Landmarks Detection by Contour Analysis in the Problem of 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 is based on an contur 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 solution is based on landmarks, its mean that all walls in the building could be discribed as a simple objects: angles and lines**

I. Introduction

SLAM is simultaneous localization and mapping. The problem of SLAM is connected with the construction of a map of unknown space by a mobile robot during navigation on this map. This problem underlies of many robot applications. Some approaches within the frame of it use light detection and ranging (LIDAR) because of its ability to accurately measure the distance to objects around it.

The SLAM task consists of the following main steps:

1) Searching landmarks in space;
2) Data association, i.e. matching of landmarks;
3) Evaluation of the robot and landmarks locations;
4) Updating of the robot and landmarks locations.

SLAM performs global optimization of measurements obtained by various sensors. Accordingly, obtaining measurements is one of the important subtasks included in the SLAM task. Pre-processing of measurement data is carried out by odometry methods. The mobile robot can determine its displacement in relation to the previous position using visual odometry methods or based on the analysis of rangefinder data.

There are different ways to realize certain subtasks within the SLAM task. Thus, it is possible to combine different implementations of individual algorithms and improve them individually.

Attention in this paper is focused on the subproblem of finding landmarks. Several algorithms based on contour analysis are compared: an algorithm based on the method of matched filtering [5] and different approaches developed for the detection of dominant points (the Teh-Chin algorithm [6, 7], the Wu algorithm [4], the method based on the scale-space theory [1, 2, 3])

The objective of this work is to compare different landmark detectors using contour analysis in the SLAM problem.

II. Landmarks searching

Landmark is a specific, clearly visible on the ground, nonmoving object, which can be used to navigate.

Requirements for landmarks:

1) Re-observed, i.e. viewed from different positions and angles;
2) Unique, i.e. distinguishable from each other;
3) Numerous in the environment;
4) Stationary.

It is proposed to use contour analysis to find landmarks. The idea of the approach is to refuse the processing of each point obtained from the laser rangefinder, and the transition to the processing of only the contour. The contour is represented as a complex-valued signal. This model allows to use more methods that are invariant to the transfer, rotation and scaling.[10].

Before using the landmark detector, the laser scanning system data are processed as follows:

1) Presentation of laser scanning system data in the polar code:
$$C(n) = r(n) \cdot cos(\alpha(n)) + i \cdot r(n) \cdot sin(\alpha(n)), \quad (1)$$
where $r(n)$ is distance from the laser scanning system to the object,d $\alpha(n)$ is the current scanning angle
2) Presentation of a polar code in a differential code:
$$C(n) = \gamma(n+1) - \gamma(n). \quad (2)$$
3) Equalization of the difference code.
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:
   a) the rest $\Delta\gamma^{(r)}_{rest}(n)$ of a vector $\gamma(n)$
   b) $(t-1)$ full elementary vectors $\gamma(n+1), \gamma(n+2), \ldots, \gamma(n+t-1)$
   c) 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

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

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

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

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

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

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

A. Landmarks searching using matched filtering

Filters matched with the shape class can be used to detect key features on the contour and to forming variety of these features

Matched filters provide a quantitative measure of similarity between the filtered contour and the reference shape. The filters matched with the reference form of the "angle" are applied here. The filter selects a fragment that consists of two straight-line segments that compose the angle $\Delta\phi$. For this purpose, the second straight-line segment is rotated by an angle $\pi - \Delta\phi$, i.e. each elementary vector of this segment is multiplied by $-e^{-i\Delta\phi}$. As a result, both sides of the angle form one straight-line segment. When the window C of the filter is $2s$, the result of filtering, equal to the sum of half normed output effects of filter, that matched with the sides of the angle shape, has the form:

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

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

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

B. Teh-Chin algorithm

It is also possible to use approaches that are designed to detect dominant points. Dominant points are points with high curvature. Dominant points are used in data compression problems and in highlighting features of the object contour.

Usually there are two main stages in the methods of detection of dominant points:

1) Curvature estimation;
2) Search for local maxima.

