SOLUTION SELECTION FOR FASTER ESSENTIAL MATRIX BASED STEREO VISUAL ODOMETRY

Huu-Hung Nguyen[1,*], *Anh-Duc Vu*[2], *Quang-Thi Nguyen*[1], *Cong-Manh Tran*[1]

Abstract

For autonomous navigation systems used in robots and self-driving vehicles, the usage of sequence images captured from single or multiple cameras for localization has been widely adopted due to both the low cost and high accuracy. The approach using essential matrix estimation for calculating the transformation is so popular that the essential matrix is estimated from five points which is the minimum number of points for pose estimation. However, each five pairs of points provides up to ten solutions for the essential matrix. In this paper, we propose an approach to improve the computation speed of the essential matrix by selecting the number of solutions based on a comparison with the previous essential vector of the two consecutive frames. The proposed method evaluated on the KITTI dataset shows at least 15% reduction in computation speed compared to the conventional method.

Index terms

Stereo visual odometry, essential matrix, solution selection, robotics.

1. Introduction

Robot navigation is the process of controlling a robot to move to a destination through important stages such as receiving environmental information, processing information, and path planning. It is applied in autonomous navigation systems such as planetary rovers, underwater vehicles, and self-driving vehicles including autonomous cars or Unmanned Aerial Vehicles (UAVs). These techniques are particularly used in dangerous environments where human safety is at high risk, such as mining tunnels or underwater environments. One of the fundamental challenges in an autonomous system is accuracy localization. Other localization strategies have been proposed, such as using Inertial Measurement Units (IMUs), GPS, Laser odometry, and more recently, Visual Odometry (VO) and Simultaneous Localization and Mapping (SLAM) methods [1]. Compared to other methods, VO or VSLAM is a low-cost technique that can provide accurate trajectory estimation.

[1] Institute of System Integration, Le Quy Don Technical University
[2] Institute of Information and Communication Technology, Le Quy Don Technical University
∗ Corresponding author, email: hungnh.isi@lqdtu.edu.vn DOI: 10.56651/lqdtu.jst.v12.n02.750.ict

Visual Odometry [2] is the process core of estimating the motion of a vehicle's position (e.g., a vehicle, a human, or a robot) incrementally by analyzing the changes in the images captured by the single camera or multiple cameras mounted on the vehicle. Two main approaches for geometric VO are indirect-based (feature-based) and direct-based methods. The direct methods do pose estimation from explicit features, for example, point correspondences, and line matches. The indirect methods solve an energy minimization of the image color and feature warp error to determine both camera motion and map parameters. For instance, the Oriented FAST and rotated BRIEF-SLAM2 (ORB-SLAM2) [3] is one of the full SLAM systems using the feature-based method with sparse ORB key-point distribution, while Direct Sparse Odometry (DSO) [4] is known as a full direct SLAM method. Besides that, VISO2 [5] is the popular VO framework where rotation and translation are simultaneously obtained by re-projection minimization is known as the PnP method. Similarly, Fanfani also applied the PnP approach with key-frame and feature selection [6].

The performance of visual odometry is not dependent only pose estimation method, it also depends on the feature selection method. To obtain higher accuracy or lower computation time, the feature selection process has been added to the main flow chart of VO. In [7], the authors divided detected features into 50×50 buckets. A limited number of features are selected in each bucket with the high age or strong. The selected features were based on their strength and age. The oldest features are selected first. The mutual information value of a feature is considered in [8] which provides the degree of statistical dependence to remove the redundant features with high correlation. The orthogonality index of a set of five points has been evaluated for selecting a set of correspondence for the essential matrix [9]. Feature selection has also been applied based on their strength [10], location [11] and reliability [12]. In this paper, we propose a solution selection method for essential matrix estimation that helps reduce the computation time of summation of epipolar error. The conventional method is usually used to evaluate all solutions by calculating the summation of epipolar errors to select the minimum one. That is not necessary while the robot or car moves at high speed.

The rest of this paper is organized as follows. Section 2 summarizes a traditional method for robot localization based on essential matrix calculation. Section 3 introduces our proposed method for speeding up essential matrix computing in VO problem. Section 4 presents our results and evaluates our methods by comparing them with other approaches on the KITTI dataset.

2. Pose estimation

This section provides an overview of the essential matrix-based VO. The transformation components for determining the vehicle's position relative to the initial position incorporates rotation matrix and translation vector of each pair of consecutive frames. The camera motion can be estimated by two processes: 1) rotation matrix by essential matrix and 2) translation vector by left and right constraint [13].

