MOBILE ROBOT CAMERA EXTRINSIC PARAMETERS AUTO CALIBRATION BY SPIRAL MOTION

Martins Ekmanis, Agris Nikitenko
Riga Technical University, Latvia
martins.ekmanis@edu.rtu.lv, agris.nikitenko@rtu.lv

Abstract. Self-localization in robotics is still a challenging task for both indoor and outdoor mobile robotic systems. The main reason is various sources of sensors and calculation errors that summarizes into a general position estimation error. While it is rather common to use some external landmarks to prevent error accumulation over time the sensors themselves used for landmark detection are significant sources of localization error. Within this paper we discuss one possible implementation of the localization mechanism, which uses an on-board camera on top of the robot for ceiling landmark detection and then through appropriate calculations estimates the actual position of the robot. While the camera is not set up perfectly and therefore has some unknown angular and displacement offset it generates a systematic error in the position estimation data. Self-calibration algorithm has to detect and compensate this offset automatically to maintain the necessary accuracy of the position estimation. Therefore, we propose an automatic calibration algorithm, which is based on spiral motion of the robot for data gathering and multi factor optimization for the error.

Keywords: robot indoor localization, auto-calibration, camera extrinsic parameters.

Introduction

Use of camera sensors for localization purposes is rather widely discussed and experiences a wide spectrum of implementations. The simplest ones are based on color tracking where each of the tracked or identified objects is marked by a previously defined color or a combination of color spots [1]. Here the main discussion object is proper use of data compression in order to preserve the unique features of the colors. In more advanced approaches a 3D vision is employed to produce depth data of the environment that allows to replace expensive sensors like Laser range finders by less costly CCD cameras. One of such methods is vSLAM (the name comes from Visual Simultaneous Localizations and Mapping) proposed by [2], which allows to track distinguished points in environment thereby producing robot displacement data necessary for position estimation. Unfortunately, not always use of two cameras is affordable due to high memory and computing power demands. Therefore, some single-camera alternatives have been developed during the last decade like EKF extended vSLAM described in [3]. There are good examples on using lightweight visual SLAM algorithms in conjunction with visual recall or visual memory algorithms that allows to embed them into mobile computing devices [4]. Another option is to use artificial or natural landmarks that can be considered to some extent as a visual memory special case. In this case the system has to identify the landmark and use this information for guiding the robot or localize with higher confidence [5]. Usually these methods are well suited for indoor systems and can be easily combined with other sensor systems like odometric localization sensors [6]. This allows to reduce the overall computation load and simplify localization algorithms [4]. However, sensor errors are still actual and need to be addressed. Therefore, we propose to use automatic camera calibration that can reduce the localization error through knowing the actual camera offset values.

The rest of the paper is organized as follows: section II describes the used sensors and defines the calibration task, its solution and proposed algorithm, section III describes the practical implementation on robotic platform and the acquired experimental data, sections IV and V propose conclusions and discussions of further work along with a list of the used references.

Formal definition of the calibration problem

In the computer vision camera calibration is required to relate the pixel coordinates (rows and columns) to the x, y, z coordinates in 3D world. It is common to split the camera parameters in two separate groups, intrinsic and extrinsic [7]. The first depends only on the camera itself, like optical distortions, sensor displacement etc. They are constant for the given camera and do not change while the camera moves [7]. The second includes the position and orientation of the camera coordinate frame relative to the world coordinate frame.

Intrinsic camera parameters are the image pixel coordinates of the principal point o_c, o_r and focal length $f_x = \lambda/s_x$, $f_y = \lambda/s_y$, where λ is the distance between the origin of the camera frame and image plane; s_x, s_y are the image plane horizontal and vertical dimensions [7]. These parameters can be easily detected by "chessboard" camera calibration algorithms, like proposed by Zhangs [12] and implemented in OpenCV library [8].

Extrinsic parameters consist of translation T and rotation R transformations relating the camera coordinates to the world coordinates [7].

$$p^c = Rp^w + T , \tag{1}$$

where p^c – represents the object coordinates in the camera coordinate frame;
p^w – the same object coordinate in the world coordinate frame.

