

The LIDAR Odometry in the SLAM

Vasilii Kirnos, Vladimir Antipov

Yaroslavl State University

Yaroslavl, Russia

v.kirnos@uniyar.ac.ru, valant777@gmail.com

Andrey Priorov, Vera Kokovkina

Yaroslavl State University

Yaroslavl, Russia

andcat@yandex.ru, thiefrus@yahoo.com

Abstract—This paper describes an algorithm that performs an contour analyzing of an environment with a single 2D Laser Imaging Detection and Ranging (LIDAR) sensor, as well as its implementation on a mobile platform using the Robot Operating System (ROS). The review of standard sensors shortcomings is provided in article. It is offered decisions on the creation of the space circuit which is based on the result received from a the scanning lazer range finder. Simulation of system is made in the Gazebo environment.

I. INTRODUCTION

The map is usually built by hand. It means that, for exact localization, the position of the reference points (for example, walls, artificial beacons, etc.) which the robot uses for self-localizing has to be precisely measured and included in the map. Unfortunately, this approach can be difficult, expensive, and can a lot of time when the size of the environment is very large or when the environment changes because of artificial modifications or dynamic objects. For example, the internal robot vacuum cleaner is supposed to work in the indoor environments. In this case the robot has to be able to find changes in the map because of the furniture shifting. Another lack of handmade maps is that the prototype of the map can be differ depending on the various perception of who makes the map.

The alternative to handmade map building is therefore automatic map building. Indeed, a robot that localizes successfully has the right sensors for detecting the environment, and so the robot ought to build its own map. This ambition goes to the heart of autonomous mobile robotics. In prose, we can express our eventual goal as follows: starting from an arbitrary initial point, a mobile robot should be able to explore autonomously the environment with its on-board sensors, gain knowledge about it, interpret the scene, build an appropriate map, and localize itself relative to this map. The recent advances in both robotics and computer vision have made this goal somewhat achieved. An important subgoal has been the invention of techniques for place recognition and for autonomous creation and modification of an environmental map. Of course a mobile robots sensors have only a limited range, and so the robot must physically explore its environment to build such a map. So, the robot must not only create a map, but it must also do this while moving and localizing to explore the environment. In the robotics community, this is often called the Simultaneous Localization and Mapping (SLAM) problem. The relevance of the SLAM problem for the robotics community owes to the fact that the solution to this problem would make a robot truly autonomous. [1]

One of the most actively developing branches of robotics

are autonomous mobile systems. The exact assessment of the robot position in the space or the exact information on the environment is required at a movement control of an independent mobile robot or a group of robots. However in a large number of situations this information can be absent. In this case for the solution of a path planning problem in a performance of the independent robot mission first of all it is necessary to determine its location and to estimate a surrounding situation.

However for this purpose a robot needs to obtain the information of the world around and the data of its own position in it, for what the set of methods, including an odometry - the readings from various types of sensors to assessment of the object movement in space, are used. Now the following 3 approaches to movement assessment are most widespread:

- 1) using of encoder system;
- 2) using of inertial measuring devices;
- 3) visual odometry;

1) **Encoder usage.**

Encoder is established on the overwhelming majority of an terrestrial wheel and crawler robots for drives control. Therefore it usually doesnt require additional expenses for supplement of coordinates reading function from the information of a drives rotation angle. Then it is enough to add the kinematic model of the robot to the software to estimate its current situation. The robot kinematic model doesnt always corresponds to the reality. Besides the simple accumulation of a mistake caused by the coordinate reading method in addition the mistake at a wheel slipping appears. It is difficult to consider all factors in the kinematic model influencing on the real robot movement: diameter and deterioration of wheels, the air pressure squeezing of wheels, distribution of weight, friction force in all support points, etc. Also the kinematic model of the robot loses reliability at the movement on a soft surface. Generally the problem is caused by emergence of a course corner measurement error. Because of accumulation even of a small error of a course the robot significantly mistakes [2].

2) **Inertial measuring devices usage.**

The inertial measuring device is used for definition of the observer orientation in space. It consists of three microelectromechanical (MEMS) sensors: gyroscope, accelerometer and magnetometer. Inertial measuring devices serve well for the orientation assessment, but worse for the shift assessment [3].