2.1. Rotation estimation

The values of the rotation matrix and the normalized translation vector can be computed from the estimated essential matrix (**E**) [2]. The essential matrix can be expressed by rotation matrix and translation vector through the following formula (1):

$$\mathbf{E} = \mathbf{T}^{\times}\mathbf{R} \tag{1}$$

where **R** and $\mathbf{T}^{\times}$ are two matrices 3×3 that are rotation and skew-symmetric translation matrices. Moreover, the essential matrix **E** must satisfy two following constraints:

$$det(\mathbf{E}) = 0. \tag{2}$$

$$2\mathbf{E}\mathbf{E}^{\mathbf{T}}\mathbf{E} - tr(\mathbf{E}\mathbf{E}^{T})\mathbf{E} = 0 \tag{3}$$

The essential matrix from the current image and the previous image represents the relationship between pairs of feature points. For instant, (p, q) is a pair of corresponding 2D feature points from the two frames, they satisfy the constraint expressed by the following equation:

$$p^{T}\mathbf{E}q = 0 \tag{4}$$

To obtain the essential matrix using the Nister five-point algorithm [2], the constraints given in equations (2) and (3) are combined. As we know, the essential matrix is computed from a set of five corresponding feature point pairs. These five pairs are used in the conditions of equations (2), (3), and (4). After transforming these equations into a tenth-order polynomial equation, there can be a maximum of 10 solutions. Once the essential matrix is estimated, the rotation matrix **R** and translation vector **t** can be recovered.

2.2. Translation estimation

Note that, the translation vector calculated from the essential matrix is a normalized vector which indicates the direction of translation. The missing scale of translation can be easily calculated in several ways: 1) solving the equation $\hat{P} = \mathbf{R}\hat{Q} + \mathbf{t}$ for a pair of 3D correspondence $(\hat{P}, \hat{Q})$ with RANSAC scheme; 2) estimating the translation by minimizing re-projection errors. The estimation of translation was affected more or less when using 3D features to calculate the value of translation estimation due to the uncertainty of 3D features. To reduce this effect, [14] proposed a method to avoid using 3D features for estimating translation. It can be summarized as follow: For each 3D feature $\hat{Q}$ in the current frame, its projection on the image plane can be described as

$$\hat{P} = \alpha\mathbf{K}(\widetilde{\mathbf{R}}\hat{\mathbf{Q}} + \widetilde{\mathbf{t}}) \tag{5}$$

where pair of ($\widetilde{\mathbf{R}}$, $\widetilde{\mathbf{t}}$)

- $\widetilde{\mathbf{R}} = \mathbf{I}$ and $\widetilde{\mathbf{t}} = \mathbf{0}$ are projection on current left frame
- $\widetilde{\mathbf{R}} = \mathbf{I}$ and $\widetilde{\mathbf{t}} = -\mathbf{b}$ are projection on current right frame
- $\widetilde{\mathbf{R}} = \mathbf{R}$ and $\widetilde{\mathbf{t}} = \mathbf{t}$ are projection on previous left frame

- $\widetilde{\mathbf{R}} = \mathbf{R}$ and $\widetilde{\mathbf{t}} = \mathbf{t}\text{-}\mathbf{b}$ are projection on previous right frame
- K is intrinsic matrix

Four projection equations are denoted in compact way as following linear equations:

$$\mathbf{A}_{8\times 6}\begin{pmatrix}\hat{\mathbf{Q}}\\ \mathbf{t}\end{pmatrix}_{6\times 1} = \mathbf{B}_{8\times 1} \tag{6}$$

This equation (6) is a linear equation including eight equations with six unknown variables that can be solved by using the Pseudo Inverse method. That mean, using only a pair of 3D feature can estimate the translation. To verify the accuracy in the real situation, this algorithm is wrapped into the RANSAC scheme with maximum 100 samples of the closest 3D features.

