Self-localization of Mobile Robot in Unknown Environment

Alexandr Prozorov, Alexandr Tyukin, Ilya Lebedev, Andrew Priorov
P.G. Demidov Yaroslavl State University
Yaroslavl, Russia
{alexprozoroff, tyukin.alexx, ilyaleb}@gmail.com, andcat@yandex.ru

***Abstract*—In this paper we propose a method for solving the SLAM problem for mobile robot when moving in an unknown environment. Our method takes computational advantages of the FastSLAM algorithm. To estimate the position of the robot, we use a particle filter. The weights for the set of particles that characterize the expected position of the robot, are determined by the condition number of the plane homography matrix. It can be considered as the projective mapping of points of the scene on the two-dimensional surface of camera sensor. A set of unscented Kalman filters is used to estimate the positions of detected landmarks which are forming the map of the observed environment. Methods for detecting and description of landmarks were not considered in this paper, as it is beyond the scope of this work.**

I. Introduction

Autonomous navigation is one of the most important tasks of modern robotics. Methods for simultaneous localization of the robot and environment mapping (also known as SLAM algorithms - Simultaneous Localization and Mapping) are carried out by many research groups around the world.

The object of the application of these algorithms in most cases is a mobile robotic platform, equipped with a set of sensors. The type of the sensor is determined by the scope of the problem: the precision in estimating the trajectory and mapping, lighting conditions and the geometry of space. This can be laser rangefinders, digital cameras - visible or infrared, sonar, etc.

The task of the robot consists of computing motion trajectory and space mapping, which are not known a priori. It is also necessary to have the possibility of subsequent positioning within a resulting map. Despite the fact that the type of calculated map is entirely dependent on the characteristics of sensors and the conditions of the surrounding area, most often the problem reduces to finding specific areas of space, the so-called features, characterized by a stable identity within the stream of data coming from the sensors.

Recent years have seen a significant increase in interest in the complex SLAM algorithms based on the use of digital cameras as the main sensor. The main reason for this is a significant increase in computing power and reduce the cost of image processing devices. The practical application of many algorithms goes to real time. Restrictions on the resolution of the original image is weakening. Although the use of high-resolution images are still associated with certain difficulties, low cost cameras in comparison with other sensors, traditionally used in real-SLAM, allowing them to find a wide range of applications.

In each particular case the position of the camera on a mobile platform is selected according to the external conditions. The presence of moving objects in the camera field of view makes the direct optical axis of the camera closer to the line perpendicular to the plane of the platform motion to offset the distortions introduced by such objects (car, pedestrian). In extreme cases, the field of view is focused directly on the ground or the ceiling, in the case of indoor navigation.

The proposed method of finding the trajectory of motion is based on the use of particle filter to estimate the position of the robot and the unscented Kalman filter to estimate the position of feature points in the image. The weights for the cloud particles that characterize the expected position of the robot, are determined by the value of the matrix plane homography as projective mapping of points of the scene on the two-dimensional surface of camera sensor.

II. Problem Statement

Over the last decade, the field of application of simultaneous localization and mapping algorithms has grown tremendously. The most frequent argument in favor of these methods is the need for robot navigation in the absence of the possibility of using global positioning systems, as well as in applications requiring greater accuracy than satellite solutions can provide.

In general, the task of localization is to estimate the current position of the mobile robot in space, depending on the previous positions, available measurements, control commands and the observation model of the surrounding space. The result of the whole set of algorithms is the map of the surrounding area, as well as full or partial path of mobile robot within the resulting map.

ISSN 2305-7254

Location of the robot is usually seen in the local coordinate system related to the initial position, as there is no a priori information about the initial coordinates. The requirements to constructed map are defined by the possibility of using it in localization task. Otherwise, localization and mapping are carried out independently, which is leading to a continuous growth of errors. Fig. 1 shows an example of the algorithm usage. It depicts two paths - a real robot trajectory and the trajectory calculated by the SLAM algorithm.

