ENHANCE STEREO VISUAL ODOMETRY PERFORMANCE BY REMOVING UNSTABLE FEATURES

Anh Duc Vu[1], *Xuan Phuc Nguyen*[2], *Van Xiem Hoang*[3], *Quang Lam Bui*[4], *Quang Hieu Dang*[2], *Huu Hung Nguyen*[2,*]

Abstract

Visual odometry includes two important stages: 1) feature extraction and 2) pose estimation. The performance of visual odometry is dependent on the quality of features including the number of features, the percentage of the correct matching, and the location of detected features. Usually, RANSAC method has been used in pose estimation to remove outlier and select a good set of features that provide higher accuracy. However, in the case the higher wrong matches, the RANSAC seems to be failing. This article proposes the removing unstable feature method by deep learning-based object detection. The proposed method evaluated on the KITTI dataset shows a higher accuracy 6 - 8% compared to the conventional method.

Index terms

Stereo visual odometry, essential matrix, object detection, unstable feature selection.

1. Introduction

Visual Odometry (VO) [1] involves estimating a vehicle's motion by analyzing image sequences from onboard cameras. Two primary geometric VO approaches exist: indirect (feature-based) and direct methods. Indirect methods focus on minimizing image color and feature warp errors, while direct methods do not rely on distinct features. Examples include ORB-SLAM2 [2] for feature-based SLAM and Direct Sparse Odometry (DSO) [3] for direct SLAM. Feature selection is critical for VO accuracy and computational efficiency. Previous studies have used various criteria such as feature strength, age, spatial location, and mutual information values to optimize feature selection and eliminate redundant features [4].

[1] Institute of Information and Communication Technology, Le Quy Don Technical University

[2] Institute of System Integration, Le Quy Don Technical University

[3] Faculty of Electronics and Telecommunications, Vietnam National University - University of Engineering and Technology

[4] Phu Xuan University

[*] Corresponding author, email: hungnh.isi@lqdtu.edu.vn

DOI: 10.56651/lqdtu.jst.v13.n01.819.ict

The efficacy of VO is contingent not solely on the pose estimation method but also on the feature selection approach employed. In the pursuit of heightened accuracy or reduced computational overhead, the integration of feature selection into the primary workflow of VO has become imperative. Notably, in previous studies [5], detected features were partitioned into 50×50 buckets, from which a limited number of features were selectively chosen based on their age or robustness. Priority was given to older features, with criteria including strength and age influencing selection. Additionally, the incorporation of mutual information values, as discussed in Kottath et al. [6], facilitated the removal of redundant features exhibiting high correlation. Furthermore, the assessment of the orthogonality index, as outlined in Nguyen et al. (2019) [7], aided in the selection of correspondences for the essential matrix. Feature selection criteria extended to considerations of strength [8], spatial location [9], and reliability [10]. Additionally, in complex environments with moving objects, using features on these moving objects affects the accuracy of pose estimation. Eliminating these features will enhance pose estimation performance and increase localization accuracy. Traditionally, methods using RANSAC for selecting points to compute pose often cannot avoid choosing points on moving objects [11].

Deep learning object detection has transformed industries by enabling automated object identification and tracking in visual data using complex neural networks. This technology allows for real-time recognition of diverse objects such as vehicles, pedestrians, animals, and household items with high accuracy. Its applications span autonomous driving, surveillance, medical imaging, and retail inventory management, enhancing efficiency and opening new possibilities in security, healthcare, and commerce, thereby creating a more intelligent and interconnected world.

This article proposes a Deep Learning Object Detection method to eliminate mobile features, reducing errors in autonomous vehicle movement. Unlike conventional methods that use all feature points for pose estimation, the proposed approach recognizes errors caused by mobile features, particularly in scenarios with moving obstacles. This article is structured as follows: Section 2 summarizes conventional localization methods, section 3 outlines the proposed method for improving VO accuracy through feature selection, and section 4 presents findings and compares the proposed method with alternative approaches using the KITTI dataset.