It is important to define a support area to estimate curvature. The support area for the curve point p_i of size k is the sequence $D(p_{i-k}, \ldots, p_{i-1}, p_i, \ldots, p_{i+k})$ (Fig. 1). If the length of the support area is large, then some dominant points are skipped; if the support area is small, then needless points appear. It is necessary to determine the support area for each point, because each point has its own local geometric features. One of the adaptive methods to find the support area is the Teh-Chin method. It uses the ratio of perpendicular d_{ik} and chord length l_{ik} :

$$r_{ik} = \frac{d_{ik}}{l_{ik}}. \quad (14)$$

The measure k the angle cosine is used as a curvature estimate:

$$cos_{ik} = \frac{\vec{a_{ik}} \cdot \vec{b_{ik}}}{|\vec{a_{ik}}||\vec{b_{ik}}|}, \quad (15)$$

where $\vec{a_{ik}} = (x_{i-k} - x_i, y_{i-k} - y_i)$, $\vec{b_{ik}} = (x_{i+k} - x_i, y_{i+k} - y_i)$.

Points with a local maximum curvature value are considered as dominant points.

Teh and Chin used the relationship of the perpendicular and the chord length to determine the region of support for each point.

The support area is searched by iterative increment starting from $k = 1$. Herewith k increases until $l_{i,k} \geq l_{i,k+1}$ one of the conditions is true, then k defines the length of the support area at a point p_i:

$r_{i,k} \geq r_{i,k+1}$ for $d_{i,k} > 0$, $r_{i,k} \leq r_{i,k+1}$ for $d_{i,k} < 0$.

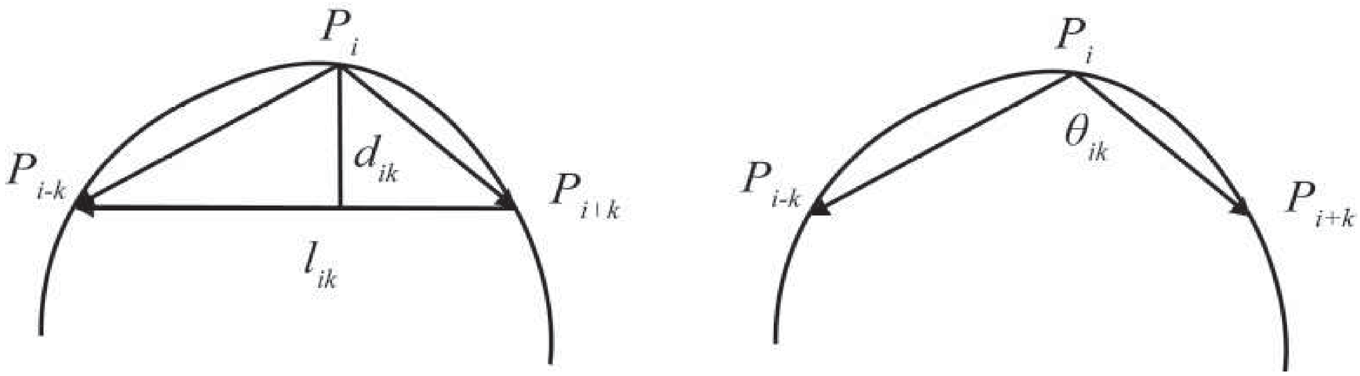


Fig. 1. Support area and angle in point p_i

As the curvature value is used directly value k, the angle cosine.

C. Wu algorithm

Teh-Chin method uses the first local maximum of the curvature value to determine the support area. However, such a local maximum can be caused by the noise presence on the curve, i.e. the Teh-Chin algorithm is not reliable in the noise presence. For solving this problem Wu proposed to choose a support area such as it has a global maximum curvature value, i.e. to determine the support area between K_{min} and K_{max}. Let k_i is the best length of support area at the point p_i. Then k_i can be defined as follows:

$k_i = k$, if $cos_{ik} = max\left\{cos_{ij} | j = K_{min}, \ldots, K_{max}\right\}$, for $i = 1, 2, 3, \ldots, n$.