2.3. Transformation integration

Since the current stereo frame, f_k, is captured, the pose of the last two frames f_{k-1} and f_{k-2} is already estimated (${}^{k-2}_{k-1}\mathbf{T}$). As we know, the transformation from the current frame to that two frames, (${}^{k-2}_{k}\mathbf{T}$, ${}^{k-1}_{k}\mathbf{T}$) satisfies a closed-loop constraint described as follows,

$${}^{k-2}_{k}\mathbf{T} = {}^{k-2}_{k-1}\mathbf{T}\,{}^{k-1}_{k}\mathbf{T} \tag{7}$$

The loop constraint can be rewritten

$${}^{k-1}_{k}\mathbf{T} = {}^{k-2}_{k-1}\mathbf{T}^{inv}\,{}^{k-2}_{k}\mathbf{T} \tag{8}$$

That means the transformation between the current and previous frames can be observed in two ways: 1) the first observation, ${}^{k-1}_{k}\bar{\mathbf{T}}$, is directly estimated from correspondences between frame f_k and f_{k-1} and 2) the second observation, ${}^{k-1}_{k}\widetilde{\mathbf{T}}$, is indirectly estimated from the relative pose of the two before last frames through equation (9).

$${}^{k-1}_{k}\widetilde{\mathbf{T}} = {}^{k-2}_{k-1}\hat{\mathbf{T}}^{-1}\,{}^{k-2}_{k}\bar{\mathbf{T}} \tag{9}$$

where ${}^{k-2}_{k-1}\hat{\mathbf{T}}$ is denoted as the refined transformation between frame f_{k-2} and f_{k-1}. Finally, the refined transformation is done by combining the direct and indirect observations by the following equation,

$${}^{k-1}_{k}\hat{\mathbf{T}} = w\,{}^{k-1}_{k}\bar{\mathbf{T}} + (1-w)\,{}^{k-1}_{k}\widetilde{\mathbf{T}} \tag{10}$$

where w is defined as the weight value. However, each transformation contains two components: rotation matrix and translation vector. The rotation component is more important, it can affect the translation estimation later. For that reason, we do refine the translation component following the rotation one. It is known that quaternion is the best way to a representative for orientation. The rotation matrix is converted to quaternion is refined as

$${}^{k-1}_{k}\hat{\mathbf{q}} = w\,{}^{k-1}_{k}\bar{\mathbf{q}} + (1-w)\,{}^{k-1}_{k}\widetilde{\mathbf{q}} \tag{11}$$

After that, the translation is refined as

$$ {}^{k-1}_{k}\hat{\mathbf{t}} = w\, {}^{k-1}_{k}\bar{\mathbf{t}} + (1-w)\, {}^{k-1}_{k}\widetilde{\mathbf{t}} \tag{12} $$

where ${}^{k-1}_{k}\bar{\mathbf{t}}$ and ${}^{k-1}_{n}\widetilde{\mathbf{t}}$ are two representative ways for translation from frame k to frame $k-1$; ${}^{k-1}_{k}\hat{\mathbf{t}}$ is the refined translation. With transformation refinement process mathematically denoted in (10), the estimated camera trajectory is more correct.

3. Solution selection for faster essential matrix estimation

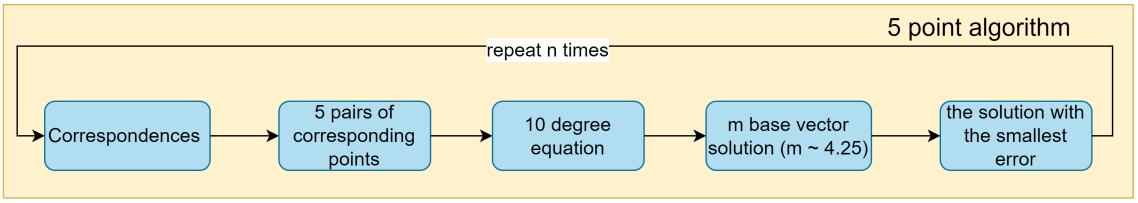
Recently, VO based on the essential matrix has shown superior performance on the KITTI dataset [15] compared to the PnP method. For example, the SOFT2 [16] method proposes a careful geometric object selection map based on the features of the geometric object after estimating the rotation using the essential matrix and then refines the estimated translation by minimizing the re-projection error. Another method, MESVO_FP [13], proposes integrating multiple frames for localization based on the essential matrix by investigating the transformation and feature integration among the last three frames in the user interface. A loop closure constraint is used to refine the relative pose between the previous and current frames. Additionally, the feature positions of the current frame are refined by geometric constraints from previous frames.

It is known that autonomous vehicles often move at high speed, resulting in little significant difference in rotation angle and translation direction, hence the essential matrix or essential vector hardly change. This allows for proposing methods to reduce computation time. This paper presents an improvement in the computation speed of the essential matrix for VO by selecting k of m solutions using stored and compared essential vectors of the two adjacent frames represented as 9×1 vectors. These k solutions have the smallest angular error compared to the essential vector of the two previous frames.

