DOI: 10.17587/mau.22.246-253

N. A. Buzlov, nikita_buzlov@outlook.com,
Bauman Moscow Technical University, Moscow, 105005, Russian Federation

Corresponding author: **Buzlov Nikita A.,** Postgraduate, Bauman Moscow Technical University, Moscow, 105005, Russian Federation, e-mail: nikita_buzlov@outlook.com

Accepted on February 08, 2021

Scan Matching for Navigation of a Mobile Robot in Semi-Structured Terrain Conditions

Abstract

To ensure unmanned autonomous movement of ground robotic means, it is required to accurately determine the position and orientation of the robot. The present study is related to the estimation of coordinates by comparing the scans of a laser scanning rangefinder in conditions of semi-structed infrastructure and the absence of a global satellite communications signal. The existing methods of comparing scans have significant drawbacks in the conditions of movement over a semi-structured terrain, associated both with the processing time of data from the laser scanning rangefinder, and with the quality of the results obtained. The scan is preliminarily placed in a map consisting of cells. Each cell of around point scan is described by forces represented by the laws of physics or probability theory. In the cells of the map, we take into account the mutual influence of all forces from each point of the scan and thus we obtain the resulting artificial potential field of the scan. The position of the robot is estimated by the change in the number of acting forces of one scan per points of the next scan taking into account their direction. We estimate the orientation of the robot based on the sum of the vector products of the forces and distances to the given forces acting on the points of the next scan. This method allows you to calculate the displacement of the robot between scans regardless of road conditions and terrain. This article presents the results of an experimental verification of the method on a mock-up of a mobile robot equipped with a Velodyne HDL-32 LIDAR. We indicate the operating conditions of the method for a given LIDAR, as well as the time spent on calculating the bias estimate. Given the peculiarities of the LIDAR, we present a method for eliminating the Doppler Effect (distortion) for the original point cloud. A comparative analysis of the developed method for integrating wheel odometry data, inertial and satellite navigation using the Extended Kalman Filter shows the applicability of this method to assess the position and orientation of the robot in conditions of its movement over rough terrain.

Keywords: *visual odometry, localization in semi-structured environments, localization in non-deterministic environments, method of scan registration, normal distribution, sequential comparison of scans, scan registration in space, measurement of the path, Doppler effect, LIDAR distortion*

For citation:

Buzlov N. A. Scan Matching for Navigation of a Mobile Robot in Semi-Structured Terrain Conditions, *Mekhatronika, Avtomatizatsiya, Upravlenie,* 2021, vol. 22, no. 5, pp. 246—253.

DOI: 10.17587/mau.22.246-253

УДК 004.896:535.8

DOI: 10.17587/mau.22.246-253

Н. А. Бузлов, аспирант, nikita_buzlov@outlook.com,
МГТУ им. Н. Э. Баумана, г. Москва

Последовательное сравнение сканов для навигации мобильного робота в условиях слабоструктурированной местности

Для обеспечения беспилотного автономного движения наземных робототехнических средств требуется точно определять положение и ориентацию робота. Настоящее исследование связано с оценкой координат с помощью сопоставления сканов лазерного сканирующего дальномера в условиях слабоструктурированной местности и отсутствия сигнала глобальной спутниковой связи. Существующие методы сопоставления сканов имеют существенные недостатки в условиях движения по слабоструктурированной местности, связанные как со временем обработки данных от лазерного сканирующего дальномера, так и с качеством получаемых результатов. Предложенный метод

основан на использовании искусственного потенциального поля фиксированного размера, создаваемого для каждой точки скана. Для простоты описания весь скан предварительно помещается в карту, состоящую из ячеек. При этом описываемые силы потенциального поля могут быть представлены законами, относящимися как к физике мира, так и к теории вероятностей. В ячейках карты происходит учет взаимовлияния всех сил от каждой точки скана, и, таким образом, получается итоговое искусственное потенциальное поле скана. Положение робота оценивается по изменению числа действующих сил одного скана на точки смежного скана с учетом их направления. Оценка ориентации осуществляется на основании суммы векторных моментов сил, действующих на точки смежного скана. Такой способ позволяет быстро оценивать смещение робота между сканами вне зависимости от условий движения и характера местности. В статье приведены результаты компьютерной апробации метода на данных, полученных от 3D-лидара Velodyne HDL-32 и обозначены условия работы метода для данного лидара, а также время, затрачиваемое на расчет оценки смещения. Ввиду особенности лидара при движении робота приводится способ устранения эффекта Доплера (дисторсии) для исходного облака точек. Проведенный сравнительный анализ разработанного метода по отношению к способу комплексирования данных от колесной одометрии, блока инерциальной и спутниковой навигации, использующий расширенный фильтр Калмана (Extended Kalman Filter), показывает применимость метода для оценки положения и ориентации робота в условиях его движения по слабоструктурированной местности.