Then the support area is the following area:

$$D_i = \left\{p_{i-k_i}, \ldots, p_{i+k_i}\right\}, \tag{16}$$

and the value of the curvature at the point p_i defined as the mean value k, the angle cosine:

$$cv_i = \frac{1}{k}\sum_{j=1}^{k_i} cos_{ij}. \tag{17}$$

D. The curvature scale space method

Dominant points can also be searched using the theory of the so-called curvature scale space. Let C is a curve, $C = (x(s), y(s))$, where $0 \leq s \leq L$, L is curve length. The curvature function can be presented as follows:

$$K(x, y) = \frac{\frac{\partial^2 y}{\partial x^2}}{\sqrt{\left[1 + (\frac{\partial y}{\partial x})^2\right]^3}}. \tag{18}$$

If it denote

$$\dot{x} = \frac{\partial x}{\partial s}, \dot{y} = \frac{\partial y}{\partial s}, \ddot{x} = \frac{\partial^2 x}{\partial s^2}, \ddot{y} = \frac{\partial^2 y}{\partial s^2}, \tag{19}$$

then

$$\frac{\partial y}{\partial x} = \frac{\dot{y}}{\dot{x}}, \frac{\partial^2 y}{\partial x^2} = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^3}, \tag{20}$$

the following curvature function is obtained:

$$K(x, y) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\sqrt{(\dot{x}^2 + \dot{y}^2)^3}} \tag{21}$$

Derivatives are searched for results of smoothing the functions $x(t)$ and $y(t)$, obtained by convolution with a Gaussian kernel, with a glance to the convolution property (differentiation rule):

$$\frac{\partial^n f(x) \otimes g(x, \sigma)}{\partial x^n} = f(x) \otimes \frac{\partial^n g(x, \sigma)}{\partial x^n} \tag{22}$$

By changing the value σ, it is possible to change the level of detail due to smoothing of details. The contour points corresponding to the curvature extremes are chosen as the dominant points.

III. Researches

For comparison the algorithm for simultaneous localization and mapping based on the extended Kalman filter was implemented. It considers the state of the system as a Gaussian distribution and constantly evaluates the mean value (expectation function) and covariance matrix. The update of the system status evaluation is carried out in two stages:

1) **Prediction**
   a) Prediction of the system state:
   $$\bar{\mu}_t = g(\mu_{t-1}, u_t). \tag{23}$$
   b) Prediction of covariance error:
   $$\bar{\Sigma}_t = G_t \cdot \Sigma_{t-1} \cdot G_t^T + R_t, \tag{24}$$
   where $\bar{\mu}_t$ is prediction of the system state at the current moment in time, $g(\mu_{t-1}, u_t)$ is a function of the system state prediction, u_t is a control action at the current moment in time, $\bar{\Sigma}_t$ is the prediction of the system state error at the current moment in time,G_t is transition matrix between states, R_t is system noise.
2) **Correction**
   a) Calculation of the Kalman Gain:
   $$K_t = \bar{\Sigma}_t \cdot H_t^T \cdot (H_t \cdot \bar{\Sigma}_t \cdot H_t^T + Q)^{-1}. \tag{25}$$

b) Update the estimate with the measurement of z_t:

$$\mu_t = \bar{\mu}_t + K \cdot (z_t - h(\bar{\mu}_t)). \qquad (26)$$

c) Update the error covariance:

$$\Sigma_t = (I - K_t \cdot H_t) \cdot \bar{\Sigma}_t \qquad (27)$$

where K_t is the Kalman Gain, H_t is measurement matrix showing the ratio of measurements and states, Q is the covariance of measurement noise, z_t is measurement at the current moment in time, I is identity matrix.

A model describing the kinematics of the robot's motion is used to evaluate the location of the robot and the landmarks.

Let the robot consist of an absolutely solid platform and a coaxial wheel system with a differential drive. It is assumed that the wheels are in point contact with the surface and move without slippage. As the variables of the wheeled robot state are considered the following values: x, y are the coordinates of the robot base point (the middle of the rotation axis of the wheels), θ is the angle between the speed vector of the robot and the positive direction of the axis Ox. The robot is controlled by linear velocity V and angular velocity ω Then the kinematic model of a mobile wheeled robot (Fig. 2) in discrete time is described by the following equation:

$$g(x_t, y_t, \theta_t) = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} V \cdot dt \cdot cos(\theta + \omega \cdot dt) \\ V \cdot dt \cdot sin(\theta + \omega \cdot dt) \\ \omega \cdot dt \end{bmatrix} \qquad (28)$$

Despite the fact that this model is a simplified model of mobile wheeled robot movement (engine dynamics, wheel deformation and other mechanical effects are not considered), it takes into account non-holonomic connections inherent in most mobile wheeled robots.