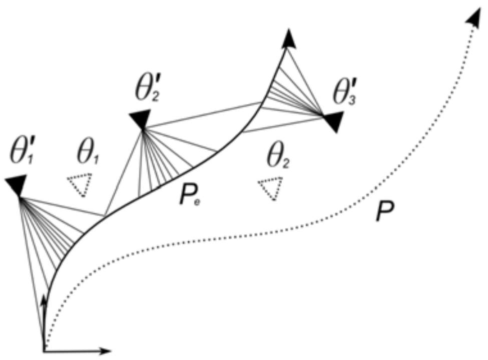

Fig. 1. Calculating the trajectory of the robot and mapping with landmarks

The actual positions of landmarks are denoted by letters $\theta_1..\theta_3$, $\theta'_1..\theta'_3$ are estimated position of landmarks. Error in constructing the estimated path P_e increases over time. Actual path is denoted as P. As can be seen, the error accumulates with each successive step. In this case, the constructed map is a two-dimensional map of landmarks.

From a general point of view, we can identify three paradigms in the approach to solve the problem of simultaneous localization and mapping:

- Extended Kalman Filter (EKF) based
- Particle Filter (PF) based
- Graph based

As a rule, the choice of a solution is determined by the characteristics of technical realization of the robot, the external conditions, as well as the performance requirements for a software system.

III. Particle Filter Based Approach

A. FastSLAM Technique

Currently, most approaches to solve SLAM problem use techniques based on extended Kalman filter. The main disadvantage of this solution is that the computational complexity of the algorithm depends strongly on the number of landmarks. This is due to the fact that the covariance matrix of the filter P is of dimension $n \times n$, where n - number of landmarks. At each step of renovation matrix P, each element must be renewed, and therefore the complexity of the algorithm is about $O(n^2)$. Thus, the extended Kalman filter is most applicable in situations where the environment has a small number of stable tracked features, which in most practical cases, several hundred items. To solve this problem in 2002 Montemerlo, Thrun, Koller, and Wegbreit developed a new approach to solve the problem of simultaneous localization and mapping [1].

FastSLAM method separates the task into many equivalent subtasks, using the independence of individual elements of the SLAM model. FastSLAM algorithm based on the use of probabilistic model of Bayesian network. The diagram of this process is schematically represented in Fig. 2.

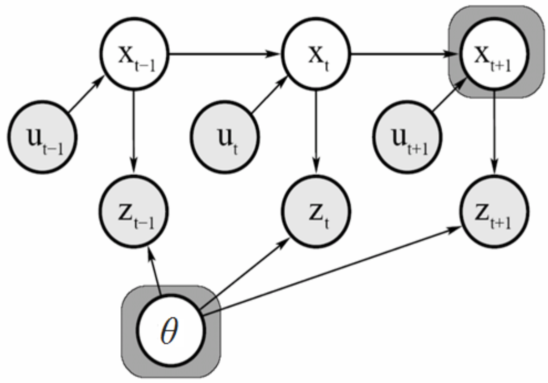

Fig. 2. Explanation of the SLAM problem

The robot moves from pose x_{t-1} through a sequence of controls $u_{t-1}..u_{t+1}$. Observable landmark θ has estimates $z_{t-1}..z_{t+1}$ at the time $t-1$ .. $t+1$. Each measurement is a function of the coordinates $z_1..z_n$ θ_i landmark, as well as the position of the robot at the time of measurement. As can be seen in the figure, the SLAM task involves important condition of independence of measurements from each other. Thus, this problem can be considered as n independent computational tasks to evaluate path of each of the landmarks. This observation is discussed in detail in [2] to develop an effective particle filter.

Based on the above mentioned conclusion, we can consider the problem of simultaneous localization of mobile robot and mapping as a problem, consisting of two parts: evaluation of robot trajectory and the landmarks position estimation, which in turn depend on the coordinates of the robot at the time of each measurement. Of course, in reality the position of the robot never known exactly, this is the very essence of the problem of SLAM. Nevertheless, the independence of landmarks from each other sufficiently motivated FastSLAM creators process each landmark separately.