Ключевые слова: *визуальная одометрия, локализация в слабоструктурированных средах, метод регистрации сканов, нормальное распределение, последовательное сравнение сканов, регистрация сканов в пространстве, измерение пути, эффект Доплера, дисторсия лидара*

Introduction

The unmanned ground vehicles (UGV) are in development in present by about 150 companies around the world [1]. The most important tasks of autonomous driving control are precise orientation and positioning. In presence of the positioning errors the vehicle cannot build the correct trajectory to reach the route endpoint [2].

Most of the known UGV developers use driving in pedestrian environment in presence of predefined map and global satellite-based positioning system (GPS etc.). However, the known works [4] say that at least in Russia about 10'000 km of roads are considered as dangerous routes for UGVs with weak, imprecise GPS signal. Because of this, the developers usually concentrate on empowering of the onboard computer with algorithms and environment observation techniques designed for semi-structed environments [3].

Usage of laser range scanner (LIDAR) with scan matching techniques allows to significantly improve estimation of UGV's position and orientation. However, for semi-structed (or so called semi-structured) environments the special methods are required to decrease the required amount of calculations and/or matching quality. Thus, the task of UGV realtime LIDAR scan matching for semi-structured environment should be considered as existing and actualized.

In this paper the sequential realtime scan matching technique is adopted to determine UGV's (also 'robot' here and below) coordinates in the model space of semi-structured environment. The technique was tested on the scans produced by Velodyne HDL-32E LIDAR sensor. As the base for comparison, the extended Kalman filter is used with the data from wheel odometry, inertial navigation unit (IMU) and GPS as input channels.

Target setting

The target set in this paper is to develop the LIDAR sequential scan matching technique estimating robot's position and orientation while driving inside semi-structed environment without usage of the high-performance computing equipment (such as GPUs). The treatment frequency should not be less than the scanning frequency itself to ensure realtime response.

Overview of the modern positioning techniques for robots

Wheel odometry and IMU sensory analysis are the traditional techniques of robot/UGV positioning. However, if the UGV is driving autonomously through the semi-structured environment the wheel odometer is vulnerable to the sliding effect and the IMUs are also vulnerable to virtations. These conditions lead to underestimation of the UGV's linear speed, roll/pitch sensory is jittered. The precision can also be improved using SLAM matching the current position of the robot to the predefined map [5—9] but these techniques require more powerful computing. The additional difficulty here is that the semi-structed enviroments don't have many unique objects for being matched with the detected ones. The sequential laser scan matching is one of the techniques that allows to perform realtime refinery of the estimated coordinates.

As the most spreading method of sequential laser scan matching the iterative closest point (ICP) method should be considered [11]. The method consists

of multiple applications of transmission and rotation (**T** and **R**) transformations to the points of the currently treated scan and then minimizing the sum:

$$E(\mathbf{R},\mathbf{T}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_i - \mathbf{R}\mathbf{p}_i - \mathbf{T}\right\|^2$$

where $X = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^3, i = 1, 2, ..., N_X\}$ are the point defined in the first scan, $P = \{\mathbf{p}_j | \mathbf{p}_j \in \mathbb{R}^3, j = 1, 2, ..., N_P\}$ — the point defined in the second scan, $N = N_X = N_P$.

Application of ICP in semi-structed enviroments is not recommended [17]. In the paper [18] indicated that this method works for the scans captured in not more than 0.5 m of distance between and 0.05 rad of angular displacement. For semi-structed environments ICP was tested as it is decribed in [10]. Authors of the paper [10] said that the main disadvantage of ICP in semi-structed environment is absence of possibility to implement realtime response. We suppose that it is conditioned by usage of the multiscan as the initial point cloud in that implementation, it leads to increased amount of calculations.

ICP has many similar methods, closely connected more or less, where in addition the point cloud filtering is implemented [12]. Also the weighted point-based approaches [13, 14] and double match approach [15] are exist. Especially for weakly structured environment the research [18] is known where Normal Distribution Transform (NDT) [19] method is used. In this method the predefined map should be created from the cells where each cell stores the value of probability density in assumption of normal distribution for the points of the cloud indexed within. Then the minimal value of mutual correlation function should be calculated for map and the scan to determine the displacement. In NDT the cell size is important. If the cell is too large, the detail of the environment would be over filtered influencing precision of localization. In contrary, if the cell is too small the map will become detailed increasing amount of calculations. NDT is difficult for implementation in realtime responding system because of its requirements.

Several other approaches such as FLIRT, NARF, FPFH [20-22] are the laser odometry techniques based on searching for peculiarities within the scan. They are very limited in application because require high quantity of flat surfaces and straight lines to be presented in the scan.

The main problem of scan matching within the task of realtime robot position estimation in semi-structed environment and absence of GPS is that the precision of estimated **T** and **R** matrices is dependent from initial displacement between the scans.