Measurement of the landmark at the current moment in time (Fig. 3.):

$$z_t = \begin{bmatrix} r \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{(x_m - x_l)^2 + (y_m - y_l)^2} \\ atan\frac{y_m - y_l}{x_m - x_l} - \theta \end{bmatrix}, \qquad (29)$$

$$x_l = x + d \cdot cos\theta, y_l = y + d \cdot sin\theta, \qquad (30)$$

where x_m, y_m are landmark coordinates, x_l, y_l are lidar coordinates, x, y, θ are robot coordinates, r is the distance between the lidar and the landmark, α is the angle between the lidar and the landmark, Q is uncorrelated white Gaussian process with zero mean and constant covariance:

$$Q = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\alpha^2 \end{bmatrix}. \qquad (31)$$

Landmarks detected by the device sensors at each step include existing landmarks on the map, as well as new landmarks. If the found landmarks are not correlated with the existing ones on the map, duplicate landmarks will be added to it, which can lead to the wrong operation of the SLAM algorithm.

The nearest neighbor association algorithm is used here. It compares the distance to existing landmarks with a predetermined threshold for the association of each observed landmark. This simple algorithm does not take into account the relation between the landmarks, so the probability of incorrect association of the algorithm is high.

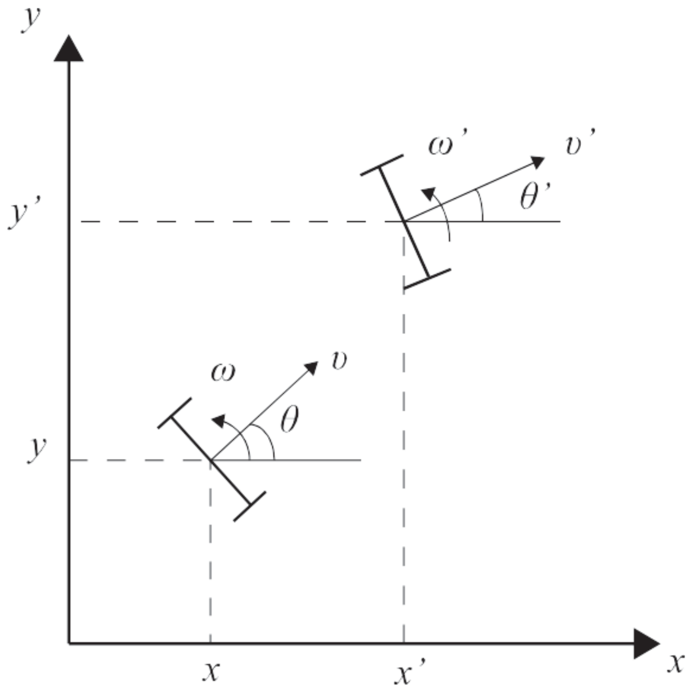


Fig. 2. Wheeled robot

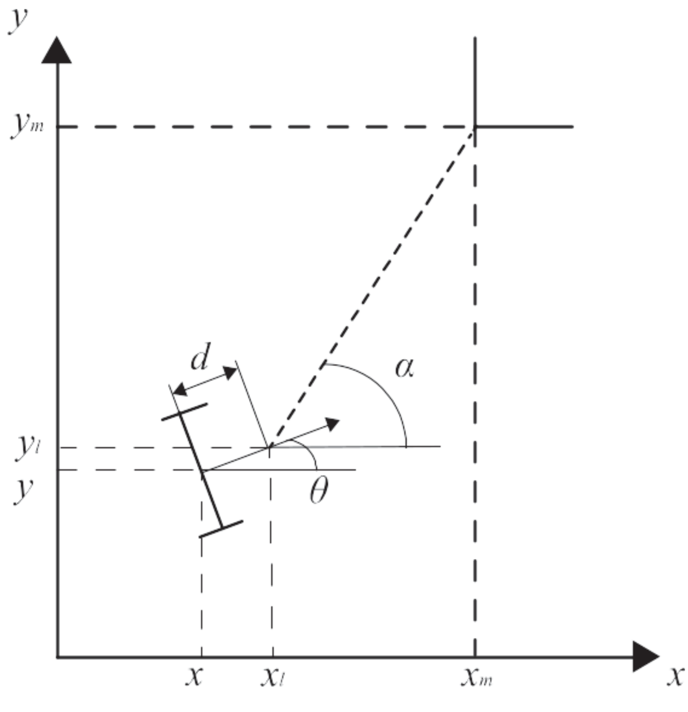
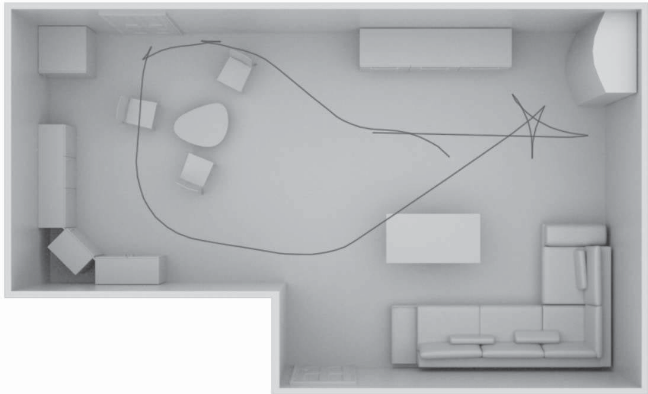
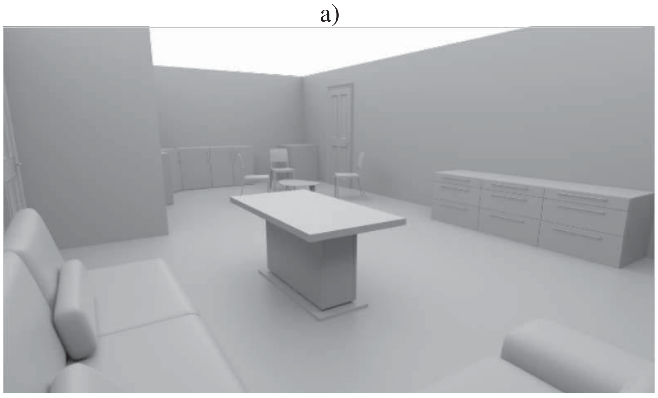


Fig. 3. The geometric relation between the robot position and landmark

Experimental researches are pursued in the virtual environment of Gazebo. For research a test scene, the 3d model of an ordinary room with furniture, is developed (Fig. 4). One run is carry out and a readings of the laser scanning system