The original version of the algorithm FastSLAM uses a modified particle filter for posterior estimating of the robot position. Each particle is characterized by a certain weight, determined by the state of *n* Kalman filters which are used to estimate the landmarks positions. This algorithm utilizes a Rao-Blackwellized representation of the posterior estimation, integrating particle filter and Kalman filter representations [3]. A naive implementation of this method has the algorithmic complexity equal to $O(MK)$, where M is the number of particles that uses a filter, and K - number of landmarks. Using tree structures storage complexity can be reduced to the value of $O(M \log K)$, which leads to significant performance benefits compared with SLAM solutions, based on extended Kalman filter.

B. Probabilistic Formulation of the SLAM Task

Now let's turn to the probabilistic formulation of the simultaneous localization and mapping task. Assume that the robot is in a one-dimensional space, and its position is characterized by a single variable x. Then $p(x)$ is the probability distribution of x, having a Gaussian shape. Now, if x represents the position of the robot and landmarks in a multidimensional space, the probability distribution $p(x)$ determines the probabilities of all possible state variables. Thus, the expression $p(x \mid \{u_0, u_1, \dots u_i\}, \{z_0, z_1, \dots z_i\})$ describes the probability of all the values of the system - sensor values and information about robot position at the time i.

The same role is played by the value of P_i and X_i in the extended Kalman filter, but there they are presented in a much more complex form. For convenience, let us introduce the notations $U_i = \{u_0, u_1, \dots u_i\}$ and $Z_i = \{z_0, z_1, \dots z_i\}$. The variable x in its turn characterizes the positions of the robot v and landmarks $p_0, p_1, \dots p_m$. The probability $p(x \mid Ui, Zi)$ can be represented as follows:

$$p(x \mid U_i, Z_i) = p(v, p_0, p_1, \dots p_m \mid U_i, Z_i). \tag{1}$$

To simplify the SLAM problem it is convenient to use the laws of the foundations of probability theory. Suppose that there are two independent random variables A and B. It can be said that $p(A, B) = p(A) * p(B)$. However, this expression is not valid for the case when A depends on B. In this case it will have the form $p(A, B) = p(A) * p(B|A)$.

It is known that estimation of landmark positions depends on the robot location, which means that probability $p(x \mid Ui, Zi)$ can be represented as follows:

$$\begin{aligned} &p(v, p_0, p_1, \dots p_m \mid U_i, Z_i) \\ &= p(v \mid U_i, Z_i) \cdot p(p_0, p_1, \dots p_m \mid U_i, Z_i, v). \end{aligned} \tag{2}$$

Due to the fact that the landmarks are independent from each other, that in the real world is observed in most cases, expression $p(p_0, p_1, \dots p_m \mid U_i, Zi, v)$ can be divided into m independent expressions:

$$\begin{aligned} &p(v \mid U_i, Z_i) \cdot p(p_0, p_1, \dots p_m \mid U_i, Z_i, v) \\ &= p(v \mid U_i, Z_i) \cdot p(p_0 \mid U_i, Z_i, v) \cdot \dots \cdot p(p_m \mid U_i, Z_i, v). \end{aligned} \tag{3}$$

Finally, the resulting expression for the probability distribution is given by:

$$p(x \mid U_i, Z_i) = p(v \mid U_i, Z_i) \cdot \Pi_m \, p(p_m \mid U_i, Z_i, v). \tag{4}$$

If we look at this expression, it becomes obvious that the problem of SLAM is divided into $m + 1$ tasks, and none of the landmark location estimates does not depend on others. This, in its turn, allows to solve the problem of polynomial complexity of the extended Kalman filter and avoid it in FastSLAM. The only price we have to pay for this simplification - is the risk to reduce precision associated with ignoring the correlation of landmarks estimation errors.

