On Augmenting the Visual SLAM with Direct Orientation Measurement Using the 5-Point Algorithm

Received 10th October 2012; accepted 22nd November 2012.

Adam Schmidt, Marek Kraft, Michał Fularz, Zuzanna Domagała

Abstract:
This paper presents the attempt to merge two paradigms of the visual robot navigation: Visual Simultaneous Localization and Mapping (VSLAM) and Visual Odometry (VO). The VSLAM was augmented with the direct, visual measurement of the robot orientation change using the 5-point algorithm. The extended movement model of the robot was proposed and additional measurements were introduced to the SLAM system. The efficiency of the 5-point and 8-point algorithms was compared. The augmented system was compared with the state of the art VSLAM solution and the proposed modification allowed to reduce the tracking error by over 30%.

Keywords: *visual SLAM, visual odometry*

1. Introduction

The ability to work in the unexplored environment plays a crucial role in the operation of a mobile robot. Recently an increasing attention has been paid to the visual navigation systems due to the decreasing price and increasing quality of the cameras, relatively simple mathematical models and high information density of images. The main drawback of such an approach is the inability to measure the depth of the unknown scene using a single camera. The visual navigation systems can be divided into two main categories: visual simultaneous localization and mapping (VSLAM) and visual odometry (VO).

The VSLAM methods focus on building a map of an unknown environment and use it to calculate the position of the robot in real-time. As they are mostly used for tracking long trajectories the feature map is relatively sparse and only few measurements are used to update the estimate of environment state. The first real-time VSLAM system was proposed by Davison and Murray [1]. Sim *et al.* [2] presented a large scale VSLAM system using the SIFT point detector and descriptor and the particle filter. Sturm and Visser [3] showed the visual compass allowing for the fast estimation of the robot's orientation. At the moment the MonoSLAM [4] is considered to be one of the most successful VSLAM systems. A modification of the MonoSLAM adapted for the hexapod robot was presented in [5].

The main purpose of the VO methods is to estimate the precise trajectory of the robot without the map of the environment. The in-depth survey of the VO algorithms has been recently presented in [6, 7]. The VO systems use relatively large number of measurements to estimate the transformation between the consecutive robot positions. Due to the necessary processing the VO algorithms tend to be slower, though there have been some attempts to increase their speed with the FPGA implementation [8].

This paper presents the fusion of both the approaches in order to improve the precision of the VSLAM system by augmenting it with the visual estimation of the robot's orientation change. The orientation change is measured using the 5-point algorithm [9] every few steps of the VSLAM system. The proposed algorithm was evaluated using the sequences registered during the Rawseeds Project [10]. The augmented VSLAM system is presented in the section 2 and the section 3 presents the concluding remarks.

2. Augmented VSLAM system

2.1. Environment

The environment was modeled using the probabilistic, feature-based map adapted from the works of Davison [1, 4]. The map contains the information on the current estimates of the robot and features position as well as the uncertainty of those estimates. The state vector x contains the state of the robot and the features while the covariance matrix P represents the state uncertainty modeled with multidimensional Gaussian:

$$x = \begin{bmatrix} x_r \\ x_f^1 \\ x_f^2 \\ \vdots \end{bmatrix}, P = \begin{bmatrix} P_{x_r x_r} & P_{x_r x_f^1} & P_{x_r x_f^2} & \cdots \\ P_{x_f^1 x_r} & P_{x_f^1 x_f^1} & P_{x_f^1 x_f^2} & \cdots \\ P_{x_f^2 x_r} & P_{x_f^2 x_f^1} & P_{x_f^2 x_f^2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{1}$$

where x_r is the state of the robot and x_f^i is the state vector of the i-th feature. The Extended Kalman Filter is used to update the probabilistic map.