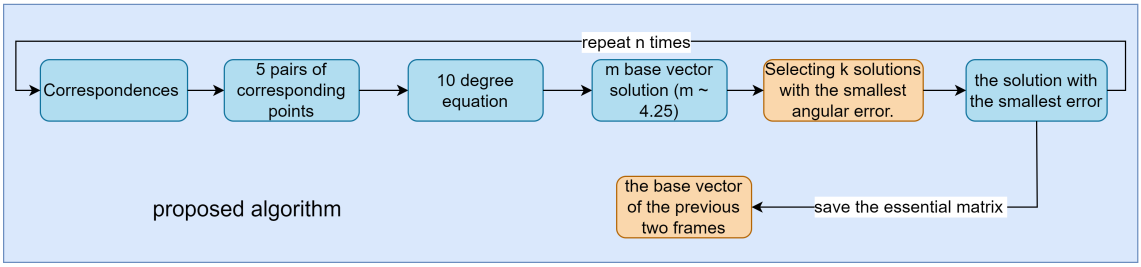
To confirm the proposed method, we made a statistic related to the dot product of two consecutive essential vectors shown in table 1. We measured the dot product of 10000 pairs of consecutive frame. The number of dot product bigger than 0.9999 is 9491 and the number of product bigger than 0.99 is 9957. That mean all most pairs of consecutive frames have the similar essential vector or similar essential matrix, which means we can use the previous essential vector for eliminating the unnecessary solutions where the angle between them is big. We also verified that at low dot product, the camera does not move.

Table 1. Dot product statistic of two consecutive frames

Total frames	Dot product > 0.9999	Dot product > 0.99
10000	9491	9957



(a) The conventional five-point algorithm [2] with the average number of solutions for error evaluation (m = 4.25) m is up to 10.



(b) The proposed solution selection (k = 1, 2, 3). Save 15% computational time with k = 3

Fig. 1. The proposed solution selection to improve the computational speed.

Figure 1(b) illustrates the proposed approach based on the well-known five-point algorithm [2] described in figure 1(a). Starting with 5 pairs of corresponding points, they are used to satisfy the conditions in (2), (3), and (4) to transform into a 10th-degree single-variable equation. Solving this equation yields a maximum of 10 solutions, and the solution with the smallest error based on epipolar constraints is selected. Our main contribution is highlighted in the orange block, where from the N obtained solutions, we rely on the vectors of the two previous adjacent frames to choose 2 solutions with the smallest error compared to the essential vector. Finally, the solution with the smallest error from the 2 chosen solutions is selected. With these improvements, the proposed method achieves an average translation error of 0.86% and an average rotation error of 0.306 degrees/100 m, slightly higher than the initial method (0.85% and 0.341 degrees/100 m), but yielding better results in some frames and shorter computation time compared to the previous method by 15% with k = 3.

According to the Nister five-point algorithm [2], on average, usually, there are average 5 - 6 solutions for each of set of five points. The solution with minimum summation of square epipolar error is selected, that is the final solution that represents to a set of five points. However, for self-driving vehicles, they often move at high speeds, so the rotation and translation change between frames are not significant. That means the essential matrices of two consecutive frames do not change much, they are similar. In another way, the angle the representatives of essential matrices in 9×1 vectors is small or dot product is almost 1. Therefore, we can take advantage of this condition to eliminate some unnecessary solutions of the essential matrix with high summation of

square epipolar error in order to reduce computation time. The 3×3 essential matrix is transformed into a 9×1 vector representation, essential vector. This vector representation of the essential matrix between the current frame and the previous frame E_{prev} is stored for removing solutions between the current frame and the previous frame. The method of selecting k out of m solutions of the essential matrix with a simple idea is described in detail in figure 2. The vector representation of the essential matrix between the two previous frames E_{prev} is stored. Another vectors are solutions of essential matrix in vector 9×1. They all are in 9D space. Usually, the final solution of essential matrix is obtained by evaluating the summation of epipolar error [17]. When finding the top m solutions of the essential matrix E_i between the current frame and the previous frame, the error in the angle difference between the two vectors is evaluated, which is represented by the dot product of two 9×1 vectors.

$$dot_i = E_{prev} E_i \tag{13}$$

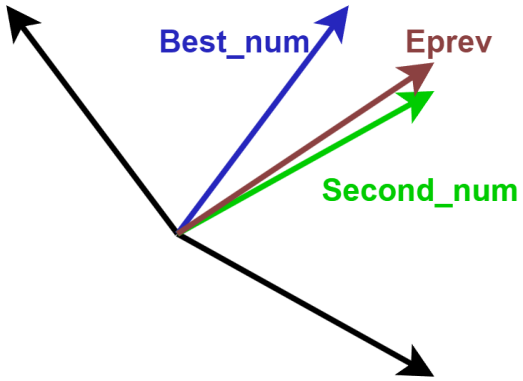


Fig. 2. The method of selecting k out of m solutions. Top k closest vectors with maximum dot product are selected for evaluation error score.