FastSLAM algorithm simultaneously track multiple possible paths, while the extended Kalman filter does not keep even one, but only works with the position of the robot - the last step of the current path. In its original form FastSLAM saves routes, but in the calculation uses only the previous step.

C. Localization Task

Particle filter is a modeling method for estimating the state of the system that cannot be fully observed. Particle filter keeps the weighted normalized set of sample states $S=\{s_1, s_2, \dots, s_m\}$, called particles. At each step, after getting the measurement o (or a vector of measurements), particle filter performs the following actions:

1) Creating new sample m of system model states $X' = \{x'_1, x'_2, \dots, x'_m\}$ from X states;
2) Jump to a new state in the Markov model of the robot position: $P(X'' \mid X')$. This action simulates the motion of the robot in the space;
3) Weighing of each state of Markov model according to observations;
4) Normalization of weights for a new set of states.

Particle filter is well suited for solving the problem of localization, where we need to track the position of the robot, which is a hidden value. Jump between the states is the movement of the robot, and observation is the result of visual odometry algorithm. Both of these values are very noisy. Motion model for the different robots with different environmental conditions can be quite different, but they are all will take into account the systematic and random errors one way or another.

The data about movement are used to predict the position of the robot. It can be obtained using visual odometry algorithm and the motion model of the robot. This model allows us to determine the next position of the robot using the values of the current state and the controls. For each particle position can be predicted as:

$$x_t^{[k]} = \begin{pmatrix} x_{t-1}^{[k]} + v_t \delta v \cos(\phi_{t-1} + \Delta\phi\delta\phi_a) \\ y_{t-1} + v_t \delta v \sin(\phi_{t-1} + \Delta\phi\delta\phi_a) \\ \phi_{t-1} + \Delta\phi\delta\phi_p \end{pmatrix}. \quad (5)$$

Values $\delta v, \delta\phi_a, \delta\phi_p$ are pseudo-random variables with Gaussian distribution, mean value equal to 1, and constant covariance. Thus, the collection of particles will have a distribution $p(x_t, M \mid U_t, Z_{t-1}, x_0)$. This distribution is often used in various particle filter implementations.

At the correction stage, map must be recalculated on the basis of the new measurement z_t. Each particle receives a definite weight using the current observations of the robot. In the pure localization problem, the robot has a total map in the memory. Each particle position corresponds to a known location and direction on the map. Hence, it is relatively easy to determine which values visual odometry algorithm should return, if the robot is in that position. Assuming that the values of the robot position and the locations of the landmarks can be considered as independent random variables, the total weight of the particle k will look like:

$$w^{[k]} = \frac{final\_distribution}{proposal\_distribution} = \frac{p(x_t^{[k]} \mid U_t, Z_t, x_0)}{p(x_t^{[k]} \mid U_t, Z_{t-1}, x_0)},$$

$$w_t^{[k]} = \frac{1}{\sqrt{2\pi L_t^{[k]}}} \exp\left(-\frac{1}{2}(z_t - \hat{z}_t^{[k]})^T \left(L_t^{[k]}\right)^{-1} \left(z_t - \hat{z}_t^{[k]}\right)\right), \quad (6)$$

$$L_t^{[k]} = \left(\Sigma_t^{x,n,[k]}\right)^T \left(P_t^{[k]}\right)^{-1} \left(\Sigma_t^{x,n,[k]}\right) + \bar{S}_t^{[k]}.$$

Here variable $\Sigma_t^{x,n,[k]}$ is a cross-covariance matrix. This value characterizes the correlation between the position of the robot and observation results. The process of resampling of the particles is shown in Fig. 3.

Resampling has a very strong impact on the performance of the algorithm as a whole. This process is as follows. Particles having smaller particle weights are replaced with larger weights. An effective amount of the particles is determined by the following criterion [4]:

$$N_{eff} = \frac{1}{\sum_{k=1}^{N}\left(\hat{w}^{[k]}\right)^2}, \quad (7)$$

where N - is the total number of particles, and value $\hat{w}^{[k]}$ means normalized weight of k particle.