The basic movement model, the 'agile camera' was proposed by Davison [1, 4]. The state vector x_r consists of the robot's Cartesian position r, the quaternion q representing the robot's orientation, linear velocity v and angular velocity ω:

$$x_r = \begin{bmatrix} r & q & v & \omega \end{bmatrix}^T \tag{2}$$

The robot's model was extended with the additional quaternion q_m describing the orientation of the robot remembered during the system initialization or the last measurement of the orientation change:

$$x_r^m = \begin{bmatrix} x_r & q_m \end{bmatrix}^T \tag{3}$$

The Inverse Depth representation of the point features was used [11] where the state of each feature consists of the camera position during the feature initialization, angles describing the bearing of the line passing through the point feature and camera center and the inverse of the depth:

$$x_f^i = \begin{bmatrix} x_0^i & y_0^i & z_0^i & \phi^i & \theta^i & \rho^i \end{bmatrix}^T \tag{4}$$

2.2. Movement model and prediction

During the prediction stage of the EKF it is assumed that the robot is the only dynamic element of the environment. Therefore, the estimates of the point features' positions do not change during the prediction stage. At each iteration the robot is affected by random, normally distributed linear (a) and angular (α) accelerations resulting in a velocity impulse:

$$\begin{bmatrix} V(k) \\ \Omega(k) \end{bmatrix} = \begin{bmatrix} a(k) \\ \alpha(k) \end{bmatrix} \Delta T \tag{5}$$

The prediction function of the 'agile camera' model takes the following form:

$$\begin{aligned} x_r(k+1|k) &= f\left(x_r(k|k), \Delta T, a(k), \alpha(k)\right) \\ &= \begin{bmatrix} r(k|k) + (v(k) + V(k))\Delta T \\ q(k|k) \times q_\omega(k) \\ v(k|k) + V(k) \\ \omega(k|k) + \Omega(k) \end{bmatrix} \end{aligned} \tag{6}$$

where $q_\omega(k)$ is the incremental rotation quaternion and $\times$ stands for the Grassmann product.

The prediction function of the new model takes two forms depending on the orientation measurement state. If the orientation change was not measured in the last iteration the q_m remains unchanged (Eq. 7). Otherwise its value is replaced with the current estimate of the robot orientation (Eq. 8).

$$\begin{aligned} x_r^m(k+1|k) &= f_1\left(x_r^m(k), \Delta T, a(k), \alpha(k)\right) = \\ &= \begin{bmatrix} f\left(x_r(k|k), \Delta T, a(k), \alpha(k)\right) \\ q_m(k) \end{bmatrix} \end{aligned} \tag{7}$$

$$\begin{aligned} x_r^m(k+1|k) &= f_2\left(x_r^m(k), \Delta T, a(k), \alpha(k)\right) = \\ &= \begin{bmatrix} f\left(x_r(k|k), \Delta T, a(k), \alpha(k)\right) \\ q(k) \end{bmatrix} \end{aligned} \tag{8}$$

2.3. Measurement

In the basic version of the MonoSLAM the state vector is updated according to the visual observations of the point features [4] (Figure 1). In the proposed system the observation vector is extended with the visual measurement of the robot orientation change. The observation vector h takes the following form:

$$h = \begin{bmatrix} h_1 & \dots & h_N & q^h \end{bmatrix}^T \tag{9}$$

where $h_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^T$ stands for the observation of the i-th point feature on the image plane and q^h is the quaternion describing the orientation change defined as:

$$q^h = (q^m)^* \times q \tag{10}$$

$$(q^m)^* = \begin{bmatrix} q_a^m & -q_b^m & -q_c^m & -q_d^m \end{bmatrix}^T \tag{11}$$

where $\times$ is the Grassmann product and $(q^m)^*$ is the conjugate of the q_m. The measurement procedure of q_m is presented in the section 2.4.

2.4. Orientation change estimation

In computer vision, the fundamental matrix $\mathbf{F}$ and the essential matrix $\mathbf{E}$ are the rank 2 matrices of 3×3 size, that relate the corresponding point pairs across two views of the same scene. If the homogeneous image coordinates of the projection of a scene point $\mathbf{X}$ on the first image is given by $\mathbf{x}$, and the projection of the same point on the second image is given by $\mathbf{x}$, then every corresponding point pair $\mathbf{x} \leftrightarrow \mathbf{x}'$ is tied by the relation given by equation 12 [12].