Scan matching method proposal

The proposed scan matching method determines the robot's position using comparison between the current obtained LIDAR scan and the synthesized potential field of the basis. As the basis the first obtained LIDAR scan is used. To calculate the potential field, the fixed-size discrete potential area is assigned. For each area in the potential field the forces are equipotentially and equidistantly distributed. Within the abovementioned discrete areas vector of force for the given point within the area is oriented to the scan point and its value is inverse-proportional to the distance between the scan point and this given point. If the radius of the area defined for the scan point is smaller than sensor's resolution factor, the forces are considered non-interfering, and in opposite case for some points the field-induced force should be calculated from the sum of vectors. Thus the initial state of the potential field is determined for the basis. Then each obtained LIDAR scan should be placed in the potential field determining the force for each point which are now considered as induced force point. Then the resultant force and torque should be calculated for the entire scan (as the mass factor 1 is taken). As the complete result the coordinate displacements are calculated by axis *Ox*, *Oy*, *Oz*. Below one of the possible implementations of such method is given.

The Algorithm

STEP 1. The first scan obtained from LIDAR is assigned as the basis. It is converted into the map stamp subdivided into the fixed-size cells and snapped to LIDAR itself. Each cell in this setting is a 3-dimensional cube defined with edge length. Then the sphere with radius R is build centered on the basis point and it also is subdivided by the cells but now the force point should be defined in the center of each cell. The graphical representation of such definition for flat 2D space is shown on Fig. 1.

Then the vector of force is determined for each force point with the following value:

$$|\mathbf{F}| = \frac{1}{r^2}$$

here r — is a distance between force point and center of the sphere, as it was described above. As an analogue here the Coulomb's law could be considered for electrically charged particles. The distribution of lengths for **F** vectors is not strictly defined,

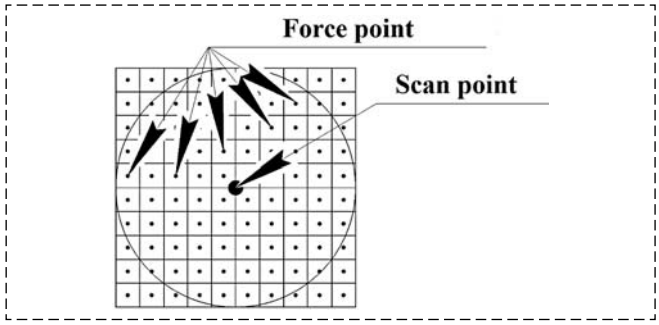

Fig. 1. An example of the formation of force points near the point of the basis scan in the plane case

so, for example, the normalized distribution with 0 — expectation value could be used here too:

$$|\mathbf{F}| = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

here σ is dispersion of the force points relatively to the center of the sphere R.

STEP 2. Now the resultant force would be calculated from all cells of the map with every point of the basis:

$$\forall j: \mathbf{F}_\Sigma^j = \sum_{i=1}^{N_i^j} \mathbf{F}_i^j \qquad (1)$$

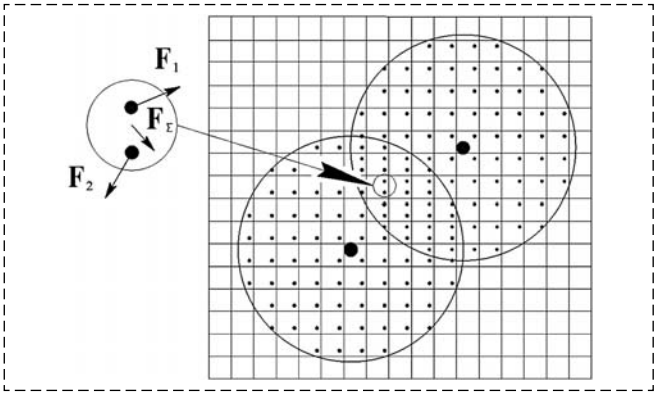

Fig. 2. An example of calculating the resultant force for two intersecting potential areas (spheres) ($\mathbf{F}_1$, $\mathbf{F}_2$, are the forces acting on the first and second points of the scan, respectively, $\mathbf{F}_\Sigma$ is the sum of forces $\mathbf{F}_1$ and $\mathbf{F}_2$)

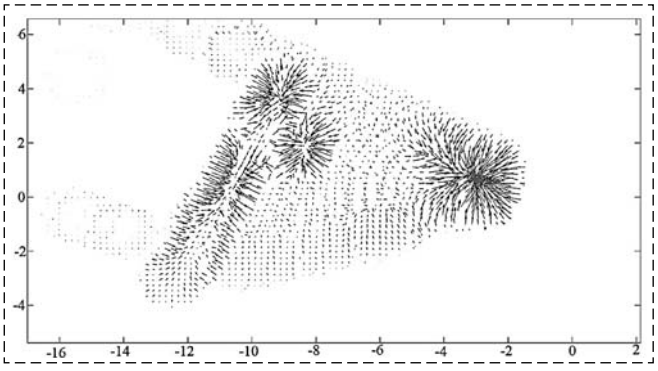

Fig. 3. The potential field of scan

here we have $\mathbf{F}_i^j$ — the mocked "gravity" for i-point located in the cell j, where $i = 1...N_i^j$ and N is quantity of the force point located in the cell j, $j = 1...M$ and M — number of the cells. The graphical representation of this step is given on Fig. 2.