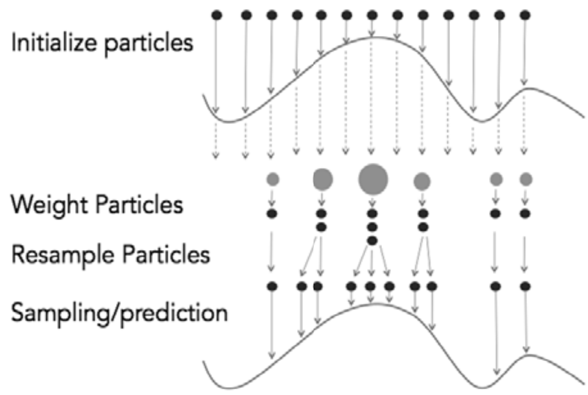

Fig. 3. Stages of particles processing - cycle from initialization of particles to the prediction step through resampling process

When the dispersion of the weights is increasing, the value N_{eff} decreases at the same time. Resampling process occurs when N_{eff} value falls below 50% of the total number of particles. This allows us to limit the growth of the error.

IV. Characteristics of the Camera System

Let's take a closer look at the parameters and geometry of the camera used in the problem of computing the trajectory of the mobile robot. It is necessary to enter a number of restrictions, allowing to set strict conditions for the mathematical interpretation of the problem.

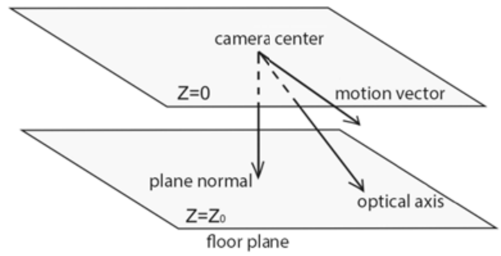

Fig. 4. Camera movement model in the system of monocular visual odometry

In this paper we consider the geometric model of pinhole camera - a camera obscura, which allows calculation without taking into account distortion caused by image blur out of focus. Along with an infinite depth of field of field, focal length can be considered only conditionally, in the process of shooting, it is a constant. Optical sensor of the camera is seen as an array of equal square pixels with zero slope. Fig. 4 shows a diagram of the geometry of the scene and a camera system.

This geometric interpretation of the camera system is used in most of the tasks of computer vision, particularly in robotics applications [5]. Since the height of the camera above the floor remains constant, it is convenient to go to a coordinate system in which the camera movement occurs in the plane $z = 0$, and the ground level is $z = z_0$. To describe the camera's orientation in space we can use three camera rotation parameters: R_φ, R_ψ and R_θ. Angle of rotation of the camera around the z-axis is characterized by R_φ. In the process of shooting the rotation angles about the axes x and y remains constant, while the angle φ and the position of the optical center of the camera $t = (t_x, t_y, 0)$ can vary from frame to frame. Thus, the position of the camera at the time t_0 determines the projection matrix for the resulting image as follows:

$$H = R_{\theta\psi} R_\varphi T_{xy} = const \cdot R_\varphi T_{xy} \tag{8}$$

where $R_{\theta\psi}$, R_φ, T_{xy} - rotation matrix and offset. The value of z_0 in this task is equal to 1, which does not impose restrictions on the decision, but only defines a global scaling factor.