Fig. 1. The exemplary point features measurements with the uncertainty ellipses.

$$\mathbf{x'^T F x} = 0 \tag{12}$$

The essential matrix $\mathbf{E}$ is related to the fundamental matrix $\mathbf{F}$ as given in (13).

$$\mathbf{E} = \mathbf{K'^T F K} \tag{13}$$

The matrices $\mathbf{K}$ and $\mathbf{K}'$ are the camera calibration matrices, so for the monocular case $\mathbf{K} = \mathbf{K}'$. The knowledge of essential matrix allows for the determination of relative pose between camera positions at which the images were registered, i.e. the rotation vector $\mathbf{R}$ and the translation vector $\mathbf{t}$ (up to an unknown scale).

The essential matrix satisfies the equation (12) only if the corresponding point pair coordinates have been normalized. i.e. the raw registered coordinates of points have been multiplied by the respective camera matrices and lens distortions have been corrected. The components of (12) can then be written as:

$$\mathbf{E} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \mathbf{x}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}.$$

From this, for a single point pair we get:

$$\begin{aligned} x'xe_{11} + x'ye_{12} + x'e_{13} + y'xe_{21} + y'ye_{22} + \\ + y'e_{23} + xe_{31} + ye_{32} + e_{33} = 0. \end{aligned} \tag{14}$$

Writing the elements of $\mathbf{E}$ as a column, in the row by row order allows to write (14) as (15):

$$\begin{bmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{bmatrix} \mathbf{e} = 0. \tag{15}$$

For any given number of point pairs n, their corresponding equations as given in (15) can be stacked forming a matrix $\mathbf{A}$, resulting in a system as given in equation (16).

$$\mathbf{Ae} = 0. \tag{16}$$

In practice, the solution to the system of equations given by (16) is found using singular value decomposition (SVD). The decomposition is performed on the matrix $\mathbf{A}$ – $(SVD(\mathbf{A}) = \mathbf{UDV}^T)$. The solution in the least squares sense is then the right singular vector corresponding to the smallest singular value. By using 8 point pairs for the construction of the matrix $\mathbf{A}$, solution is obtained directly from the SVD and is given by the right singular vector corresponding to the smallest singular value.

The minimum number of point pairs allowing for the computation of the essential matrix is five, but in this case additional constraints given by the equations (17) and (18) must be taken into account. This is referred to as the 5-point algorithm.

$$det(\mathbf{E}) = 0 \tag{17}$$

$$2\mathbf{EE}^T\mathbf{E} - trace(\mathbf{EE}^T)\mathbf{E} = 0 \tag{18}$$

The method used for essential matrix computation was based on the work presented in [13]. The process starts with performing the SVD for the computation of the 4-dimensional nullspace over the matrix $\mathbf{A}$ constructed using five point pairs. The essential matrix is a linear combination of the four singular vectors corresponding to the four singular values that are equal to zero:

$$e = xe_1 + ye_2 + ze_3 + we_4 \tag{19}$$

where e_i are the vectors spanning the nullspace, and x, y, z and w are some scalars. As the essential matrix is determined up to scale, w can be substituted with 1. Substituting e to (17) and (18) gives ten 3rd degree polynomial equations, consisting of 20 monomials. The monomials arranged according to GrLex order constitute the X monomial vector, allowing to rewrite the system of equations in the form given in (20).

$$MX = 0 \tag{20}$$

The matrix M is some 10×20 matrix. According to [13], the system of polynomial equations can be solved by defining a so called Gröbner basis and using it for action matrix computation. The action matrix has a size of 10×10. The Gröbner basis is obtained by performing the Gauss-Jordan elimination over (20):

$$\begin{bmatrix} I & B \end{bmatrix} X = 0 \tag{21}$$

The action matrix is constructed by choosing appropriate columns from the matrix $\begin{bmatrix} I & B \end{bmatrix}$ after the elimination was performed. The solution of the system of equations is encoded in the left singular vectors of the action matrix corresponding to the real eigenvalues.

As the point pairs are detected and matched automatically, the set of matched points contains a certain fraction of outliers (false matches). Even a single incorrect match used as input data for the 5-point algorithm results in producing incorrect output. To deal with this issue, the 5-point algorithm is applied within the robust estimation framework based on the RANSAC algorithm [14]. The 5-point algorithm has higher computational requirements than the 8-point algorithm. However, the 8-point algorithm requires more RANSAC iterations, especially when the fraction of outliers is high. Furthermore, it cannot deal with the case in which the points used for computation are coplanar (degenerate configuration).