a)



b)

Fig. 4. Three-dimensional model of the room

TABLE I. RESEARCH RESULTS

	Matched filtering method	Wu algorithm	Teh-Chin algorithm	The curvature scale space method
RMSE of x	0.071	0.171	0.522	0.235
RMSE of y	0.071	0.112	0.272	0.317
RMSE of θ	0.016	0.024	0.052	0.05

are recorded in the simulated room. Then, according to the recorded data, the SLAM algorithm with different detectors of landmarks is operate 4 times.

The Table I shows that method of detection using matched filtering and the Wu algorithm works the best. SLAM using Teh-Chin algorithm works the worst because of this algorithm sensitivity to noise.

Figure 6 shows the error and 2σ-boundaries of the x, y, θ coordinates of the robot using the matched filtering method. Figure 5 shows the constructed room map based on the results of the SLAM algorithm.

The false negative detection rate, as shown on the Fig. 7, was calculated according to following procedure: for every landmark that is detected within the field of view of a given LIDAR scan, we attempt to match it to previously-known landmarks. If it is successfully associated with an existing landmark, the observation count o for that landmark is incremented. If a landmark should have been detected, but was not or it was too far away for the lidar, the failure count f for that landmark is incremented. We compute the false negative rate as $f/(f+o)$. Our detector is able to identify features which are middle stable: they are observed many times with low false negative rates. The figure also shows that some features are detected but are observed less frequently and less consistently; empirically, these seem to correspond to small and difficult to observe features.