3) Visual odometry

Visual odometry is a method of linear and angular shift assessment of the robot or other device by the analysis of image sequence shot at the camera established on it. The visual odometry belongs to the methods of the coordinate reading and, like the wheel odometry, measures increments of linear and angular coordinates. The current position of the robot towards the starting point is estimated as addition of these shifts for all the movement time. A method of a visual odometry is used, for example, in optical computer mouse, but such application areas as robotics, augmented reality and automobile systems dominate. But problems of the visual odometer accuracy assessment, its integration with data of other sensors and the real application as a part of a control system of robots, including for automatic return remain unprocessed [2].

The biggest interest is attracted to the methods of formation of movement trajectory based on the data received by LIDAR that is caused by its advantages, such as high speed and accuracy. The task lie in finding of an angle of the rotation, the size and the direction of a point cloud shift which describe the movements of the robot, according to a LIDAR. For this purpose it is necessary to combine the point cloud received at the moment time with the previous data set [4].

In this work we discuss benefits of using the contour analysis. At first it is presented step by step the theoretical base of the contour formation on the LIDAR data. After that we present our solution of an odometry algorithm.

II. METHODOLOGY

The algorithm on the base of contour analysis and RANSAC is consider in this work. The contour analysis is applied to data processing, received by LIDAR, and findings of features, and RANSAC for finding of size and the direction of features shift.

A. The current scan contour formation

For processing of the laser scan it is offered to use the contour analysis. The laser scan is exposed to contour coding, i.e. each value of the laser scan is puts in compliance to a certain number. The sequence of such numbers will be called a contour code. In this work the laser scan is submitted a polar code on the following formula:

$$C(n) = r(n) \cdot \cos(\alpha(n)) + i \cdot r(n) \cdot \sin(\alpha(n)), \quad (1)$$

where $r(n)$ is distance from a LIDAR to n - the purpose, and $\alpha(n)$ is the current angle of scanning

Further the polar code will be transformed to a differential code:

$$C(n) = \gamma(n+1) - \gamma(n) \quad (2)$$

B. The contour equalization

For the subsequent processing of a contour it is necessary to carry out a contour code equalization because the methods of the contour analysis mean the identical length of elementary vectors. The following method of an equalization is chosen: it is division into the predetermined quantity p of identical on length pieces, which ends are connected by vectors. These vectors are elementary vectors $\varepsilon(r)$ of a new equalization contour $E = \{\varepsilon(r)\}_{|0,p-1}$. Each elementary vector consists of three parts:

- 1) the rest $\Delta\gamma_{rest}^{(r)}(n)$ of a vector $\gamma(n)$
- 2) $(t-1)$ full elementary vectors $\gamma(n+1), \gamma(n+2), \dots, \gamma(n+t-1)$
- 3) the used part $\Delta\gamma_{used}(n)$ of an elementary vector $\Delta\gamma_{used}(n+t)$ of L-type contour

The length of pieces into which the L-type contour breaks is equal:

$$\varepsilon = \frac{1}{p} \sum_{n=0}^{k-1} |\gamma(n)| \quad (3)$$

On every r -th step of equalization is checked a condition in the beginning

$$|\Delta\gamma_{rest}^r(n)| \geq \varepsilon(1) \quad (4)$$

When this condition performing a part of an elementary vector $\varepsilon(r)$ doesn't enter L-type contour full elementary vector and the elementary vector is allocated from the rest $\Delta\gamma_{rest}^{(r)}(n)$ of an elementary vector $\gamma(n)$ of an initial L-type contour. Then

$$\varepsilon(r) = |\varepsilon| \frac{\Delta\gamma_{rest}^r(n)}{|\Delta\gamma_{rest}^r(n)|} \quad (5)$$

and the rest vector $\Delta\gamma_{rest}(n-1)$ for the following, $(r+1)$ -th an equalization step, is equal

$$\Delta\gamma_{rest}^{r+1}(n) = \Delta\gamma_{rest}^r(n) - \varepsilon(r) \quad (6)$$