To solve for rotation and translation, the method proposed in [12] was used. The SVD decomposition of the essential matrix $\mathbf{E}$ is in ideal case given by the equation (23). The singular values s_1 and s_2 have equal values, and the smallest singular value equals 0, as $\mathbf{E}$ is rank-deficient and of rank 2.

$$SVD(\mathbf{E}) = \mathbf{UDV}^T, \text{ where } \mathbf{D} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{22}$$

In practice however, due to the image noise, quantization and numerical inaccuracies, this is not the case. There are three nonzero singular values – one with a value close to zero (s_3), and the other two with significantly greater value, which are roughly similar (s_1, s_2). To improve the accuracy of the rotation and translation estimation, rank 2 can be imposed onto the matrix $\mathbf{E}$ by setting the lowest singular value to zero and substituting the other two singular values with their average:

$$SVD(\hat{\mathbf{E}}) = \mathbf{U}\hat{\mathbf{D}}\mathbf{V}^T, \text{ where } \hat{\mathbf{D}} = \begin{bmatrix} \frac{s_1+s_2}{2} & 0 & 0 \\ 0 & \frac{s_1+s_2}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{23}$$

Now, let us introduce two support matrices according to the equation (24):

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{W}^T = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{24}$$

If the first camera location is denoted by $\mathbf{P}$, with rotation and translation $\mathbf{P} = [\mathbf{I}|0]$ (zero rotation expressed by the identity matrix $\mathbf{I}$, zero translation), and the second camera matrix by $\mathbf{P}$, four solutions for rotation and translation exist:

$$\begin{aligned} \mathbf{P}' &= [\mathbf{UWV}^T|\mathbf{u_3}] \\ \mathbf{P}' &= [\mathbf{UWV}^T|-\mathbf{u_3}] \\ \mathbf{P}' &= [\mathbf{UW}^T\mathbf{V}^T|\mathbf{u_3}] \\ \mathbf{P}' &= [\mathbf{UW}^T\mathbf{V}^T|-\mathbf{u_3}] \end{aligned} \tag{25}$$

The term $\mathbf{u_3}$ stands for the 3rd column of $\mathbf{U}$. Out of four solutions given by (Eq. 25) only one is physically correct – the one for which the world points lie in front of both cameras (Figure 2).

3. Results and conclusions

The experiments were performed using the data gathered during the Rawseeds Project [10]. The dataset contains

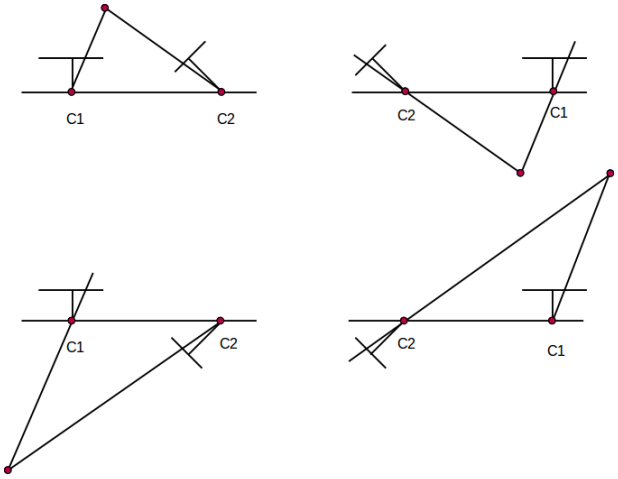

Fig. 2. Four possible camera configurations, the upper-left is physically correct.

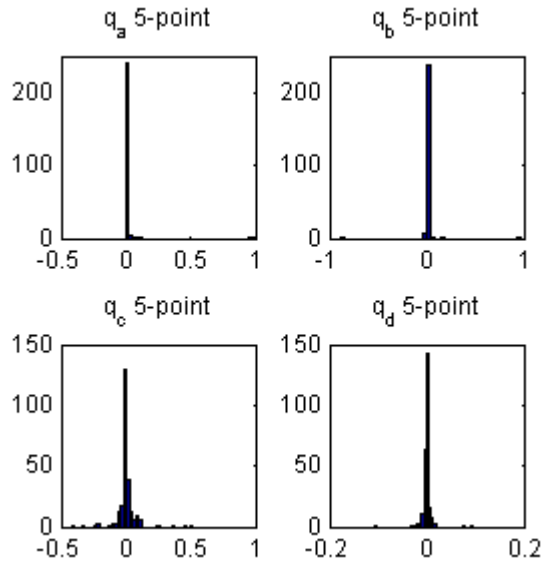

Fig. 3. The error histograms of the orientation change measurements using the 5-point algorithm.