In static robotics manipulators these parameters are calculated from kinematics equations of the manipulator (eye in hand systems) [10]. In mobile robots the whole body of the robot moves and makes it significantly more difficult to separate rotation and translation caused by the body movement R_b, T_b from constant rotation and translation caused by the camera placement R_c, T_c on the robot's body.

A mobile robot usually has more than one sensor for position estimation and appropriate data fusion methods are implemented for the final estimate. For obvious reasons it is good to know the exact values of variance for more accurate data fusion but usually there is no error-free reference to measure it. There are methods allowing some calibration and estimates can be made using appropriate mathematical calculations on data gathered during periodic rotation near the reflective object and fusing them with IMU (Inertial Measurement Unit) data. As external measurement artificial or natural landmarks are used [9]. Others, like the camera extrinsic parameters, are more difficult to obtain due to lack of reference measurements. We focus on using artificial landmarks because they provide a reliable source of data needed for calibration and are easy to use in practical applications.

Let us assume that observation of known landmark provides a straight-forward means for robot position p calculations. However, in practice the camera has limited resolution, the robot vibrates during motion, the camera position and angle relatively to the expected are not matching, etc. All this will affect position estimation in terms of disturbances around the true values. Therefore, the actually observed or calculated position we assume being a Gaussian variable $p = (x, \mu_c, \sigma_c^2)$, where misperception of the camera position and rotation is reflected by variance σ^2.

If comparing two position estimates p and g that are calculated from independent sources the delta should have zero mean (assuming that the initial position is aligned among alternative position sources) and nonzero variance (2). It is not known, which of the sources is responsible for this deviation but changing the variance for one of the estimates will change their common variance as well.

$$V(p-g) = Var(p) + Var(g) - 2Cov(p,g), \tag{2}$$

where $Cov(p, g)$ – covariance of p and g position estimates.

Assuming that the position estimates are independent, they do not correlate, therefore, their covariance is zero $Cov(p, g) = 0$ and this variable can be omitted from equation (3).

$$V(p-g) = Var(p) + Var(g). \tag{3}$$

The calibration task itself is formulated as a search for the extrinsic parameters EP, which minimize the common variance (4).

$$EP^* = \arg\min_{EP} Var(p \mid EP - g), \tag{4}$$

where EP^* – optimal extrinsic parameters to minimize the equation;
$p|EP$ – camera position estimate with known EP;
g – alternative position estimate (inertial sensors, odometry).

From (4) it follows that this is a multi-parameter optimization task, where it is important to define constraints and importance of the parameters within the optimization task.

The camera as any object in 3D space may have at least 6 degrees of freedom – three rotational and three translational [7]. Moreover, the robot location is not known before calibration what gives three more variables to find. If one intends to optimize the whole set of the parameters it might be too costly in terms of time and computing power for a mobile robot. In the particular implementation let us assume then the landmark is located at origin of the world coordinate frame (Fig. 1). The camera coordinate frame is attached by a solid link to the robot body and is assumed to be unknown but constant. The ground plane and ceiling plane are assumed to be parallel, therefore $z = z' - z''$ distance between the camera and landmark is constant as well where z'' is fixed z offset of the camera from the ground frame. The robot moves on the ground plane and its body has only one rotational degree (roll), and two movement degrees (x, y) of freedom, therefore, for the location task the camera yaw, pitch and z offset have indirect influence. Camera x offset dx_c and pitch θ together with z distance summarize to single offset $dx_r = f(dx_c, \theta, z)$ in the robot coordinate frame (Fig. 1). Identically camera y offset dy_c and yaw ψ together with z summarize to single offset $dy_r = f(dy_c, \psi, z)$. As all arguments are constant, it is assumed that dx_r := const and dy_r := const making the optimization task much simpler as we can directly focus on finding dx_r and dy_r instead of five other parameters.