2. Pose estimation based essential matrix selection

This section offers an insight into essential matrix-based VO without close-loop detection and pose graph optimization. Firstly, blobs and corners are extracted from stereo images and are matched in loop [5]. After that, the fundamental transformation components for ascertaining the vehicle's position concerning the initial position involve the rotation matrix and translation vector for each pair of successive frames. The estimation of camera motion encompasses two key processes: 1) Determination of the rotation matrix through the essential matrix and 2) Calculation of the translation vector utilizing the 3 consecutive frames constraints.

2.1. Rotation estimation and translation estimation

The rotation matrix and the normalized translation vector can be derived from the estimated essential matrix (**E**) [1]. The essential matrix can be represented in terms of the rotation matrix and translation vector using the following formula:

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

To compute the essential matrix using the Nister five-point algorithm [1] by combing the constraints defined earlier. As per the algorithm, the essential matrix is derived from a set of five corresponding feature point pairs, which satisfies the conditions specified. These conditions are then transformed into a tenth-order polynomial equation, yielding up to 10 possible solutions. Upon obtaining the essential matrix, the rotation matrix **R** and translation vector **t** are subsequently recovered by solving a linear equation [12].

For each 3D feature $\hat{Q}$ in the current frame, its projection onto the image plane can be described as:

$$\hat{P} = \alpha\mathbf{K}(\widetilde{\mathbf{R}}\hat{\mathbf{Q}} + \widetilde{\mathbf{t}}) \tag{2}$$

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

$$\mathbf{A}_{8\times6}\begin{pmatrix}\hat{\mathbf{Q}}\\ \mathbf{t}\end{pmatrix}_{6\times1} = \mathbf{B}_{8\times1} \tag{3}$$

The equation (3) represents a linear equation system consisting of eight equations with six unknown variables, which can be effectively solved using the pseudo-inverse method. This implies that a single pair of 3D features is adequate for estimating translation.

2.2. Solution selection

This method uses the five-point algorithm [1]. It starts with 5 pairs of corresponding points, forming a 10th-degree polynomial equation. From the N solutions, vectors from the two previous frames are used to identify the 2 solutions with the least error compared to the essential vector. The one with the smallest error is selected [13].

The 3×3 essential matrix is transformed into a 9×1 vector, known as the essential vector (E_{prev}), to efficiently compare frames. This vector captures the essential matrix between the current and previous frames, facilitating comparisons with future frames.

Only k of m vectors (with small error compared to E_{prev}) are kept for error evaluation, discarding $m-k$ vectors with larger angles. The vector with the smallest error is chosen for each set of 5-point correspondences. This process is repeated n times with N sets to select the final basis vector. The rotation matrix and translation vector computation are based on [1], and the basis vector is stored in E_{prev} for future comparisons.

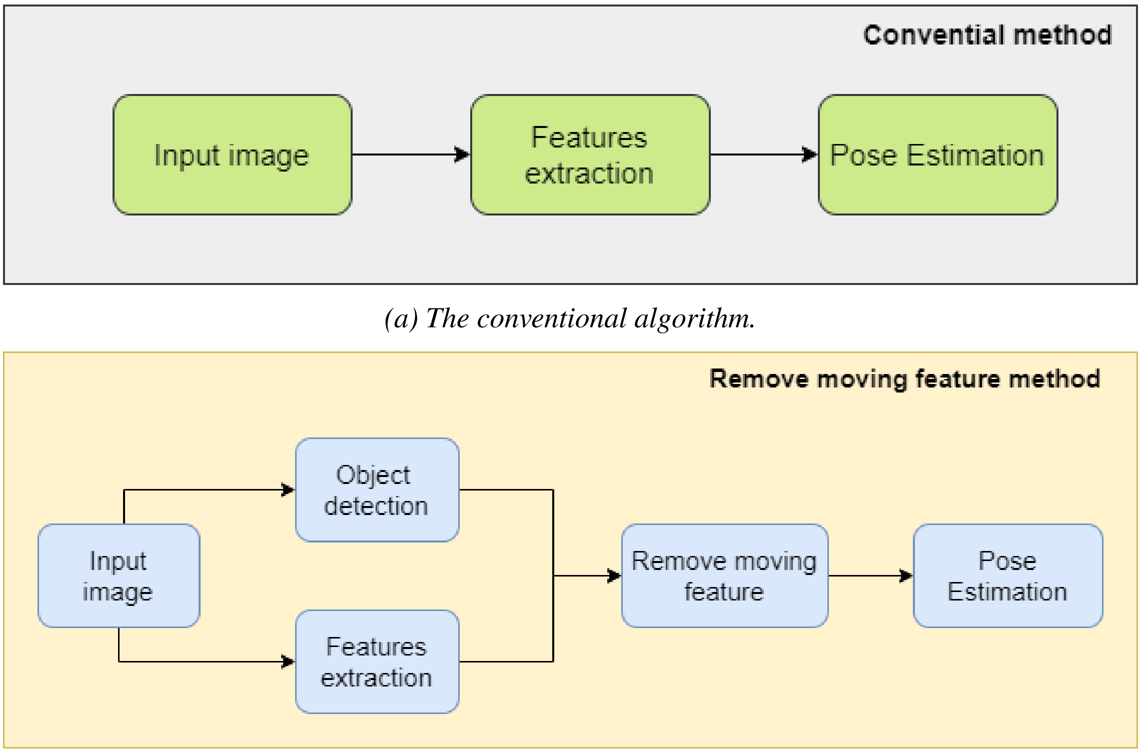
3. Feature selection by removing unstable features

