. and Thrun, S., “Generalized-ICP (probailistic ICP tutorial),” Robot. Sci. Syst. 2, 435 (2009).

[15] Wójcik, P., “A Simple Proof of the Polar Decomposition Theorem,” Ann. Math. Silesianae 31(1), 165–171 (2016).

[16] Jain, A. K., Murty, M. N. and Flynn, P. J., “Data clustering: a review,” ACM Comput. Surv. 31(3), 264–323 (1999).

[17] Zhang, X., Xu, W., Dong, C. and Dolan, J. M., “Efficient L-shape fitting for vehicle detection using laser scanners,” IEEE Intell. Veh. Symp. Proc., 54–59 (2017).