Only k of m vectors (k top smallest angular error compared to E_{prev}) are kept for the error evaluation stage, and $m - k$ vectors are discarded, often due to having too large angle with E_{prev}. The result of the error evaluation selects the vector with the smallest error for a set of 5-point correspondences. This process is repeated n times with N sets of five point correspondences to select the final basis vector. The computation of the rotation matrix and translation vector is done in [2], and the basis vector of the current and previous frame is saved in E_{prev} for comparison with next frames.

4. Experimental results

We use the KITTI dataset [15] to evaluate the performance of our proposed method. The KITTI dataset is a well-known dataset in the self-driving car research

community, consisting of 22 sequences in total divided into two groups: 1) Training dataset (00-10) and 2) Testing dataset (11-21). The training dataset provides the ground-truth trajectories for each frame in each sequence. The testing dataset does not include the ground-truth trajectories, and the evaluation is performed by loading the ground-truth information from the publicly available website, and the errors are automatically calculated. Note that, the dataset is collected under various environmental conditions such as speed, lighting, darkness, moving objects to accurately evaluate the algorithm's performance. The dataset also provides a tool to automatically evaluate the performance of the algorithm by measuring the relative errors ($RMSE$) including the rotation error (RE) and translation error (TE). It determines the average errors of all sub-sequences with lengths of 100, 200, ..., 800 meters. To obtain objective evaluation, we directly compare our improvement with the different approach, MESVO_PF [13]. Beside that, we compare accuracy of our approach with another conventional method such as VISO2 [5].

Table 2. Evaluation of time and number of solutions

	MESVO_PF [13]	**Five-point algorithm [2]**	**Proposed method**		
Number of solutions	4.2	5-6	k=1	k=2	k=3
Average time (ms)	59.9	60	**42.9**	45.6	50.9

Table 3. Evaluating accuracy on the KITTI dataset for the method of improving computational speed

Sec Num	**MESVO_PF k= all (5-6)**		**Proposed method**					
			k = 1		**k = 2**		**k = 3**	
	TE (%)	RE ($\frac{deg}{100m}$)	TE (%)	RE ($\frac{deg}{100m}$)	TE (%)	RE ($\frac{deg}{100m}$))	TE (%)	RE ($\frac{deg}{100m}$)
Avg	0.85	0.341	**0.81**	**0.290**	0.86	0.306	0.85	0.306
00	0.85	0.355	0.84	0.350	**0.82**	**0.338**	0.82	0.338
01	-	-	-	-	-	-	-	-
02	**0.76**	0.287	0.79	0.281	0.81	**0.269**	0.81	0.269
03	0.75	0.281	0.88	**0.229**	**0.71**	0.275	0.71	0.275
04	**0.57**	0.161	0.66	**0.138**	0.70	0.161	0.70	0.161
05	0.72	0.315	0.60	0.292	**0.59**	**0.258**	0.59	0.258
06	**0.87**	**0.304**	0.95	**0.332**	0.97	0.338	0.97	0.338
07	1.08	0.785	**0.56**	0.430	0.66	**0.407**	0.66	0.407
08	1.07	0.344	**0.86**	**0.264**	0.99	0.298	0.99	0.298
09	0.99	0.269	**0.84**	**0.212**	1.03	0.269	1.03	0.269
10	**0.80**	**0.304**	1.09	0.373	1.28	0.447	1.28	0.447

We evaluated the computation time of the essential matrix for two methods: 1) the 5-point method used in MESVO_FP [13], 2) five-point algorithm [2], and 3) the proposed method of selecting k ($k = 1, 2, 3$) vector solutions. Additionally, we measured the average number of solutions for three methods. The results are summarized in table 2. Note that, the two conventional methods use all solutions for essential matrix estimation. The average computation time of the proposed method for $k = 1, 2, 3$ is 42.9 ms, 45.6 ms, and 50.9 ms, compared to 59.9 ms for the 5-point method used in MESVO_FP,

resulting in reductions of 28.4%, 24.9%, and 15.0% for $k = 1, 2, 3$. Definitely, using smaller number of solution makes the program run faster because checking sum of error from all points. The relative errors (RMSE) of different k shown in table 3 indicates that the errors do not change much between k selection or all selection. The changes are mostly affected by RANSAC scheme. That means, we can select only one solution ($k = 1$) for the essential vector. In that case, the computation time for the essential matrix can speed up 28.4%.