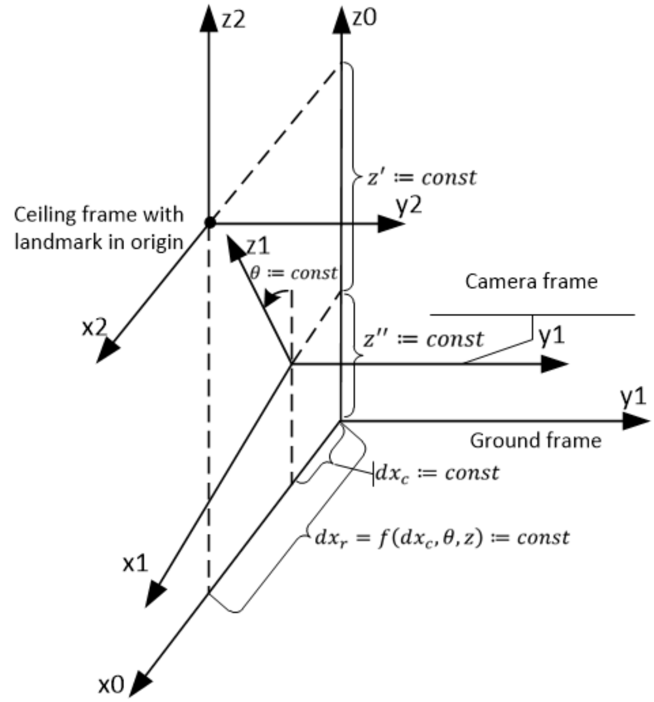

Fig. 1. **Coordinate frames and main parameters explained**

Unfortunately, the required offsets cannot be calculated from a single set of measurements, as there is not known the initial robot position in the ground plane. Several independent measurements should be made, therefore the robot cannot be in fixed position during calibration. Trajectory for the calibration must be chosen to observe the landmark from different angles and positions (it is common for most camera calibration procedures) [12; 13].

Moreover, each measurement has its own error, where the camera has normal error distribution whereas odometry has "banana distribution" [15]. As it is known, the odometry error accumulates over time and its distribution depends on the trajectory chosen [14]. If the robot moves straight (Fig. 2. a), the odometry error distribution does not satisfy Chi-square test against normal distribution (hypothesis was tested in simulated environment with 5 % significance level). To solve the proposed optimization task (4) the trajectory should be chosen to satisfy Chi-square test in the same time minimizing odometry variance and providing as much as possible unique (in means of position and angle) measurements. From another perspective the trajectory should be chosen so that the robot keeps seeing the landmark all the time.

The circle trajectory (Fig. 2. b) satisfies Chi-square test starting from ~180º of the first circle but keeping drive by circle will give non unique positions for every loop. Therefore, a more complicated trajectory was chosen. Spiral motion gives a unique position for every measurement, the same keeping normal error distribution and the landmark keeps visible long enough to collect the required amount of measurements.