Fig. 3 contains the example of the potential field generated for the use case.

Now every cell defined in the basis is assigned with the vector of force.

STEP 3. Now the new obtained scan $P = \{\mathbf{p}_i | \mathbf{p}_i \in \mathbb{R}^3, i = 1, 2, ..., N_P\}$ is assigned to the map. Every point p_i is now represented with the map point c_j where $(\mathbf{p}_i, c_j) \in S$, $S: P \to C$, $S \subseteq P \times C$, and $C = \{c_j \mid j = 1...M\}$ is the set containing all existing cells for the instance of algorithm. For the every point $\mathbf{p}_i$ the force $\mathbf{F}_\Sigma^J$ is now determined.

STEP 4. The rebase criteria: if the quantity of the forces defined for the current obtained scan is less than 60 % of the forces defined in the basis, rerun the algorithm from the STEP 1. The criterion for 60 % is described below.

STEP 5. Estimation: angular displacement. The orientation axis $\mathbf{a}_z$ is determined from:

$$\mathbf{a}_z = \sum_{i=1}^{N_p} \frac{\mathbf{F}_\Sigma^J \times \mathbf{L}_i}{\left\|\mathbf{F}_\Sigma^J\right\| \left\|\mathbf{L}_i\right\|}$$

here we have $\mathbf{L}_i$ — the vector that locates the point p_i relatively to the center of the scan (LIDAR origin point).

The rotation matrix $\mathbf{a}_z$:

$$\mathbf{R} = \begin{pmatrix} 1 - 2 \cdot z^2 & -2 \cdot w \cdot z \\ 2 \cdot w \cdot z & 1 - 2 \cdot z^2 \end{pmatrix}$$

here $z = \sin(|\mathbf{a}_z|/2)$, $w = \cos(|\mathbf{a}_z|/2)$.

The linear displacement estimation is calculated as the sum of all forces defined in the current LIDAR scan divided by quantity of them.

$$\left\|\mathbf{F}_\Sigma\right\| = \frac{\sum_{i=1}^{N_p} \left\|\mathbf{F}_\Sigma^J\right\|}{K}$$

here K is the quantity of cells where $\left\|\mathbf{F}_\Sigma^J\right\| \neq 0$, $K \leqslant M$.

STEP 6. The expression (1) is used to calculate the overall displacement and quantity of the force vectors. The stepping model is described below.

STEP 7. Iterate over steps 4—6 until the completion condition is achieved: the quantity of forces for the current iteration is over 99 % of the forces determined on the previous iteration.

STEP 8. The current scan is considered as the new basis after the completion condition is achieved before iterating over steps 1—7 again.

Experimental implementation and test

For the described algorithm, the implementation was built based on Robot Operating System (ROS) Melodic Morenia [23] on language C++. As the data source the.bag files, containing scans recorded with Velodyne HDL-32E were used. Scanning frequency is 10 Hz, linear speed of the UGV 5.5 m/s, angular aperture 180° in 0-starting range. The angular resolution is about 0.2°. The obtained point clouds were distortion is corrected and filtered by distance ($r_{\min}$, $r_{\max}$) to increase stability.

Correction of point cloud

The LIDAR cloud deforms while the robot is moving. The total distortion depends on the values of the linear and angular speed robot. In [24], it is proposed to correct this distortion by the Cauchy method only in case of 2D space. However, due to the movement of the robot in semi-structured terrain, ignoring the distortion of the cloud in the roll and pitch angle can have a negative effect on the operation of the algorithm. Therefore, it is suggested to use the following idea to correction distortion point cloud.

All scan points P can be divided into subsets formed by vertical scan points for a fixed value of the LIDAR rotation angle q in the horizontal plane:

$$P^q = \{\mathbf{p}_i^q | i = \overline{1,32}\},\ P = \bigcup_{q=\Delta\delta}^{2\pi/\Delta\delta} P^q,\ \Delta\delta = \omega_L \cdot \Delta t_q$$

here Δt_q — time interval of forming a set of points P^q in total scan time dT; $\Delta\delta$ — horizontal resolution LIDAR; ω_L — LIDAR angular speed.