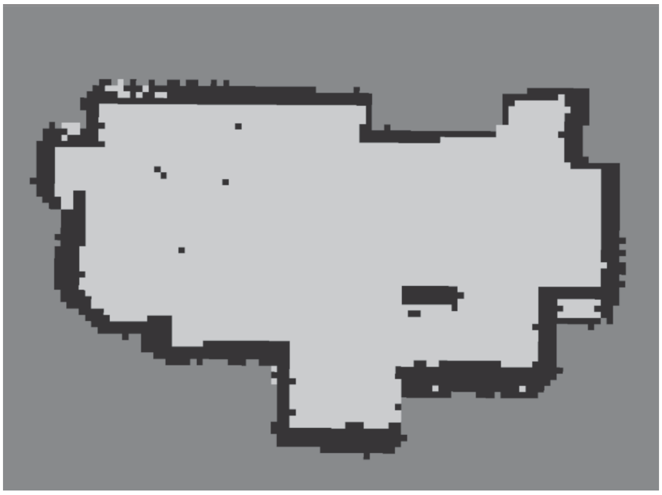


Fig. 5. Constructed room map

IV. CONCLUSION

In this work, we discussed the landmark algorithms by using a LIDAR as a sensor to control the odometry of the robot. Algorithms comparison of the contur analysis is made on the basis of 4 algorithms: Wu, Teh-Chin, Matched filtering, Curvature scale space, and 2 criteria: Root Mean Square Error (RMSE), False Negative Rate (FNR). According to criterion of RMSE the best are Matched filtering in comparison with others by 3-8 times(Fig. 6). In the analysis by criterion of FNR the best are Matched filtering and Curvature scale space (Fig. 7), but additional measurements in real environments about use of real kinematic model are necessary for decision-making.

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

1) **YDLidar X4**
2) **Arduino Mega**
3) **Raspberry PI 3 with ROS**
4) **Raspberry Pi camera with 220 fisheye**

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

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

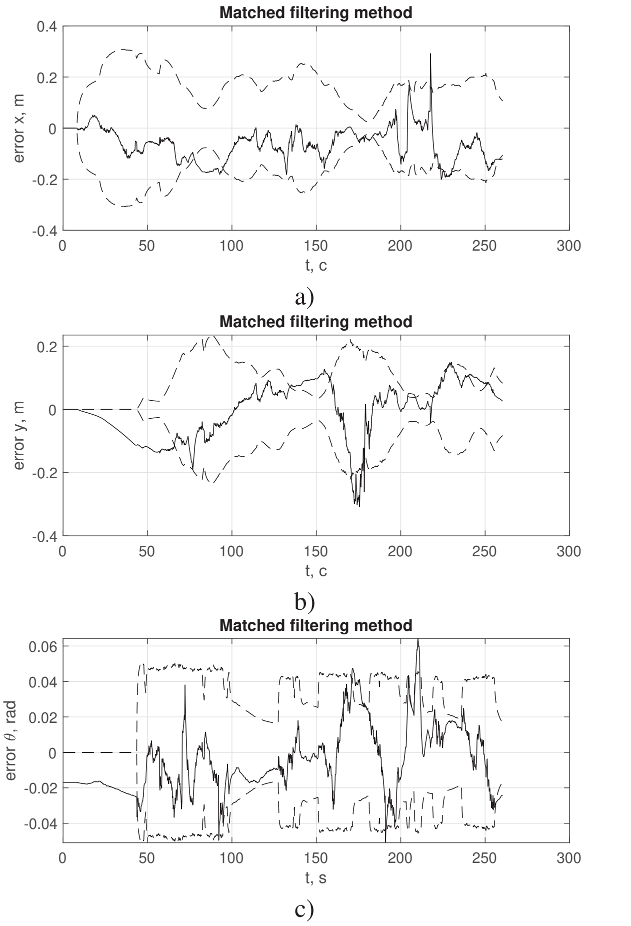


Fig. 6. Error and 2σ-boundaries: a) x coordinate; b) y coordinate; c) θ coordinate