When features are located on moving objects such as other vehicles on the road, they can interfere with the pose estimation process of the self-driving car. These features are often unstable and change continuously, causing the VO system to misinterpret the actual movement of the autonomous vehicle. Consequently, moving features can lead to incorrect calculations of the car's direction and speed, resulting in errors in estimating its current position. Furthermore, tracking and processing features on moving objects increase the computational load and reduce the system's processing speed, negatively impacting the overall performance and reliability of pose estimation. Therefore, to improve performance, it is necessary to remove features on moving objects through methods such as detecting and eliminating moving objects.

In previous studies using feature selection methods, although certain promising results were achieved. In some sequences, the accuracy remained suboptimal such as sequence 2 and sequence 7 [13]. This could potentially affect the accuracy when conducting experiments in real-world environments. These sequences involved a relatively high number of vehicles and very fast movement speeds. Since the points on the vehicles are mobile and moving, selecting these points for pose estimation results in significant errors. Therefore, this article propose a method to remove mobile, fast-moving features, specifically those on the vehicles. The specific implementation of the model is as follows figure 1.

To enhance the accuracy of car detection on the demo dataset, this article utilized the demo data itself to create a training set for the YOLO (You Only Look Once). Distinctive images are selected from each subset of the dataset to enable the model to extract a broader range of features, thereby improving its detection capabilities. These images were labeled using annotation software like Roboflow to tag car data, which were then incorporated into the training process. Various data augmentation techniques also were applied to improve the quality of the training data. Since the task only required filtering out features that appear on cars, the proposed method focused solely on car objects for detection. Finally, the model was trained with approximately 10000 labeled images, capturing diverse features, over 400 epochs to enhance the model's learning capacity. After obtaining the model, that model was used to perform detection on the KITTI data set to create labels corresponding to each image. Label will store the boundingbox values of cars appearing in the image. Each data sequence will have 2 folders, label and image, which are included in the remove moving feature process to filter out all features appearing in the car's boundingbox area based on the included label folder.

The conventional algorithm shown by the gray block, directly feeds extracted features into pose estimation to determine the vehicle's trajectory. In the propose method (yellow block), mobile features firstly are identified and removed before performing pose estimation. The proposed method define mobile features as those located within a vehicle's moving area. Therefore, the first step is to determine bounding boxes around vehicles in the input images.



(a) The conventional algorithm.

(b) The proposed feature selection.

Fig. 1. The proposed feature selection to improve the accuracy.

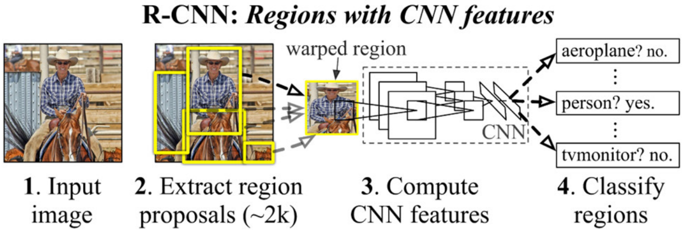


Fig. 2. Operation diagram of the R-CNN model in the Object detection problem.