Then, the set of points with corrected distortion $P^q_{correct}$ derived from the original P^q by multiplying each point $\mathbf{p}_i^q = (x, y, z, 1)^T$ to the homogeneous transformation matrix $\mathbf{T}_{\Delta t_q}$:

$$\forall i : \mathbf{p}^q_{i\ correct} = \mathbf{T}_{\Delta t_q}\mathbf{p}_i^q;$$

$$\mathbf{T}_{\Delta t_q} = \begin{pmatrix} \mathbf{R}_{\Delta t_q} & \mathbf{r}_{\Delta t_q} \\ 0 & 1 \end{pmatrix}$$

here $\mathbf{R}_{\Delta t_q}$ и $\mathbf{r}_{\Delta t_q}$ — rotation matrix and translation vector, respectively.

The rotation matrix has the form:

$$\mathbf{R}_{\Delta t_q} = \begin{pmatrix} c_\alpha^q c_\beta^q & -s_\beta^q c_\alpha^q c_\gamma^q & c_\gamma^q s_\beta^q s_\alpha^q + c_\alpha^q s_\gamma^q \\ s_\beta^q & c_\alpha^q c_\beta & -c_\beta^q s_\alpha^q \\ -c_\beta^q s_\gamma^q & s_\gamma^q c_\beta^q c_\alpha^q + s_\alpha^q c_\gamma^q & s_\beta^q s_\alpha^q c_\alpha^q + c_\gamma^q s_\alpha^q \end{pmatrix}$$

here $c_\alpha^q = \cos(\omega_\alpha^q \Delta t_q)$, $s_\alpha^h = \sin(\omega_\alpha^q \Delta t_q)$, $c_\beta^h = \cos(\omega_\beta^h \Delta t_q)$, $s_\beta^q = \sin(\omega_\beta^q \Delta t_q)$, $c_\gamma^q = \cos(\omega_\gamma^q \Delta t_q)$, $s_\gamma^q = \sin(\omega_\gamma^q \Delta t_q)$, $\omega_\alpha^q, \omega_\beta^q, \omega_\gamma^q$ — projections of the angular speed the robot movement on the axes Ox, Oy, Oz basis scan during formation the set of points P^q.

Translation vector $\mathbf{r}_{\Delta t_q}$ defined as:

$$\mathbf{r}_{\Delta t_q} = \begin{pmatrix} v_x^q \Delta t_q \\ v_y^q \Delta t_q \\ v_z^q \Delta t_q \end{pmatrix}$$

here v_x^q, v_y^q, v_z^q — projections of the linear velocity of the robot movement on the axes Ox, Oy, Oz basis scan formation the set of points P^q.

Filtering point clouds by distance

Velodyne LIDAR HDL32E returns a large number of points, which increases the computational load. In order to reduce the load, the point cloud is filtered by distance ($r_{\min}$, $r_{\max}$). This procedure significantly reduces the computational load, but a small amount of information could be lost.

The median value of the number of points (17500) falls on a distance of 8 meters. Based on the interquartile range for setting the near and far distance thresholds, 75 % of the average number of points above and below the median were taken. Thus, $r_{\min} = 5$ m and $r_{\max} = 22$ m were determined.

Scan registration time

Computer for testing of the registration of scans was carried out using an intel core i7 7700HQ processor. The time for calculating the force field for one scan averaged 5 ms, and the time for searching for the displacement was 9 ms (Fig. 4) in the obtained software implementation of the algorithm

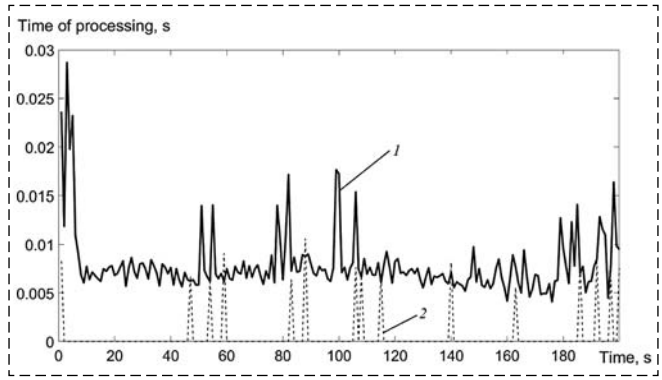

Fig. 4. Time of potential field calculation and scan alignment:
1 — time to calculate displacement; *2* — potential filed calculation time

The radius of the sphere R for one power point in the computer experiment was taken equal to 1 m. The time was measured using the class *ros::Time*.

Accuracy of the method depends on distance between basis and inserted scan on

After first procedure registration scan, for new scan is applied using the uniform transformation matrix. This method has a limitation due to the fact that the scans can differ from each other during the movement, therefore the number of points involved in attraction will be less, and the estimate with damage and orientation will be incorrect.

Therefore, it was proposed to update the basis scan depending on three parameters: the distance between the new scan from basis scan, the number of points attracting into the force field and the number of iterations to estimate the displacement.

To determine the boundary conditions of the algorithm as the ground-truth position of the robot, the Extended Kalman Filter (EKF) estimate [25], which received wheel odometry data, inertial data, and satellite navigation data, is used (Fig. 5).