IV. Evaluation of Interframe Homography

Let's look at the images obtained sequentially while moving the camera. Geometric characteristics of the system are similar to those described above paragraph. Let the global coordinate system is defined by the position and orientation of the camera at the time of the first image. In this case, the projection matrices P_1 and P_2, connecting these images may be defined as follows:

$$\begin{aligned} P_1 &= R_{\theta\psi} \cdot I_0, \\ P_2 &= R_{\theta\psi} \cdot T_{xy} \cdot R_\varphi \cdot I_{-t}. \end{aligned} \tag{9}$$

Thus, the point $X = [x\ y\ 1\ 1]^T$ on the plane $z = z_o$ is projected on the planes of the first and second images as:

$$x_1 = P_1 \cdot XR_{\theta\psi} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$

$$x_2 = P_2 XR_{\theta\psi} T_{xy} R_\varphi \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cdot R_{\theta\psi} R_\varphi \cdot \begin{bmatrix} x - t_x \\ y - t_y \\ 1 \end{bmatrix}, \tag{10}$$

$$T = \begin{bmatrix} 1 & 0 & -t_x \\ 1 & 0 & -t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

Based on the above conclusions homography matrix generally can be written as:

$$H = R_{\theta\psi} \cdot R_\varphi \cdot T_{xy} \cdot R^T{}_{\theta\psi}. \tag{11}$$

Thus, homography matrix can be determined with a finite accuracy for two partially overlapping images of the observing scene [6]. Since the rotation matrices $R_{\theta\psi}$ and R_φ are orthogonal, the matrix H and T must have the same condition number, this value is denoted as k.

Based on this conclusion we can assume that there is a formula that relates the value of the condition number k and the distance between the two frames. To find this relation we define k using eigenvalues $T^T \cdot T$. The characteristic polynomial for $T^T \cdot T$ is:

$$(\sigma - 1)(\sigma^2 - (2 + t_x^2 + t_y^2)\sigma + 1). \tag{12}$$

Singular value of T is equal to 1, as the multiplication of the other two singular numbers for which we have:

$$\sigma_1 \geq 1, \sigma_2 = 1, \sigma_3 = \frac{1}{\sigma_1} \leq 1, \tag{13}$$

therefore we can define k as follows:

$$k = \frac{\sigma_1}{\sigma_3} = \sigma_1^2. \tag{14}$$

Now let us introduce the value of the distance z between the images and define this value through the condition number:

$$\begin{aligned} k &= \sqrt{t_x^2 + t_y^2}, \\ k &= (1 + \frac{z^2}{2} + \frac{z}{2}\sqrt{4 + d^2})^2, \\ z &= \frac{\sqrt{k} - 1}{\sqrt[4]{k}}. \end{aligned} \tag{15}$$

This value characterizes the weights of the particles in the task of predicting the position of the robot.

Next, we have to validate proposed relationship between the traveled distance and the condition number of a homography matrix through experiments.

We use the test set of images obtained in an urban environment with a digital camera mounted on the vehicle [7]. For each pair of successive positions of the camera the actual traveled distance and the angle of the camera rotation are known. For images we can build homography matrix and calculate the distance using the last equation.

As can be seen from the Fig. 5, the data repeats general trend. Deviation from the true values obtained using global satellite positioning system, can be explained by several factors. The test suite is the actual data obtained from a digital camera, hence, external conditions (contrast lighting, extra movement in the image), and the characteristics of the optical system of the camera brings an additional error in the calculation results.

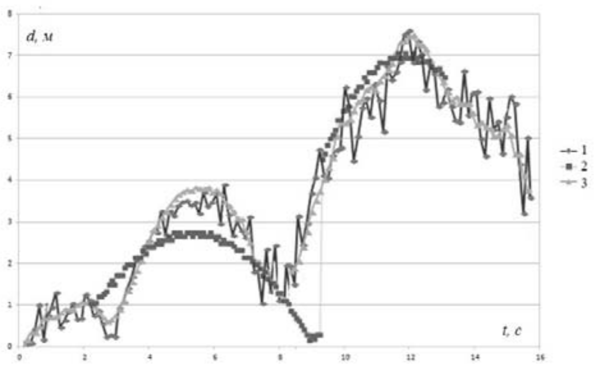

Fig. 5. The instantaneous values of the distance traveled in the time interval between adjacent frames: 1) estimated values, 2) actual values, 3) values obtained at the output of the unscented Kalman filter

V. Results

So, we have developed the method of finding the trajectory of motion using particle filter to estimate the position of the robot and the unscented Kalman filter to estimate the position of feature points in the images. The weights for the set of particles that characterize the expected position of the robot, are determined by the value of the matrix plane homography as projective mapping of points of the scene on the two-dimensional surface of camera sensor. An example of the resulting trajectory is shown in the Fig. 6.