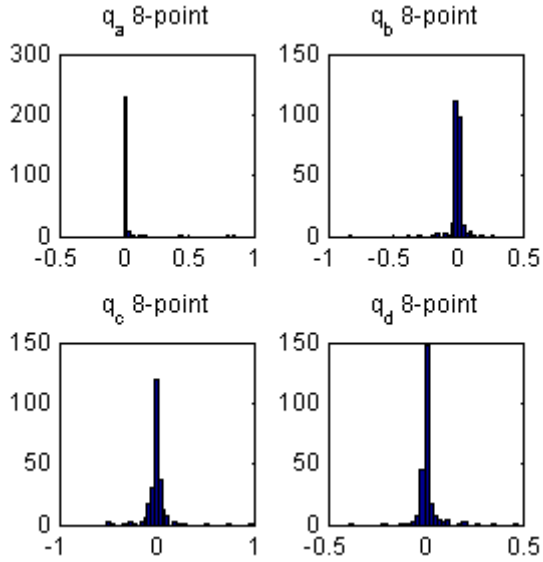

Fig. 4. The error histograms of the orientation change measurements using the 8-point algorithm.

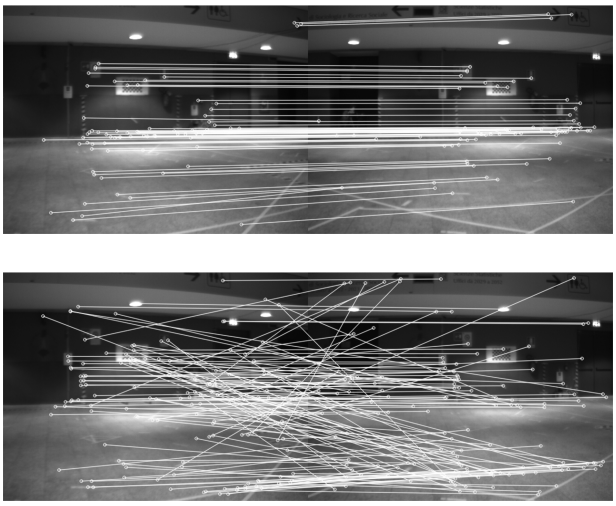

Fig. 5. Results of the feature matching and orientation estimation: inliers (top) and outliers (bottom).

the visual sequences recorded by the onboard cameras of the mobile robot along with the ground truth trajectory of the robot.

In the first stage of the experiments the precision of orientation change algorithm was assessed. A set of 250 image pairs with known relative orientation of the camera was used in the experiment. The SURF algorithm was used to detect and describe point features on each image. The point features on each image pair were matched using the brute force matching and the matches were used to estimate the camera ego motion using the 5-point [9] and the 8-point algorithm [12]. The quaternions representing the orientation change were compared with the quaternions obtained from the ground truth data. The Figure 5 presents the results of the feature matching and orientation estimation on two images from the video sequence.

The error histograms of the 5-point algorithm are presented on the Figure 3 and the error histograms of the 8-point algorithm are presented on the Figure 4. The experiment verified the assumed Gaussian character of the orientation measurement error. Moreover, the standard deviation of the measurements obtained with the 5-point algorithm was significantly smaller thus only this algorithm was used to augment the visual SLAM system.