As can be seen from Figure 6, along the *Ox* axis at a mark of 9 meters, the error with respect to the EKF was 15 cm, and at 12 the algorithm could not determine the offset. Thus, the basis scan needs to be updated at a certain distance. It can be seen from the figure that it should not be changed 8 meters.

Accuracy of the method depends on the number of iterations

In the process of computer modeling, it was found that at a distance of 6 meters from the basis scan, the number of iterations does not exceed 50 (Fig. 6). Then small fluctuations (60...70 iterations) appear in the interval from 5 to 8 meters.

At a mark of 10 meters, the number of iterations overcomes the threshold of 200, while the error in estimating the displacement between scans increases (Figure 6). Thus, in order for the scans to be matched correctly, the number of iterations should not exceed 100.

Accuracy of the method depends on the number of points inside the potential field

In the process of registration scans, the number of points of the new scan attracting of the artificial potential field changed in accordance with the dependence, the graph of which is shown in Fig. 7.

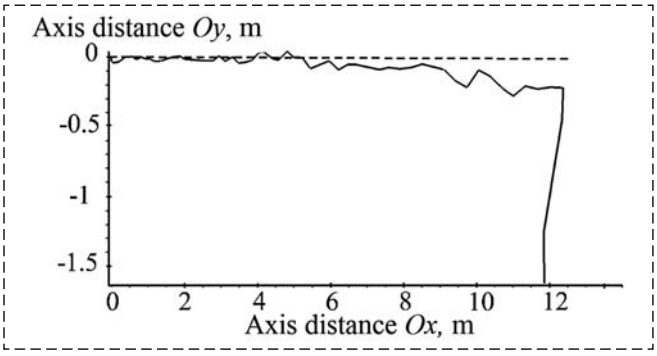

Fig. 5. Estimation of the displacement of the coordinates of the robot. Dashed Main line — output from Kalman filter, Main line — odometry by laser scanning rangefinder

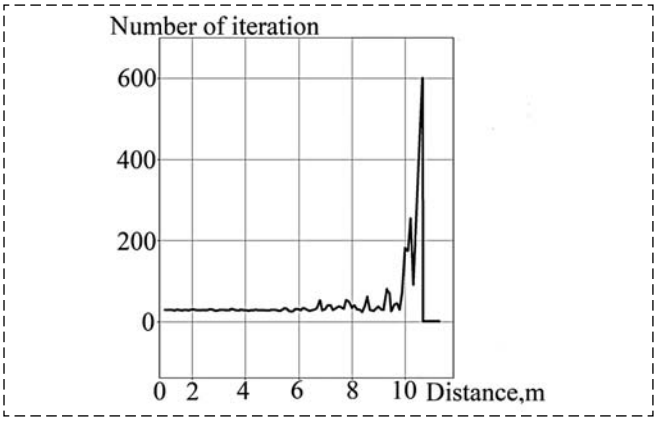

Fig. 6. Changing the number of iterations depending on the distance between scans

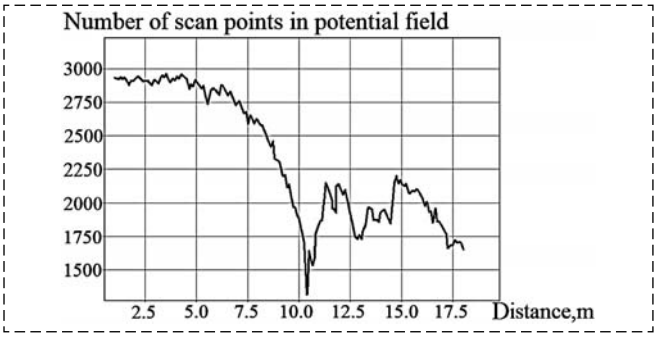

Fig. 7. Changing the number of scan points inside the artificial potential field depending on the distance travelled by the robot

The figure shows that with a sphere radius $R = 1$ m, the number of points attracting of the force field of the basis scan after 10 meters of movement was about 63 %, after which the displacement between the scans was not estimated.

Thus, to update the basis scan, the following criterions were adopted: the distance of movement is 8 meters, the maximum number of iterations is 100, the number of points that fall into the force field of the basis scan is at least 60 %.

Threshold the offset step estimation

The scan registration time can be reduced by choosing the correct initial offset step. One of such approaches is the introduction of a mathematical model of the movement of the robot.

In this work, to speed up the operation of the algorithm, a rather rough assumption was made that the movement of the robot corresponds to the movement of a material point. Accordingly, during the time of receiving two nearby scans dT, the robot can move by no more than the value $\hat{x}$:

$$\hat{x} = \hat{v}dt - \frac{\hat{a}dT^2}{2}$$

here $\hat{x}$ — estimation of the longitudinal displacement of the robot; $\hat{v}$ — estimation of the current longitudinal speed of the robot in the direction of movement; $\hat{a}$ — estimation of the current longitudinal acceleration of the robot in the direction of movement.