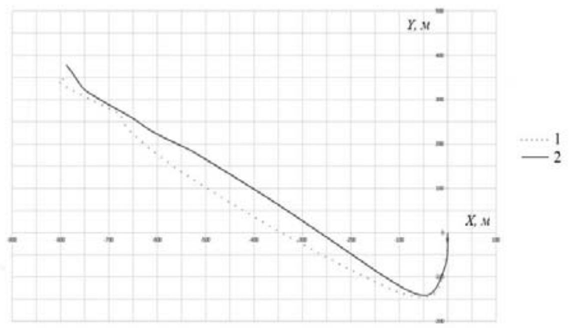

Fig. 6. The trajectory of the camera within 360 frames: 1) real path, obtained with the help of GPS, 2) estimated path, obtained using proposed method

Classical variant of the FastSLAM algorithm uses 2x2 versions of extended Kalman filter to estimate coordinates of landmarks. In our approach we solve nonlinear nature of the landmark movement using sigma-point type of Kalman filter. Tests show that the use of the unscented Kalman filter allows to increase the accuracy up to 5% compared with the extended Kalman filter.

However, one should mention that the results are highly dependent on external conditions, motion model and geometry of the system. In the future, to obtain more accurate estimates characterizing the accuracy and range of applicability of the proposed method, we plan to fulfill a series of tests on several test sets with different types of digital cameras and external conditions.

VI. Conclusion

In this paper we propose a method for solving the SLAM problem. Our method takes computational advantages of the FastSLAM algorithm. To estimate the position of the robot, we use a particle filter. The weights for the set of particles that characterize the expected position of the robot, are determined by the value of the plane homography matrix as the projective mapping of points of the scene on the two-dimensional surface of camera sensor. A set of unscented Kalman filters is used to estimate the positions of detected landmarks which are forming the map of the observed environment. Methods for detecting and description of landmarks were not considered in this paper, as it is beyond the scope of this work.

Research has shown that the movement of the camera between adjacent frames can be characterized by the condition number of the homography matrix. In the task of the landmarks locations estimation, unscented Kalman filters allows to increase the accuracy up to 5% compared with the extended Kalman filters. However, the results are highly dependent on external conditions, motion model and geometry of the system. Development of the SLAM algorithm that works well in all conditions, is a very urgent task today.

References

[1] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: *A factored solution to the simultaneous localization and mapping problem*. In AAAI, 2002
[2] K. Murphy. *Bayesian map learning in dynamic environments*. Computer Science Division. University of California. Berkeley, CA 94720-1776.
[3] K. Murphy and S. Russell. *Rao-blackwellized particle filtering for dynamic bayesian networks*. In Sequential MonteCarlo Methods in Practice, Springer, 2001
[4] A. Doucet, A. Freitas, N. Gordan, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001
[5] M. Wadenbäck, A. Heyden, *Ego-Motion Recovery and Robust Tilt Estimation for Planar Motion Using Several Homographies.* 9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2014), Proceedings of p.635-639
[6] R. Hartley, A. Zisserman *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2014
[7] J. Blanco, F. Moreno, J. Gonzalez *A Collection of Outdoor Robotic Datasets with centimeter-accuracy Ground Truth*. Autonomous Robots, 2009
[8] M. Wadenback, A. Heyden *Trajectory Estimation Using Relative*
[9] *Distances Extracted from Inter-Image Homographies*. Computer and Robot Vision (CRV), 2014 Canadian Conference on, Montreal, 2014, 232 - 237p.