In cases where machine learning models are employed, such as convolutional nural networks, training on annotated datasets is necessary to teach the model to recognize objects effectively. Once trained or selected, the detection algorithm is applied to the image, resulting in the generation of bounding boxes around detected objects. These bounding boxes are accompanied by class labels (e.g., person, car, dog) and confidence scores, indicating the algorithm's certainty in each detection. Post-processing steps may follow to refine the detection results, such as non-maximum suppression to eliminate redundant bounding boxes. The final output provides crucial information including the coordinates of bounding boxes and associated class labels, which find applications across numerous domains including autonomous driving, surveillance, and medical imaging. This process unfolds as figure 2 [14].

Object detection using YOLO represents a significant advancement in computer vision, particularly in real-time detection tasks. Unlike traditional methods that process images through multiple stages, YOLO operates by analyzing the entire image in a single pass, resulting in remarkable speed and efficiency. YOLO algorithm employs a convolutional neural network architecture to simultaneously predict bounding boxes and class probabilities for objects within an image. To automate vehicle detection, a deep learning model is used – specifically YOLOv8. This provides images with bounding boxes around detected vehicles (Figure 3). Features located on moving objects as car, truck are identified and removed. The resulting feature set, now free of mobile elements, is used for subsequent pose estimation. While YOLO may not offer the absolute highest accuracy compared to other models, its rapid processing speed suits the real-time detection needs. Since the goal is primarily to remove mobile features, the highest possible accuracy isn't a strict requirement.

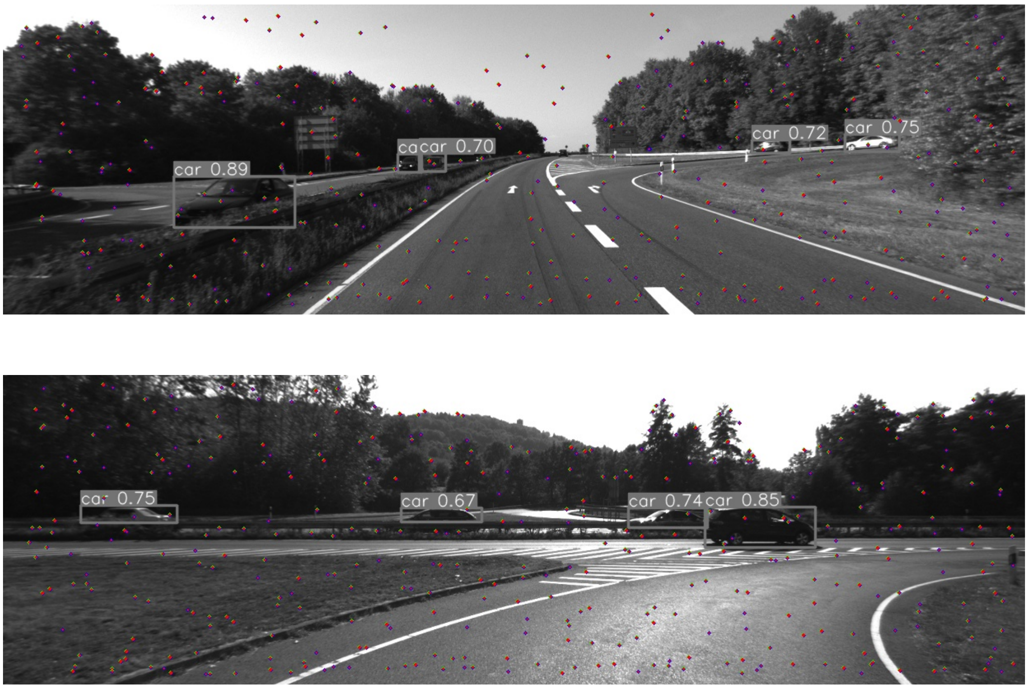


Fig. 3. Image example using Object detection model YOLOv8.