If the condition (4) isn't satisfied, then the value at which the condition begins to be satisfied is checked

$$|\Delta\gamma_{rest}^r(n)| + \sum_{j=1}^t |\gamma(n+j)| \geq \varepsilon, t = 1, 2, \dots \quad (7)$$

Lets define it, further for receiving an elementary vector $\gamma(n+t)$ of a L-type contour as:

$$|\Delta\gamma_{used}(n+t)| = |\varepsilon| - |\Delta\gamma_{rest}^r(n)| - \sum_{j=1}^{t-1} |\gamma(n+j)|, \quad (8)$$

$$\Delta\gamma_{used}(n+t) = \gamma(n+t) \frac{\Delta\gamma_{used}(n+t)}{\gamma(n+t)} \quad (9)$$

In this case an elementary vector $\varepsilon(r)$ for r -th tep of an equalization and a residual vector $\Delta\gamma_{rest}^{r+1}(n+t)$ on $(r+1)$ -th step of an equalization will have an appearance

$$\varepsilon(r) = \Delta\gamma_{rest}^r(n) + \Delta\gamma_{used}(n+t) + \sum_{j=1}^{t-1} \gamma(n+j), \quad (10)$$

$$\Delta\gamma_{rest}^{r+1}(n+t) = \gamma(n+t) - \Delta\gamma_{used}(n+t) \quad (11)$$

C. The contour filtration

The filters matched with a form class are used for detection of key features on a contour and formations of a set of such features. It is supposed that there are areas on contours which are identical and it is possible to find affine transformation for this areas.

The matched filters provide formation of a quantitative measure of similarity between the filtered contour and a reference form. In this work the filters matched with a reference "corner" form are used, but further it is possible to apply also other reference forms. The filter selects a fragment which consists of two rectilinear pieces making a corner $\Delta\phi$. For this purpose the second rectilinear piece is exposed to turn on a corner $\Pi - \Delta\phi$, i.e. each elementary vector of this piece is multiplied on $-e^{-i\Delta\phi}$. As a result both sides of angle form one rectilinear piece. When the window C of the filter is equal $2s$, the result of filtration equal to the sum of half rated output effects of the filters matched with the parties of a figure of a corner has an appearance:

$$P(q, z) = \frac{\sum_{n=z}^{q-1} v(n+m)}{\sqrt{\sum_{n=z}^{q-1} |v(n+m)|^2}} \quad (12)$$

$$|\eta_n(m)| = \frac{1}{2\sqrt{s}} \left| P(s, 0) - P(2s, s) \right|, \quad (13)$$

where $\eta_n(m)$ is the module of an output rated signal of the filter, s is square of norm of a reference fragment, $v(n)$ is elementary a vector of a contour [5].

After filtration in each received fragment the local maximum of the module of an output rated signal of the filter is allocated. Each such local maximum is considered a special point.

D. Algorithm of unambiguous compliance establishment between contour points

After receiving special points it is required to establish one-to-one correspondence between points of contours. For establishment of one-to-one correspondence between points the distance between contours according to expression is accepted as a similarity measure:

$$R^2 = \|\Gamma\|^2 + \|\mathbf{N}\|^2 - 2\Re(\Gamma, \mathbf{N}), \quad (14)$$

where Γ is the current contour, \mathbf{N} is the previous contour. In both contours origin of counting are shifted in way that it coincided with the location of the considered special point.

For the purpose of productivity increase and also reduction of influence of the changed sites of a contour the measure of similarity counts only on local fragments of contours. Then for each couple of points the N pair of points received by a contour round on identical distance from initial couple is added.

E. Affine transformation matrix calculation

To estimate the size and the direction of the movement of the robot, it is necessary to find affine transformation between couples of points:

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \quad (15)$$

where x_i, y_i coordinates in old system, x'_i, y'_i - coordinates in new system, a, b, c, d, e, f coefficients of linear transformation on the plane.