In the second stage of the experiments the performance of the SLAM system was evaluated. As no direct stereo-vision algorithms were used, the scale of the estimated trajectory differs from the real data and was rescaled using the Procrustes analysis in order to estimate the system's precision. A sequence consisting of 400 images was used and the performance of the original 'agile camera' model and the proposed model was compared. In order to minimize the correlation between the point feature observations and the orientation change visual sequence from two onboard cameras were used: the first only for the point features observations and the second only for the orientation change estimation. The orientation change was measured every

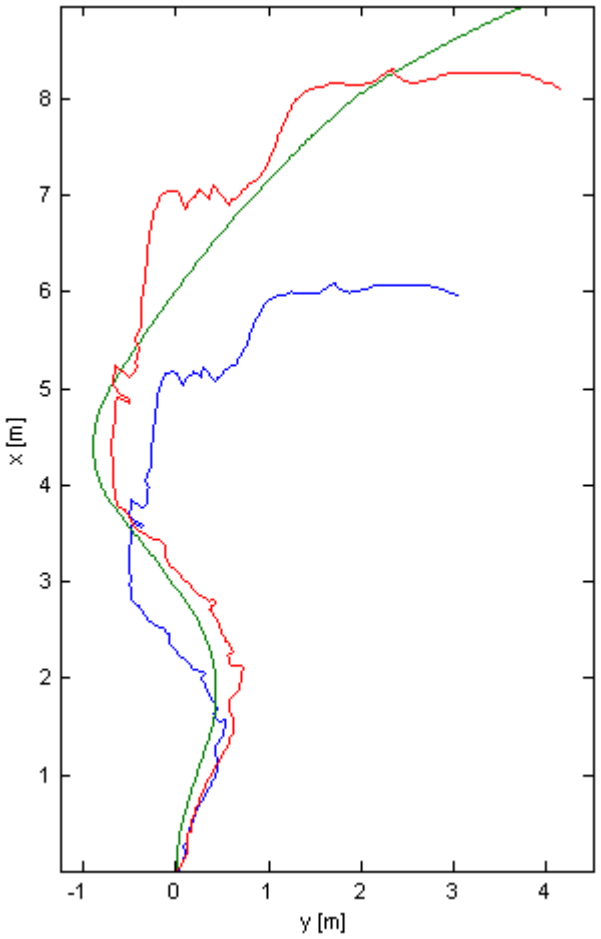

Fig. 6. The trajectory obtained using the agile camera model: GT trajectory - green, estimated trajectory - blue, estimated trajectory scaled using the Procrustes analysis - red.

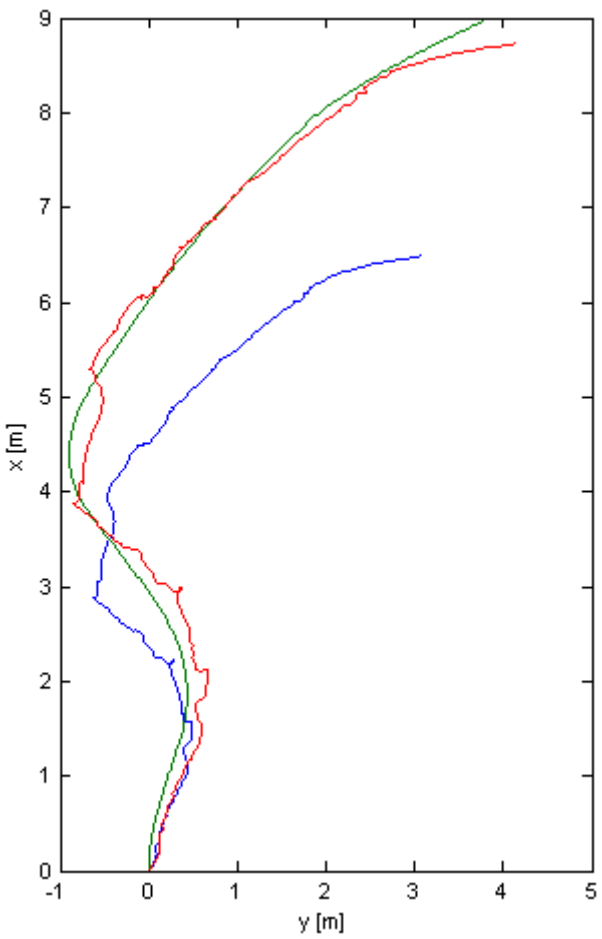

Fig. 7. The trajectory obtained using the measurable camera model: GT trajectory - green, estimated trajectory - blue, estimated trajectory scaled using the Procrustes analysis - red.

4 iterations of the SLAM system.