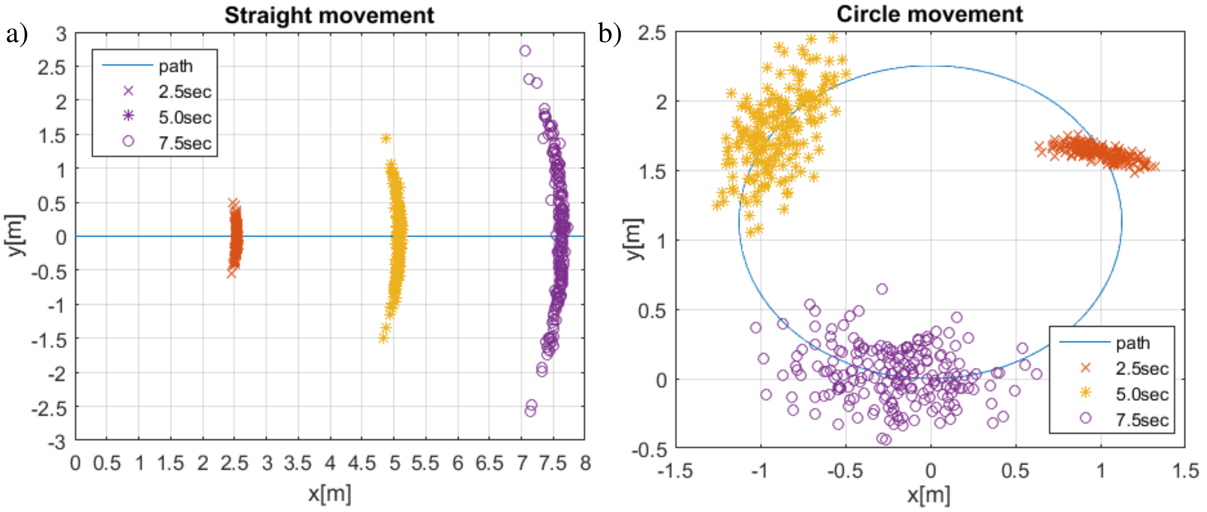

Fig. 2. **Odometry error distribution:** a – in straight movement; b – in circle movement

Experimental data together with dx_r, dy_r variation over some offset interval show that the error surface (Fig. 3) has only one extreme. Therefore, gradient descent or a similar method can be used to obtain the optimal offset values [11].

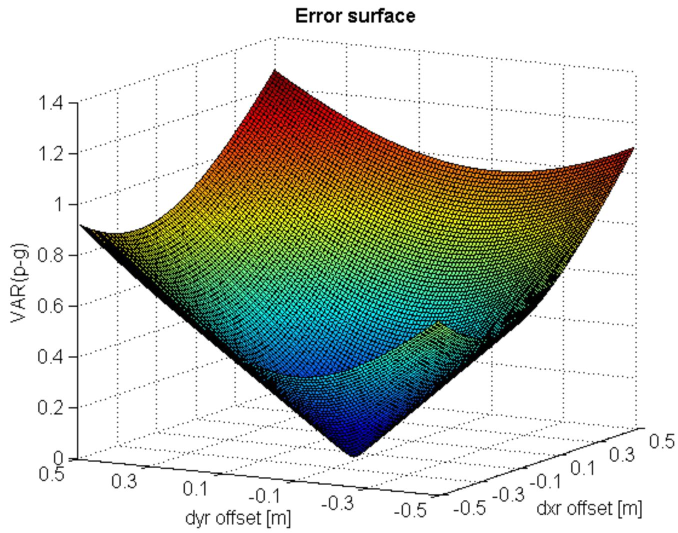

Fig. 3. **Numeric calculation of error surface**

Knowing the most important features of the error surface – a single extreme problem, methods that employ random sampling are not necessary. Instead, systematic search algorithms would give more confidence of the expected result. Another advantage is time saving during the optimum search while *g* position estimate accumulates error over time during the search itself.

The proposed algorithm can be described as a pseudo code function FIND-EXTRINSIC (Fig. 4) where arguments are odometry and landmark traces together with three configurable parameters. The first parameter ε represent search stopping threshold. The second parameter α limit meaningful range to find parameters and third s is search step. According in experimental part 0.01, 0.5 and 0.025 values were used.

Sub function CALCSTD (Fig. 5) takes as arguments odometry and landmark traces, project data to common time axis by piecewise linear interpolation and calculates the resulting variance between independently obtained positions.

```
function FIND-EXTRINSIC (O_t,L_t,ε,α,s) return Extrinsic
        inputs:   O_t odometry and L_t observed landmark traces
                  ε min value change to stop search
                  α max meaningful delta to search within
                  s step value
        local:    EXT[i] n-attribute extrinsic
                  V_min, V_max attribute min-max values
                  V_mid atribute middle value
                  V_prev attribute previous value
        for each index i in EXT do
                  V_min ← – α; V_max ← α; V_mid ← 0; V_prev ← α
                  while |V_mid – V_prev| > ε
                            EXT[i] ← V_mid + s; ε_1 ← CALCSTD(O_t,L_t,EXT)
                            EXT[i] ← V_mid – s; ε_2 ← CALCSTD(O_t,L_t,EXT)
                            if ε_1 > ε_2 then V_max ← V_mid else V_min ← V_mid
                            V_prev ← V_mid
                            V_mid ← (V_min + V_max)/2
                  EXT[i] ← V_mid
return EXT
```