Or

$$C'_i = H \times C_i, \quad (16)$$

where C_i - the initial position of an i point, C'_i - position of the point in new system, H a transformation matrix. Accepting coefficients of transformation as unknown sizes and writing down them for $n > 3$ points and after application of a method of the smallest squares we receive system which gives required coefficients of transformation [6].

$$T = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i & 0 & 0 & 0 \\ \sum x_i y_i & \sum y_i^2 & \sum y_i & 0 & 0 & 0 \\ \sum x_i & \sum y_i & \sum 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum x_i^2 & \sum x_i y_i & \sum x_i \\ 0 & 0 & 0 & \sum x_i y_i & \sum y_i^2 & \sum y_i \\ 0 & 0 & 0 & \sum x_i & \sum y_i & \sum 1 \end{bmatrix}, \quad (17)$$

$$T \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} \sum x_i x'_i \\ \sum y_i x'_i \\ \sum x'_i \\ \sum x_i y'_i \\ \sum y_i y'_i \\ \sum y'_i \end{bmatrix}, \quad (18)$$

F. An algorithm RANSAC application

In the received couples of points is available emissions (incorrect couples) which lead to the incorrect decision. One of widespread and successful methods which allow to eliminate emissions is the RANSAC method (Random Sample Consensus). The principle of work of an algorithm consists in iterative selection of the minimum subset of elements necessary for calculation of parameters of model, assessment of these parameters by means of some criterion and on completion of all iterations the choice of the best set of parameters and the subset corresponding to them [7], [8], [11].

III. ODOMETRY ALGORITHM

Odometry is the part of the SLAM. Short scheme of an algorithm:

- 1) On an entrance couples of points move
- 2) Selection of "trial" couples of points is made (fig. 1)
- 3) The model a hypothesis by method described in a step 6 is calculated
- 4) Assessment of compliance of features of model is calculated: on the basis of H the cloud is calculated $C'_i = H C_i$ and checking with C_i, C'_i to a hypothesis on the basis of assessment function $\|C_{i,t} - C'_{i,t}\| < t$

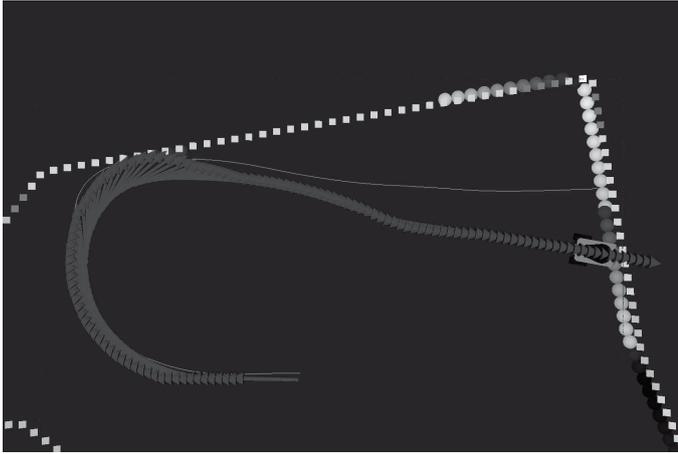


Fig. 1. Contur analyzer algorithm in processing

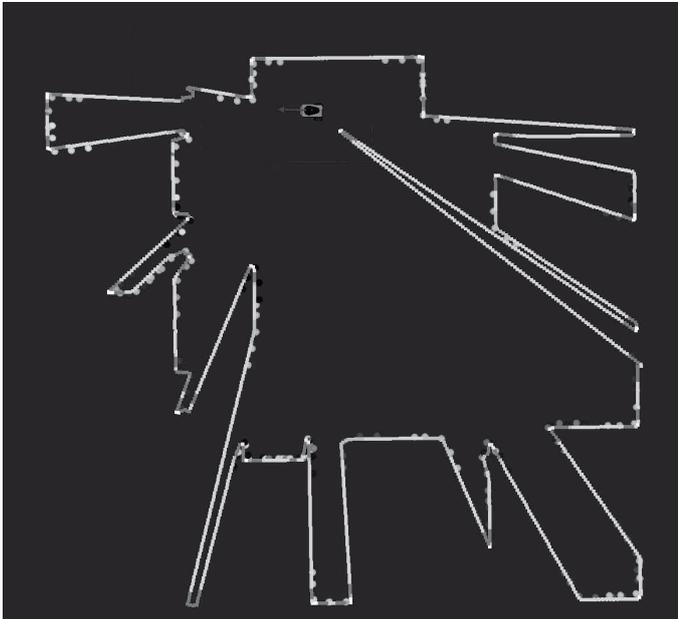


Fig. 2. The result of contur analyzer algorithm

where t is the defect, which is set by the user. As a result all couples are marked or "good":