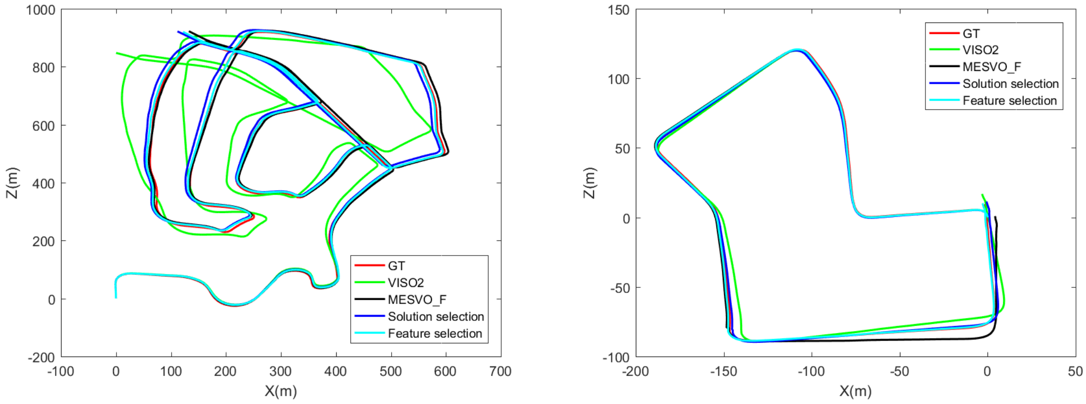
4. Experimental results

The proposed method's performance is assessed using the KITTI dataset, a widely recognized resource in the field of self-driving car research, as referenced by Geiger et al. (2012). This dataset comprises 22 sequences categorized into two sets: a Training dataset (sequences 00-10) and a Testing dataset (sequences 11-21). Within the Training dataset, ground-truth trajectories are provided for every frame across all sequences. Conversely, the Testing dataset lacks ground-truth trajectories, necessitating the retrieval of this information from a publicly accessible website for evaluation purposes, with errors computed automatically. Notably, the dataset encompasses diverse environmental conditions, encompassing variations in speed, lighting, darkness, and the presence of moving objects, facilitating a thorough assessment of algorithmic performance. Additionally, the dataset incorporates a tool for automated performance evaluation, quantifying relative errors (RMSE) encompassing both rotation (RE) and translation (TE) errors. This evaluation methodology computes average errors across sub-sequences of lengths ranging from 100 to 800 meters.

Table 1. Evaluating accuracy on the KITTI dataset for feature selection method

Sec Num	**VISO2**		**MESVO_F**		**Solution selection**		**Feature selection**	
	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	0.84	0.326	0.86	0.296	**0.79**	**0.276**
00	2.46	1.181	**0.78**	0.339	0.83	**0.322**	0.84	0.344
01	-	-	-	-	-	-	-	-
02	2.19	1.808	0.83	0.256	0.81	0.261	**0.75**	**0.256**
03	2.54	1.198	0.77	0.239	**0.71**	0.267	0.75	**0.222**
04	1.02	0.866	0.75	0.222	0.70	0.156	**0.64**	**0.128**
05	2.07	1.124	0.61	0.261	0.59	**0.250**	**0.57**	0.256
06	1.31	0.917	**0.85**	0.311	0.97	0.328	0.97	**0.306**
07	2.30	1.771	1.20	0.850	0.66	**0.394**	**0.66**	0.444
08	2.74	1.336	1.03	0.300	0.99	**0.289**	**0.97**	0.306
09	2.76	1.152	**0.78**	**0.194**	1.03	0.261	0.90	0.206
10	1.64	1.118	0.82	**0.283**	1.29	0.433	**0.79**	0.300

The proposed algorithm is compared with other methods such as VISO2 [15], MESVO_PF [16] and solution selection [13] 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 1. It shows the RMSE of all 11 sequences as well as their average details for the three approaches. Look at table 1, it is clear 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. The proposed algorithm achieved lower average rotation errors compared to its previous versions in multiple sequences. For example, the average rotation errors of MESVO_FP, solution selection, and the proposed method were 0.326, 0.296, and 0.276, respectively, 0.239, 0.261, and 0.222 in sequence 3, and finally, 0.222, 0.156, and 0.128 in sequence 4. Although the proposed method is based on a similar essential matrix approach as



(a) Sequence 2th. *(b) Sequence 7th.*

Fig. 4. Trajectory of sequence 02th, 07th for four methods (VISO2, MESVO_F, Solution selection, Feature selection) compared to the ground truth.