Our algorithm is compared with other methods such as VISO2 [5], MESVO_PF [13] and ESVO [14] to evaluate its performance. The RE is the average rotation matrix error (degree/100 m) and the TE is the translation error (%) summarized in table 4. It shows the $RMSE$ of all 11 sequences as well as their average details for three approaches. Look at table 4, we can realize that the proposed method achieves lower errors for rotation in almost all sequences. This result indicates that the proposed method enhances the accuracy of rotation estimation. Our proposed algorithm achieved lower average rotation errors compared to its previous versions in multiple sequences. For example, the average rotation errors of MESVO_FP and our method were 0.324 and 0.306, respectively, 0.527 and 0.407 in sequence 00, and 0.286 and 0.269 in sequence 07. Although our method is based on a similar essential matrix approach as MESVO_FP, our method reduced the rotation error by about 1% compared to MESVO_FP, and ESVO reduced the rotation error by about 7% compared to ESVO. However, the computation for estimating essential matrix of the proposed method is less than that of MESVO and ESVO at least 15%. It means that the accuracy does not change much while the computation time is significantly reduced. Compare to the accuracy of the VISO2, the average translation error of the proposed method reduces from 2.43% to 0.85%.

Table 4. Evaluating accuracy on the KITTI dataset for method of selecting 2 out of m solutions

Sec Num	**VISO2**		**ESVO**		**MESVO_PF**		**Ours method k=2**	
	TE (%)	RE ($\frac{deg}{100m}$)	TE (%)	RE ($\frac{deg}{100m}$)	TE (%)	RE ($\frac{deg}{100m}$)	TE (%)	RE ($\frac{deg}{100m}$)
Avg	2.43	1.106	1.04	0.483	0.85	0.324	**0.85**	**0.306**
00	2.46	1.181	1.04	0.487	0.82	0.355	**0.82**	**0.338**
01	4.42	1.015	-	-	-	-	-	-
02	2.19	0.808	0.85	0.327	**0.76**	0.286	0.81	**0.269**
03	2.54	1.198	0.81	0.401	0.92	0.315	**0.71**	**0.275**
04	1.02	0.866	0.62	0.427	**0.55**	0.195	0.70	**0.161**
05	2.07	1.124	0.71	0.378	0.67	0.304	**0.59**	**0.258**
06	1.31	0.917	1.22	0.522	0.97	0.338	**0.96**	**0.338**
07	2.30	1.771	1.31	0.980	0.88	0.527	**0.66**	**0.407**
08	2.74	1.336	1.40	0.447	1.18	0.361	**0.99**	**0.298**
09	2.76	1.152	1.00	0.321	**0.93**	**0.252**	1.03	0.270
10	1.63	1.118	1.21	0.539	**0.79**	**0.327**	1.29	0.447

To demonstrate the accuracy, we plot the trajectory of the autonomous vehicle in sequence 00 and sequence 07 of the KITTI dataset. Figure 3 visualizes the sequence 00 and figure 4 visualizes the sequence 07. The red line represents the ground-truth

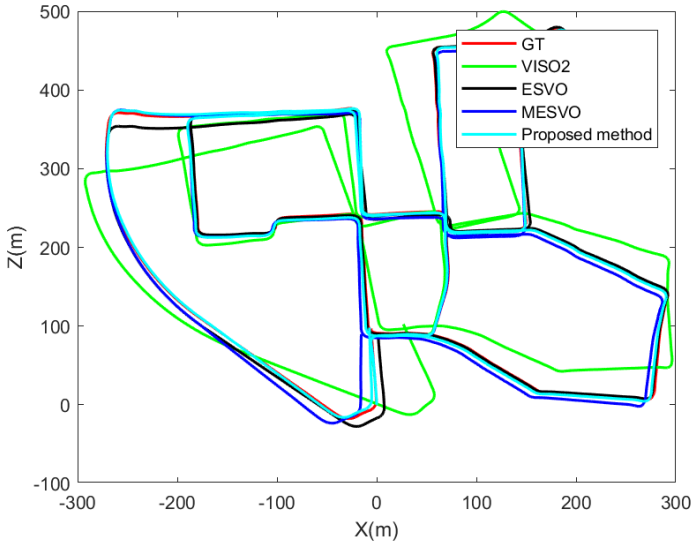


Fig. 3. Trajectory of sequence 00th for four methods (VISO2. ESVO, MESVO_FP, ours) compare to the ground-truth.