$$c_{i,l} \in G_i, c'_{i,l} \in G'_i \quad (19)$$

or emissions:

$$c_{i,l} \notin G_i, c'_{i,l} \notin G'_i \quad (20)$$

- 5) On the basis of the number of couples in clouds G_i, G'_i and criterion estimates $\sigma = \frac{1}{n} \sum \|g_{i,l} - g'_{i,l}\|$ the hypothesis is marked with the best and the updated model on all couples G_i, G'_i is calculated, otherwise it is rejected.
- 6) We repeat points 2-5 k times, k the number of iterations set by the user. (fig. 2)

This algorithm is developed in the semi-natural modeling ROS complex (Robot Operating System). ROS is a meta-operating system for robots with an open source code. It isn't

TABLE I. RESULTS OF THE ROBOTS ARRIVALS ON DIFFERENT PATHS OF MOVEMENT

of arrival	S-type	L-type	Y-type	O-type
LIDAR rotation frequency is 5 Hz				
1	6.9%	13.8%	5.2%	5.3%
2	9.8%	9.3%	6.7%	5.8%
3	5.9%	14%	2.9%	3.5%
4	7.2%	13.1%	5.3%	5.9%
5	4.6%	13.8%	4.9%	7.7%
6	5.7%	19.6%	2.7%	7.5%
7	5.6%	13.7%	5.1%	3.7%
8	8.9%	12%	3.8%	2.3%
9	8.4%	17.5%	5.4%	6.3%
10	4.9%	14.4%	2%	5.8%
median	6.8%	14.1%	4.4%	5.4%
LIDAR rotation frequency is 2 Hz				
1	2.5%	10.5%	1%	1.1%
2	1.4%	9.5%	0.6%	1.1%
3	3.3%	14.7%	2.8%	1.3%
4	2.3%	16%	2%	2.2%
5	2.8%	10.7%	4.2%	2.9%
6	3.1%	18.8%	1.3%	3.3%
7	2.7%	10%	2.3%	5.7%
8	3.2%	15.4%	2%	2.8%
9	1.9%	12.3%	2.7%	1.7%
10	2%	13.1%	2.3%	2.8%
median	2.5%	13.1%	2.1%	2.6%
LIDAR rotation frequency is 1 Hz				
1	4.4%	5.3%	2.7%	1.5%
2	4.4%	12%	3.9%	2.7%
3	5.6%	4.4%	2.9%	2.8%
4	5%	10.6%	1.5%	2.2%
5	3.2%	11.8%	3.2%	2.8%
6	2.5%	1.6%	3.5%	2.4%
7	4.2%	7.9%	1.9%	1.8%
8	2.9%	5.7%	4%	2.8%
9	3.1%	12.9%	1.5%	2.2%
10	5%	6.2%	2.7%	2.4%
median	4%	7.8%	2.8%	2.4%

replaced an operating system, but it expands with a set of the modules that is necessary for management of robotic systems.

The experimental studies were provided in the virtual environment Gazebo (fig. 4). There are four types of the robot arrivals on the Willow Garage map: S-type, L-type, Y-type and O-type (fig. 3). The error of measurement of the robot coordinates that concerning trajectory length is measured in each arrival. The twist algorithm of movement is used here. It means that every measure is performed when the robot stops.

Results of the robot arrivals on various trajectories with different rotation frequency of the movement is presented on Table I.

The result of measurement of an average error of measurement of the robot coordinates concerning length of a trajectory of equal 2-3 % is considered good, i.e. the algorithm works approximately as well as an algorithm of a wheel odometry.

The result of measuring the average error of the robot coordinates relative to the trajectory length equal 2-4% is considered good, i.e. the algorithm works approximately like the wheel odometric algorithm.

It is also possible to observe the dependence of this error on the rotational rate of the LIDAR. At a LIDAR rotation rate of 5 Hz, the algorithm works worse. When the frequency of the LIDAR rotation decreases at the continuous movement of the robot, the distance passed by the robot increases to following measurements. It promotes reduction of the repeating area, the robustness of the algorithm decreases.