In addition to the maximum step, it is necessary to set the minimum threshold at which the algorithm will stop looking for an offset. Although the algorithm can converge to both local and global minimum, it is necessary that the lower threshold for the number of iterations does not significantly affect the accuracy. To determine the minimum displacement step, the root-mean-square deviation (RMS) of the linear and angular velocities of the robot is selected if the robot is not moving. The variance of linear and angular and velocities, presented in Table, was determined over a 10 second interval.

Dispersion of velocities if the robot is not moving

Coordinates	Value dispersion
V_x	4,1098e-07
V_y	6,7807e-07
V_z	1,3276e-06
W_x	3,4141e-04
W_y	6,2856e-04
W_z	5,2658e-04

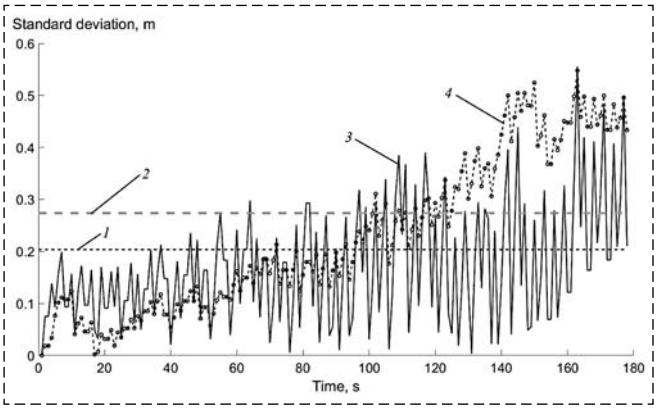

Fig. 8. Deviation of the trajectory obtained using laser odometry compared to the output from the Kalman Filter:
1 — *X* Standard deviation; *2* — *Y* Standard deviation; *3* — *X* offset; *4* — *Y* offset

Based on the given data, the standard deviation of linear and angular displacements was obtained — 0.0001 m, 0.002 rad, respectively. Thus, at a LIDAR scanning frequency of 10 Hz, the linear and angular displacement steps should be at least 0.00001 m and 0.0002 radians, respectively.

Accuracy of the method depends on the radius sphere of force filed each point

The last tunable parameter of the algorithm is the radius of the sphere. This parameter strongly depends on the model of the LIDAR, since the number of force spheres of the basis scan directly depends on the resolution of the LIDAR. If the total amount of forces is too small, the estimate can be made with essential error. However, a significant increase in the size of the sphere will increase the time for calculating the potential field and registering scans. During the operation of the algorithm, it was experimentally established that for the Velodyne HDL32E LIDAR, the sphere radius equal to 1 m is optimal. With a decrease in the sphere radius, the estimation accuracy sharply decreases. As the radius of the sphere increases, the time for calculating the force field increases, which is why the first scan is skipped and information about the robot's displacement movement is lost. Therefore, developers are invited to experimentally search for the required sphere size depending on the LIDAR model.

Calculation of the estimation error of the developed method

Additionally, a comparison was made with the estimate obtained with the EKF by the following formula:

$$c = \sqrt{\frac{\sum_{k=0}^{K}(x_k - x'_k)^2}{K}}$$

here x — estimate position robots according to the developed method, x` — estimate from EKF, K — number of scans.

The comparison results are shown in Fig. 8.

The calculated error was 0,27 m on the axis *Ox*, 0,21 m on the axis *Oy*, 0,15 m on the axis *Oz*.

The results of the experiment show that the laser odometry method allows one to accurately determine the path went by the robot. The deviation error of the developed method with respect to EKF was 3.4 %.

Conclusion

This article proposes a new method for scan matching to estimate the displacement of a robot in a semi-structured area. This method showed good results: when working on only one core of the Intel Core i7-7700-HQ processor, the time for estimating the offset was 9 ms, while the cost of calculating the artificial potential field of the basis scan was 5.0 ms. Although sometimes the calculation and formation time reached 24 ms. It should be noted that the time spent on calculating the offset is much less than in the NDT method, and the amount of computing resources is not significant.

The magnitude of the error in estimating the displacement in comparison with the EKF was 3.4 % in a section lasting 18 seconds with an average linear speed of 5.5 m / s. The approach used in this method allows the offset to be determined regardless of the environment, since the scans are attracted of the potential field created around the points of the basis scan. It should be noted that only one force will act on each point of the scan, due to the resolution of the LIDAR, and therefore the direction with this approach is known a priori, and the time to search for the displacement will be spent only on determining its value.

The disadvantages of this method include the need to adjust the algorithm depending on the type of LIDAR.

In the future, it is planned to replace the simplest model of the motion of a material point with a kinematic model taking into account the mass of the robot with boundary conditions for evaluating the convergence.

References

1. **Brands** and companies. Top 50 developers of autonomous cars and technologies in the world, Cars section of the drone website, available at: https://bespilot.com/marki-i-kompanii.