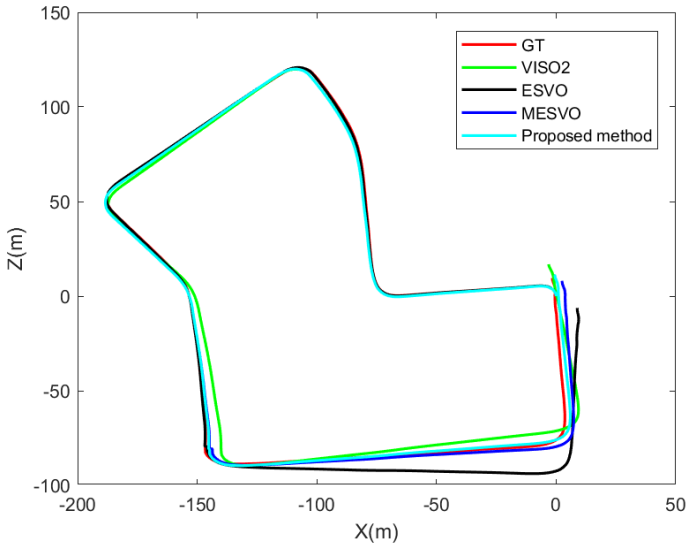


Fig. 4. Trajectory of sequence 07th for four methods (VISO2. ESVO, MESVO_FP, ours) compare to the ground-truth.

trajectory built from GPS and IMU, which is considered the accurate path of the vehicle. The green, black, and blue lines are the results of the VISO2, ESVO and MESVO_FP methods, respectively. The cyan line is the result of our proposed method, which is equivalent to MESVO_FP since it is built upon this method. The trajectory of the proposed method is clearly closer to the ground-truth than VISO2.

5. Conclusions

We have investigated the improvement of speed in visual odometry by employing a solution selection method from maximum 10 essential matrix solutions, and finally selecting the solution with the smallest error. The computation speed has also been significantly reduced at least 15% while the accuracy is almost the same. We also have demonstrated that using this method results in a relative improvement in accuracy compared to the previous method. In the future, we will continue to explore other avenues for further performance improvement in visual odometry for autonomous vehicles and unmanned aerial vehicles.

References

[1] H. P. Moravec, "The stanford cart and the cmu rover," *Proceedings of the IEEE*, vol. 71, no. 7, pp. 872–884, 1983. doi: 10.1109/PROC.1983.12684

[2] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, vol. 1. IEEE, 2004. doi: 10.1109/CVPR.2004.1315094 pp. I–I.

[3] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017. doi: 10.1109/TRO.2017.2705103

[4] R. Wang, M. Schworer, and D. Cremers, "Stereo dso: Large-scale direct sparse visual odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017. doi: 10.48550/arXiv.1708.07878 pp. 3903–3911.

[5] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010. doi: 10.1109/IVS.2010.5548123 pp. 486–492.

[6] M. Fanfani, F. Bellavia, and C. Colombo, "Accurate keyframe selection and keypoint tracking for robust visual odometry," *Machine Vision and Applications*, vol. 27, no. 6, pp. 833–844, 2016.

[7] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," in *2015 European Conference on Mobile Robots (ECMR)*, 2015. doi: 10.1109/ECMR.2015.7324219 pp. 1–6.

[8] R. Kottath, S. Poddar, R. Sardana, A. P. Bhondekar, and V. Karar, "Mutual information based feature selection for stereo visual odometry," *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 1559–1568, 2020. doi: 10.1007/s10846-020-01206-z

[9] H. H. Nguyen and S. Lee, "Orthogonality index based optimal feature selection for visual odometry," *IEEE Access*, vol. 7, pp. 62 284–62 299, 2019. doi: 10.1109/ACCESS.2019.2916190

[10] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," in *2015 European Conference on Mobile Robots (ECMR)*. IEEE, 2015. doi: 10.1109/ECMR.2015.7324219 pp. 1–6.

[11] L. De-Maeztu, U. Elordi, M. Nieto, J. Barandiaran, and O. Otaegui, "A temporally consistent grid-based visual odometry framework for multi-core architectures," *Journal of Real-Time Image Processing*, vol. 10, pp. 759–769, 2015. doi: 10.1007/s11554-014-0425-y

[12] W. Zhou, H. Fu, and X. An, "A classification-based visual odometry approach," in *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 2. IEEE, 2016. doi: 10.1109/IHMSC.2016.212 pp. 85–89.