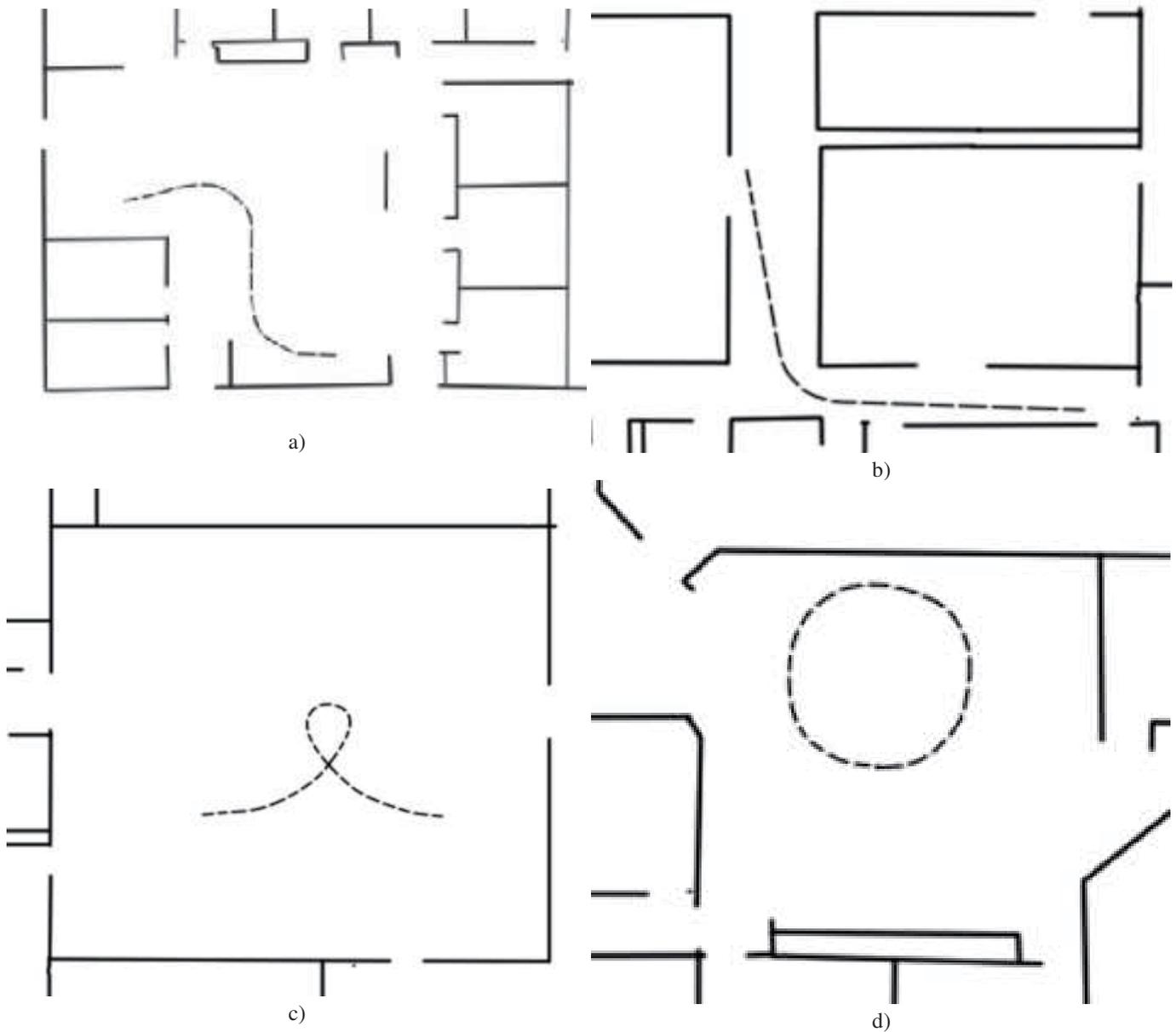


Fig. 3. Types of paths: a) S-type, b) L-type, c) Y-type, d) O-type

The algorithm is not worse than visual odometry, in which the error equals about 12% and sometimes can reach 20% [3], [10], [9]. The disadvantage of the algorithm is the presence of objects that have angles, near the robot.