Fig. 4. **FIND-EXTRINSIC function pseudo code**

```
function CALCSTD (O_t,L_t,EXT) return Variance
        inputs:   O_t odometry trace
                  L_t landmark trace
                  EXT extrinsic values
        local:    t[i] discretication time points
                  P_O[i], P_L[i] position estimates arrays
                  δ^2 ← 0 variance
        foreach time t in t[i]
                  P_O[i] ← position calculated from interpolated O_t|t
                  P_L[i] ← position calculated from interpolated L_t|t,EXT
        foreach attribute a in P[i]
                  δ^2 ← δ^2 + std(P_O[i][a] – P_L[i][a])
return δ^2
```

Fig. 5. **CALCSTD function pseudo code**

Practical implementation and experimental results

For practical implementation a well known vacuum cleaning robot platform Roomba 580 was selected. This robot has a pre-set function allowing to clean some small area – spot. In general the trajectory follows the spiral pattern that makes it a very convenient choice for experimental purposes. The motion pattern is as follows: first the robot moves 20 cm straight, then rotates by 360 degrees and then moves 10 cm following the spiral trajectory as depicted in Fig. 6.

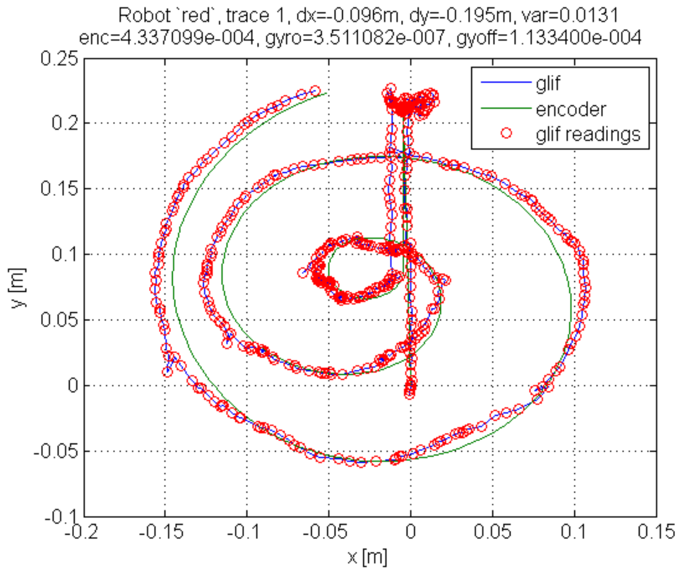

Fig. 6. **Camera and odometry position estimate during Roomba "spot" action**

Thereby, as discussed above, the motion pattern is useful for automatic calibration. However, we believe that in general the pattern is rather easy to implement for most of indoor mobile robot platforms. In Fig. 6 "glif" is a special ceiling marker (sign recognized by the robot) used for positioning. The algorithm is implemented in C++ and runs on Raspberry Pi (RPI), mounted on Roomba 580 platform (Fig. 7).

Fig. 7. **Roomba 580 with Raspberry Pi add-on**

The landmark detection was performed by RPI camera at 640x480 px resolution at 30 fps rate. Each calibration maneuver took ~20 sec, where 16 sec were active data capture for calibration.

The experimental setup includes four robots where eight calibration traces were made per each robot. The robot starting position was changed after each trace. Each robot's gyro and encoders were calibrated once before the experiment and the same values were used for all eight traces.

Calibration data (table 1) clearly show the difference between the robots. Fig.8. indicates that the calculated offset is robot specific and calibration algorithm gives repeatable results.

Despite the fact that some calibration data were of weak quality because of missing or delayed readings, the offset variance is near camera resolution (1 px ~0.5 cm at 3 m). Moreover, bad or missing data (temporary light conditions prevent from landmark detection, wheel slip, etc.) can be easily detected by the variance found after gradient descent algorithm. If the value is too high the robot can automatically repeat calibration instead of using lower quality values.