[13] H.-H. Nguyen, T.-T. Nguyen, X.-P. Nguyen, C.-M. Tran, and Q.-T. Nguyen, "Multiple frame integration for essential matrix-based visual odometry," in *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. IEEE. doi: 10.1109/IMCOM53663.2022.9721757 pp. 1–6.

[14] H.-H. Nguyen, Q.-T. Nguyen, C.-M. Tran, and D.-S. Kim, "Adaptive essential matrix based stereo visual odometry with joint forward-backward translation estimation," in *Industrial Networks and Intelligent Systems: 6th EAI International Conference, INISCOM 2020, Hanoi, Vietnam, August 27–28, 2020, Proceedings 6*. Springer, 2020. doi: 10.1007/978-3-030-63083-6 pp. 127–137.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The Kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012. doi: 10.1109/CVPR.2012.6248074 pp. 3354–3361.

[16] I. Cvišić, I. Marković, and I. Petrović, "Soft2: stereo visual odometry for road vehicles based on a point-to-epipolar-line metric," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 273–288, 2022. doi: 10.1109/TRO.2022.3188121

[17] M. E. Fathy, A. S. Hussein, and M. F. Tolba, "Fundamental matrix estimation: A study of error criteria," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 383–391, 2011. doi: 10.1016/j.patrec.2010.09.019

Manuscript received 06-09-2023; Accepted 21-12-2023.

Huu-Hung Nguyen received his Ph.D. degree at Sungkyunkwan University, South Korea in computer vision, in 2020. He is currently researcher at Institute of System Integration, Le Quy Don Technical University. His research interests include computer vision, simultaneous localization and mapping (SLAM), 3D point cloud processing, deep learning and AI.E-mail: hungnh.isi@lqdtu.edu.vn

Anh-Duc Vu is a fifth-year student at Le Quy Don Technical University, Vietnam, majoring in information technology. Currently researching the fields of computer vision and autonomous vehicles.

Quang-Thi Nguyen received his Ph.D. degree at Changchun University of Science and Technology, China in Communication and Information System in 2014. His research interests include computer vision, blind deconvolution, image processing and pattern recognition.He is currently researcher at Institude of System Integration, Le Quy Don Technical University.

Cong-Manh Tran got his master degree in computer science from Le Quy Don Technical University of Vietnam, in 2007. In 2017, Manh got his PhD degree from Department of Computer Science, National Defense Academy, Japan. His current research interests include network security, intelligent computing, and data analysis. Currently, Dr. Manh works as a researcher in Le Quy Don Technical University, Hanoi, Vietnam.

LỰA CHỌN NGHIỆM CHO ƯỚC LƯỢNG NHANH HƠN VỊ TRÍ PHƯƠNG TIỆN SỬ DỤNG CAMERA ĐÔI

Nguyễn Hữu Hùng, *Vũ Anh Đức*, *Nguyễn Quang Thi*, *Trần Công Mạnh*

Tóm tắt

Đối với các hệ thống định vị tự động cho các phương tiện tự hành như robot và xe tự lái, việc sử dụng hình ảnh tuần tự được chụp từ một hoặc nhiều camera để định vị vị trí của phương tiện đã được áp dụng rộng rãi do chi phí thấp và độ chính xác cao. Cách tiếp cận sử dụng ước lượng ma trận cơ sở để tính toán ma trận chuyển trong đó ma trận cơ sở được tính toán từ năm điểm là số điểm tối thiểu dùng để ước lượng. Tuy nhiên, mỗi năm cặp điểm cung cấp tới mười nghiệm cho ma trận cơ sở, việc lựa chọn nghiệm phải đánh giá tổng sai số bình phương của toàn bộ các cặp điểm. Trong bài báo này, chúng tôi đề xuất phương pháp cải thiện tốc độ tính toán của ma trận cơ sở bằng cách chọn số nghiệm dựa trên sự so sánh với vectơ cơ sở của hai khung liên tiếp trước đó. Phương pháp đề xuất được đánh giá trên bộ dữ liệu KITTI cho thấy tốc độ tính toán giảm ít nhất 15% so với phương pháp thông thường.

Từ khóa

Ma trận cơ sở, ước lượng vị trí, lựa chọn nghiệm, lựa chọn điểm.