2. **Sokolov S. M., Boguslavsky A. A.** Unified layout of integrated intelligent systems for information support of autopilots of mobile devices, *Izvestia of the Southern Federal University. Technical Science,* 2016, no. 2 (175), pp. 200—213 (in Russian).

3. **Gerasimov V. N., Mikhailov B. B.** Solving the problem of controlling the movement of a mobile robot in the presence of dynamic obstacles, *Vestnik BMSTU. Instrumentation. Special issue "Robotic systems"*, 2012, no. 6, pp. 83—92 (in Russian).

4. **Reutov E. V., Golovina L. S.** Problematic aspects of the operation and design of unmanned vehicles, *International Scientific Research Journal.* 2019. no. 5-2 (83), pp. 14—18 (in Russian).

5. **Adamova A. D., Zhukabaeva T. K., Mukanova Zh. A.** Review of methods for localization and construction of maps of the surrounding area of mobile robots, *Automation. Computer Science*, 2018, no. 2 (43), pp. 57—62 (in Russian).

6. **Kaess M., Dellaert F.** Probabilistic structure matching for visual SLAM with a multi-camera rig, *Computer Vision and Image Understanding*, 2010, vol. 114, pp. 286—296.

7. **Carrera G., Angeli A., Davison A. J.** SLAM-based automatic extrinsic calibration of a multi-camera rig, *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 2652—2659.

8. **Davison A., Gonz lez Y., Kita N.** Real-time 3D SLAM with wide-angle vision, *IFAC Proceedings Volumes*, 2004, vol. 37, no. 8, pp. 868—873.

9. **Altukhov V. G.** Overview of omnidirectional vSLAM mobile robot navigation technology, *Collection of scientific papers of the Novosibirsk State Technical University*, 2018, no. 2, pp. 81—92 (in Russian).

10. **Borisov A. G., Gol S. A., Luksha S. S.** Analysis of the efficiency of algorithms for processing three-dimensional laser scans in the problem of building maps for navigation of mobile robots, *Bulletin of the Ryazan State Radio Engineering University,* 2013, no. 46 (3), pp. 34—42 (in Russian).

11. **Besl P. J. and McKay N. D.** A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992. Vol. 14, pp. 239—256.

12. **Removing** outliers using a Conditional or Radius Outlier removal, PointCloud.org, available at: http://pointclouds.org/documentation/tutorials/remove_outliers.php (дата обращения 20.12.2019).

13. **Eggert D. W., Fitzgibbon A. W., Fisher R. B.** Simultaneous registration of multiple range views for use in reverse engineering of CAD models, *Comput. Vis. Image Underst.,* 1998, vol. 69, pp. 253—272.

14. **Bergevin R., Soucy M., Gagnon H., Laurendeau D.** Towards a general multi-view registration technique, *IEEE Trans. Pattern Anal. Mach. Intell.*, 1996, vol. 18, no. 5 pp. 540—547.

15. **Lu F.** Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans, *Journal of Intelligent and Robotic Systems*, 1997, no. 18 (3), pp. 249—275.

16. **B dkowski J, Mas owski A.** GPGPU implementation of On-Line point to plane 3D data registration, *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics,* 2011, Jul 17, pp. 1—6.

17. **Minguez J., Montesano L., Lamiraux F.** Metric-Based Iterative Closest Point Scan Matching for Sensor Displacement Estimation, *IEEE Transactions on Robotics*, 2006, vol. 22, iss. 5, pp. 1047—1054

18. **Biber P., Strasser W.** The normal distributions transform a new approach to laser scan matchin, *Proceedings of Intelligent Robots and Systems*, 2003. Vol. 3, pp. 2743—2748.

19. **Magnusson M.** The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection Philosophy doctor, Örebro universitet, 2009.

20. **Steder B., Rusu R. B., Burgard W.** Point feature extraction on 3D range scans taking into account object boundaries, *Computer Science Published in IEEE International Conference*, 2011, DOI:10.1109/ICRA.2011.5980187.

21. **Steder B., Rusu R. B., Konolige K., Burgard W.** NARF: 3D range image features for object recognition, *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010, Oct 8. Vol. 44.

22. **Radu B. R., Blodow N., Beetz M.** Fast Point Feature Histograms (FPFH) for 3D registration, *2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009.

23. **Robot** Operating System, available at: http://www.ros.org/wiki/ (date of access 04.03.2018).

24. **Merriaux P., Dupuis Y., Boutteau R., Vasseur P., Savatier X.** LiDAR point clouds correction acquired from a moving car based on CAN-bus data, arXiv preprint arXiv:1706.05886. 2017 Jun 19.

25. **Moore T., Stouch D. A.** Generalized extended Kalman filter implementation for the robot operating system, *Intelligent autonomous systems,* 2016, no. 13, pp. 335—348.