Table 1

Experimental calibration results

Robot name	Trace id	*dx*, m	*dy*, m	var
red	1	-0.0955	-0.1950	0.0131
red	5	-0.0922	-0.1965	0.0121
red	2	-0.0990	-0.1970	0.0161
red	6	-0.0948	-0.1970	0.0126
red	3	-0.0988	-0.1965	0.0172
red*	7	-0.0703	-0.1913	0.0702
red	4	-0.0932	-0.1945	0.0113
red	8	-0.0850	-0.1930	0.0156
blue	1	-0.0167	-0.1210	0.0171
blue*	5	-0.0373	-0.1218	0.0664
blue	2	-0.0255	-0.1192	0.0264
blue	6	-0.0170	-0.1230	0.0358
blue	3	-0.0145	-0.1205	0.0211
blue	7	-0.0177	-0.1165	0.0223
blue	4	-0.0142	-0.1208	0.0160

Table 1(continued)

Robot name	Trace id	*dx*, m	*dy*, m	var
blue	8	-0.0197	-0.1198	0.0351
black	1	-0.1195	-0.0665	0.0197
black	5	-0.1135	-0.0677	0.0199
black	2	-0.1208	-0.0697	0.0240
black	6	-0.1145	-0.0690	0.0247
black	3	-0.1165	-0.0623	0.0141
black*	7	-0.0992	-0.1052	0.0846
black	4	-0.1142	-0.0637	0.0148
black*	8	-0.1377	-0.0613	0.0989
white	1	-0.0395	-0.0862	0.0175
white*	5	-0.0707	-0.0887	0.0657
white*	2	-0.0653	-0.0870	0.0590
white*	6	-0.0805	-0.0912	0.0666
white	3	-0.0385	-0.0810	0.0181
white	7	-0.0553	-0.0870	0.0427
white	4	-0.0347	-0.0840	0.0143
white*	8	-0.0792	-0.0872	0.0741

* Some bad or missing data in trace

As it seen from the experimental results (Fig. 8), the extrinsic parameters are robot specific and therefore must be calibrated at least once when deploying the robot to its workspace. However, any robot collisions, work in high physical stress environment, transportation, vibration, maintenance etc. can cause offset changes and therefore the authors recommend to recalibrate the robot periodically.

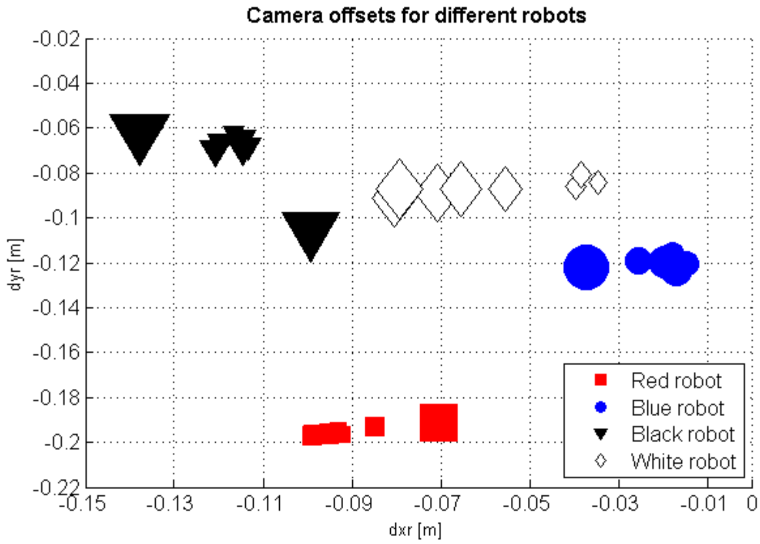

Fig. 8. **Detected camera offsets, where size represents calibration variance**

Conclusions and future work

The proposed auto calibration algorithm provides an affordable method (from time and memory prospective) to reduce a systematic error of self-localization caused by imperfect camera set for applications where the ceiling markers are used for indoor mobile robot position estimation. The conducted set of experiments shows that the proposed algorithm reveals specific offsets for each of the used robots thereby providing the necessary data to reduce the overall position estimation error.