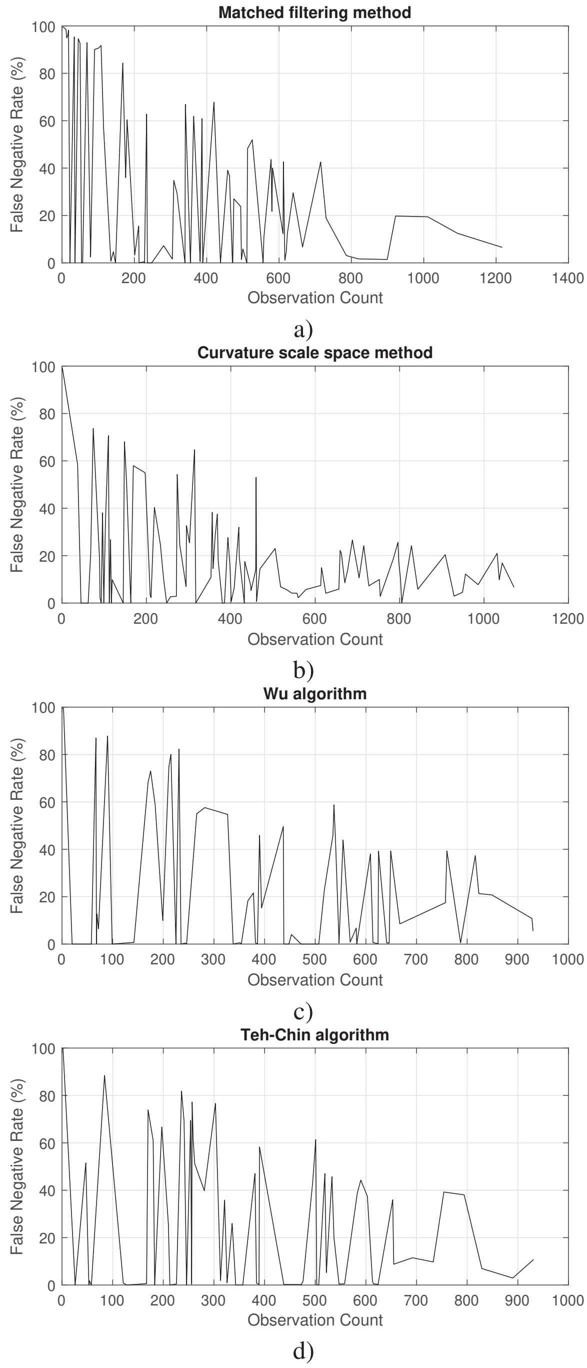


Fig. 7. Observation count for different algorithms: a) matched; b) curvature scale space method; c) Wu; d) Teh-Chin

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, insted of other protocols and operation systems.

Raspberry PI camery is based on the top of the robot and calculate the all visiable data and improve it with the data from the LIDAR. On this data we can construct the 3D scene of the world around the robot.

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 Iliya Trapeznikov and Alexandr Prozorov.

REFERENCES

[1] Garrido A., Perez De La Blanca N., Garcia-Silvente M. ”Boundary simplification using a multiscale dominant-point detection algorithm”, Pattern Recognition, 31(6), pp. 791-804

[2] Witkin A.P. ”Scale-space filtering”, Proc. 8th Int, Joint Conf. Artif. Intell., Karlsruhe, Germany, 1983. pp. 1019-1022.

[3] Rattarengsi A. and Chin R.T. ”Scale-Based Detection of corners of planar curves”, IEEE Trans. Pattern Anal. Mach. Intell. 1992.

[4] Wu W.Y. ”An adaptive method for detecting dominant points”, Pattern Recognition 36, 2003. pp. 2231-2237.

[5] Furmanov Ya.A., Krevetzkiy A.V., Peredreev A.K., Rodjentzov 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] Teh C.H., Chin R.T. ”On the detection of dominant points on digital

curves", IEEE Trans. Anal. Mach. Intell, 1989. pp. 859-872.

[7] Ray B.K., Ray K.S. "Detection of significant points and polygonal approximation of digitized curves", Pattern Recognition Lett. 13, 1992. pp. 443-452.

[8] Siegwart, Roland and Nourbakhsh, Illah R. and Scaramuzza, Davide, "Introduction to Autonomous Mobile Robots", 2nd, The MIT Press, 2011

[9] 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

[10] Favorskaya, et al. M. N. "Computer Vision in Control Systems, volume 2", 2015, Springer