The Figure 6 presents the estimate of the robot's trajectory obtained with the classic 'agile camera' model. The average estimation error equaled $0.34\,\mathrm{m}$ and the maximal error equaled $0.93\,\mathrm{m}$. The Figure 7 presents the trajectory estimated using the proposed model. The average error was $0.22\,\mathrm{m}$ and the maximal error was $0.57\,\mathrm{m}$.

This paper presents an attempt to merge two paradigms of the visual navigation: the visual odometry and the visual SLAM. The monocular SLAM system was augmented with visual estimation of the robot orientation change. The experiments showed that the proposed modifications allowed to reduce the average tracking error by 35% and the maximal tracking error by 38%. It the future it is planned to test the performance of other orientation change estimation techniques.

AUTHORS

Adam Schmidt* – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Adam.Schmidt@put.poznan.pl.

Marek Kraft – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Marek.Kraft@put.poznan.pl.

Michał Fularz – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Michal.Fularz@put.poznan.pl.

Zuzanna Domagała – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Zuzanna.Domagala@cie.put.poznan.pl.

*Corresponding author

Acknowledgements

Adam Schmidt, Marek Kraft and Michał Fularz are scholarship holders within the project "Scholarship support for PH.D. students specializing in majors strategic for Wielkopolska's development", Sub-measure 8.2.2 Human Capital Operational Programme, co-financed by European Union under the European Social Fund.

References

[1] A.J. Davison, D.W. Murray, "Simultaneous Localization and Map-Building Using Active Vision", *IEEE Trans. PAMI*, vol. 24(7), 2002, pp. 865–880.

[2] R. Sim, P. Elinas, M. Griffin, "Vision-Based SLAM Using the Rao-Blackwellised Particle Filter". In: *Proc. IJCAI Workshop Reasoning with Uncertainty in Robotics*, 2005.

[3] J. Sturm, A. Visser, "An appearance-based visual compass for mobile robots", *Robotics and Autonomous Systems*, vol. 57(5), 2009, pp. 536–545.

[4] A.J. Davison, I. Reid, N. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM", *IEEE Trans. PAMI*, vol. 29(6), 2007, pp. 1052–1067.

[5] A. Schmidt, A. Kasiński, "The Visual SLAM System for a Hexapod Robot", *Lecture Notes in Computer Science*, vol. 6375, 2010, pp. 260–267.

[6] D. Scaramuzza, F. Fraundorfer, "Visual Odometry: Part I - The First 30 Years and Fundamentals", *IEEE Robotics and Automation Magazine*, vol. 18(4), 2011, pp. 80–92.

[7] F. Fraundorfer, D. Scaramuzza, "Visual Odometry: Part II - Matching, Robustness and Applications", *IEEE Robotics and Automation Magazine*, vol. 19(2), 2012, pp. 78–90.

[8] M. Fularz, M. Kraft, A. Schmidt, A. Kasiński, "FPGA Implementation of the Robust Essential Matrix Estimation with RANSAC and the 8-Point and the 5-Point Method", *Lecture Notes in Computer Science*, vol. 7174, 2012, pp. 60-71.

[9] H. Li, R. Hartley, "Five-Point Motion Estimation Made Easy", in *Proc. of the 18th International Conference on Pattern Recognition*, vol. 1, 2006, pp. 630–633.

[10] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D.G. Sorrenti, P. Taddei, "RAWSEEDS ground truth collection systems for indoor self-localization and mapping", *Autonomous Robots Journal*, vol. 27(4), 2009, pp. 353–371.

[11] J. Civera, A.J. Davison, J.M.M. Montiel, "Inverse Depth Parametrization for Monocular SLAM", *IEEE Transactions on Robotics*, vol. 24(5), 2008, pp. 932–945.

[12] R.I. Hartley, A. Zisserman, "Multiple View Geometry in Computer Vision", 2nd edition, 2004, Cambridge University Press.

[13] H. Stewénius, C. Engels, D. Nistér, "Recent developments on direct relative orientation", *ISPRS J. of Photogrammetry and Remote Sensing*, vol. 60(4), 2006, pp. 284–294.

[14] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Comm. of the ACM*, vol. 24(6), 1981, pp. 381–395.