IV. CONCLUSION

In this work, we discussed the advantages of using a LIDAR as a sensor to control the odometry of the robot. The algorithm accuracy is on same level as the others. It is planned to improve the algorithm, i.e. generalize it in way that the different reference forms in a matched filtering can be used.

The simulation model can't describe completely all the real environment noise distortions. It means that we ought to use some real equipment in the real. This solution will based on:

- 1) **YDLidar X4**
- 2) **2WD miniQ Robot Chassis with encoders**
- 3) **Arduino Mega**
- 4) **Raspberry PI 3 with ROS**

This robotics system is developed to collect the information from wheels and the LIDAR. The main data is calculated outside of the system, on the server under the ROS.

YDLidar X4 is LIDAR which is used in autonomous mobile vacuumcleaners. The measured length is 10 meters, it is enough for indoors localization and mapping.

The LIDAR model, that is used in Gazebo enviroment is not YDLidar X4. The virtual model is based on Hokuyo 720. The accuracy of these LIDARs are similiar.

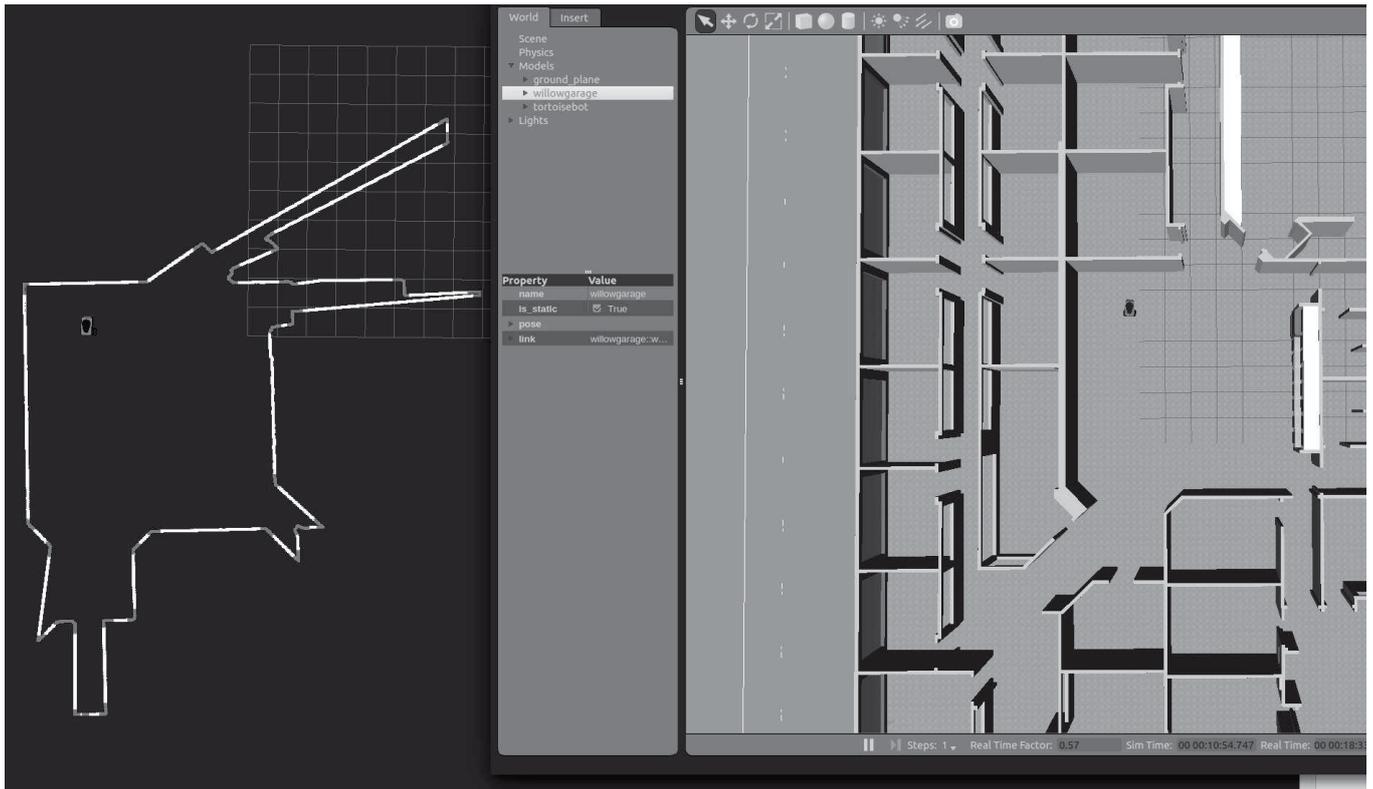


Fig. 4. Gazebo environment with the Willow Garage model

The real robot will be based on Arduino Mega. It means that a lot of low level sensors such as compass, accelerometer, gyroscope could be used for better robot localization.

The arduino is needed for getting data from wheels and encoders. For better resolution we will use **2WD miniQ Robot Chassis** and **MiniQ Robot chassis Encoders**

Raspberry PI is used as node of the ROS system. In the future we can use a lot of robots which are based on such architecture for the mapping process in SLAM task and this solution will be easier to use the ROS, instead of other protocols and operation systems.

There is also solution to use the accelerometer for high accuracy. It will help us to build full kinematic model of the robotics system.

ACKNOWLEDGMENT

We would like to show our gratitude to Yuriy Brjukhanov and Vladimir Volohov for sharing their pearls of wisdom with us during the course of this research.

Special thanks for the Iliya Trapeznikov and Alexandr Prozorov

REFERENCES

[1] Siegwart, Roland and Nourbakhsh, Illah R. and Scaramuzza, Davide, "Introduction to Autonomous Mobile Robots", 2nd, The MIT Press, 2011