Another important aspect proposed here is use of the spiral motion pattern for calibration data gathering, what helps to eliminate errors caused by the motion pattern symmetric properties or reoccurrence during calibration. While the spiral pattern is more complex than quadratic or circular patterns, it is still rather straight forward to implement for most of the indoor mobile robot platforms or use the ones implemented by the robot producers like iRobot's Roomba vacuum cleaners have.

At the moment we use dedicated ceiling markers for calibration purposes but for the future we see it possible to extend the algorithm for fully autonomous calibration and use of an arbitrary object in environment that can be recognized by the robot. Thereby, the algorithm will have wider application extending the application to outdoor mobile robots as well.

At the same time our intention is to extend the application to any camera used by the robot, but this will introduce more unknown variables and might require more complex optimizations with risk to lose its practical value for embedded robotic systems due to time and memory requirements.

Acknowledgments

This research has been supported by the Latvian state research program 2014-2017 project “The Next Generation of Information and Communication Technologies (NexIT)”.

References

1. Said T., Ghoniemy S., Ismail G., El-Ghafar I.A. Multi-Object Color Tracking for Multi-Robot Systems Enhancemnet. Proceedings of 13th International Conference “Aerospace science & Aviation Technology”, May 26-28, 2009, Cairo, Egypt.
2. Karlsson N., Di Bernardo E., Ostroqski J., Goncalves L., Irjanian P., Munich M. The vSLAM Algorithm for Robust Localization and Mapping. Proceedings of International Conference “Robotics and Automation (ICRA)”, April 18-22, 2005.
3. Benmessaoud M.L., Lamrani A., Nemra K., Souici A.K. Single-Camera EKF-vSLAM. World Academy of Science, Engineering and Technology, 2008, pp. 307-312.
4. Milford M.J., Achill F., Corke P., Mahony R., Wyeth G. Aerial SLAM with a Single Camera Using Visual Expectation. Proceedings of International Conference “Robotics and Automation (ICRA)”, Shanghai, China, 2011, pp. 2506-2512.
5. Yoon-Gu K., Jinung A., Kee-Dong L. Localization of Mobile Robot Based on Fusion of Artificial Landmark and RF TDOA Distance under Indoor Sensor Network. Proceedings of International Conference “Advanced Robotics”, 2011, pp. 203-211.
6. Nikitenko A., Liekna A., Ekmanis M., Kuļikovskis G., Andersone I. Single Robot Localization Approach for Indoor Robotic Systems through Integration of Odometry and Artificial Landmarks. Applied Computer Systems, vol. 14, 2013, pp. 50-58.
7. Spong M.W., Hutchinson S., Vidyasagar M. Robot Modeling and Control. USA: John Wiley & Sons, Inc., 2006.
8. Bradski G., Kaehler A. Learning OpenCV. CA: O'Reilly Media Inc., 2008.
9. Ekmanis M. Self-Calibration in Differential Drive Dynamics/Kinematics Model. Proceedings of the 2013 International Conference “Systems, Control and Informatics (SCI 2013)”, September 28-30, 2013, Vencie, Italy, pp. 145-150.
10. Strobl K.H., Hirzinger G. Optimal Hand-Eye Calibration. Proceedings of the IEEE/RSJ International Conference “Intelligent Robots and Systems (IROS 2006)”, 2006, Beijing, China, pp. 4647-4653.
11. Snyman A. Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing, 2005.
12. Zhang Z. A flexible new technique for camera calibration. IEEE Transactions “Pattern Analysis and Machine Intelligence", vol. 22(11), 2000, pp. 1330-1334.
13. Heikkila, J. Geometric camera calibration using circular control points. IEEE Transactions “Pattern Analysis and Machine Intelligence”, vol. 22, 2000, pp. 1066-1077.
14. Long A.W., Wolfe K.C., Mashner M.J., Chirikjian G.S. The Banana Distribution is Gaussian: A Localization Study with Exponential Coordinates. Robotics: Science and Systems, 2012.
15. Mächler P. Robot odometry correction using grid lines on the floor. Proceedings of 2nd International Workshop “Mechatronic Computer Systems for Perception and Action”, 1997, Pisa, Italy.