 [2] Devyaterikov E. "The multisensor system of independent navigation for automatic resetting of the mobile robot with use of the visual odometer data", N.E. Baumanov MSTU, PhD work, 2016, 152 p.

[3] Mariano J.T. "Motion estimation, 3D reconstruction and navigation with range sensors", Munchen Technical University, Doctor work, 2017, 199 P.
 [4] Abramov A.I., Abramov I. V., Mazitov T.A., Palmov A.M. "Application of a bee algorithm for data handling of the laser scanning system in case of navigation of mobile robots" *Vestnik Kalashnikov M.T. ISTU*, ISSN 1813-7903, 2016 2(70), pp. 101-104
 [5] Furmanov Ya.A., Krevetkiy A.V., Peredreev A.K., Rodjenzov A.A., Hafizov R.G., Egoshin I.L., Leuhin A.N., "Introduction to the planimetric analysis; applications to image processing and signals", Phizmatlit, 2003, 592 p.
 [6] Degtyaryov A.M., Yaltihov V.V. "Identification of transformation model on the basis of affine transformation" *Automated technologies of researches and design*, 2(49), 2013, pp. 71-74
 [7] Mashentzev V.Y. "Navigation using RANSAC of filtering point features" *Graphicon*, Young scientists school, 2013, pp. 271-274.
 [8] Fernandez-Rodriguez S. "Recognition of 2D-objects using RANSAC", Shefiled Hallam University, 2008.
 [9] Efimov A.I. "Development and research of algorithms of images combination from onboard video sensors with the virtual model of terrain", PhD work, Moscow, 2016
 [10] Mariano J., Monroy J. G., Gonzalez-Jimenez J. "Planar odometry from a Radial Laser Scanner A range flow-basse approach" *IEEE International Conference on Robotics and Automation*, 2016, pp.
 [11] Fontanelli D., Ricciato L., Soatto S., "A fast Ransac-based registration algorithm for accurate localization in unknown environments using lidar measurements" *IEEE International Conference on Automation Science and Engineering*, 2007, pp. 155-168
 [12] Furmanov Ya., Yanshin V., "Extraction and Linear Filtering of Closed Polygonal Contours of Images" *Pattern Recognition and Image Analysis*, 1994, V.4 2, pp. 146-166
 [13] J. Wang and Y. Yagi, "Robust location recognition based on efficient feature integration," in Proceedings. *IEEE International Conference on Robotics and Biomimetics*, (ROBIO 2012). IEEE, 2012, pp. 97101.