MESVO_FP, however the proposed method reduced the rotation error by about 15% compared to MESVO_FP, and reduced about 7% compared to the solution selection. The translation estimate of MESVO_F, solution selection, and the proposed method are 0.84%, 0.86%, and 0.79%, respectively. Compared to the accuracy of the MESVO_FP and the solution selection, the average translation error of the proposed method reduces the errors by around 6% compared to MESVO_FP and 8% to the solution selection method.

To demonstrate the accuracy, the trajectory of the autonomous vehicle in sequence 2 and sequence 7 of the KITTI dataset were plotted in figure 4. The red line represents the ground-truth trajectory built from GPS and IMU, which is considered the accurate path of the vehicle. The green, black, and blue colors are the results of the VISO2, MESVO_FP, and solution selection methods, respectively. The sky blue line is the result of the proposed method, which is equivalent to the solution selection since it is built upon this method. The trajectory of the proposed method is closer to the ground truth than other methods.

5. Conclusions

This article has investigated the accuracy improvement of stereo VO by removing unstable features where features on other vehicles are discarded by vehicle detection with YOLO. The proposed method evaluated by KITTI dataset and compared with the conventional method indicates that the accuracy is improved by about 6 - 8%. In the future, we will focus on detecting moving objects in the dynamic environment and/or feature selection including close-loop detection and pose graph optimization for SLAM.

References

[1] 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, pp. I–I, doi:10.1109/CVPR.2004.1315094.

[2] 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

[3] 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, pp. 3903–3911, doi:10.48550/arXiv.1708.07878.

[4] 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.

[5] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *2011 IEEE intelligent vehicles symposium (IV)*. IEEE, 2011, pp. 963–968.

[6] 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

[7] 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

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

[9] 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

[10] 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, pp. 85–89, doi:10.1109/IHMSC.2016.212.

[11] Z. L. Yixuan Liu, Xuyang Zhao and C. Yu, "Real-time dynamic SLAM using dynamic odject tracking and key-point filtering," *2023 IEEE International Conference on Unmanned Systems (ICUS)*, 2023. doi: 10.1109/ICUS58632.2023.10318260

[12] 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, pp. 127–137, doi:10.1007/978-3-030-63083-6.

[13] H.-H. Nguyen, A.-D. Vu, Q.-T. Nguyen, and C.-M. Tran, "Solution selection for faster essential matrix based stereo visual odometry," *Journal of Science and Technique-Section on Information and Communication Technology*, vol. 12, no. 02, 2023. doi: 10.56651/lqdtu.jst.v12.n02.750.ict

[14] C. B. Murthy, M. F. Hashmi, N. D. Bokde, and Z. W. Geem, "Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—a comprehensive review," *Applied sciences*, vol. 10, no. 9, p. 3280, 2020. doi: 10.3390/app10093280

[15] 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, pp. 486–492, doi: 10.1109/IVS.2010.5548123.

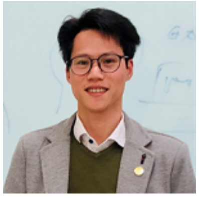
[16] H. H. Nguyen, T. T. Nguyen, X. P. Nguyen, C. M. Tran, and Q. T. Nguyen, "Multiple frame integration for essential matrix-based visual odometry," in *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. IEEE, pp. 1–6, doi:10.1109/IMCOM53663.2022.9721757.

Manuscript received 31-03-2024; Accepted 25-06-2024.

■

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. E-mail: vduc0240@gmail.com

Xuan-Phuc Nguyen received his MS. degree at Le Quy Don Technical University, Viet Nam in control engineering and automation, in 2020. He is currently researcher at Institute of System Integration, Le Quy Don Technical University. His research interests include computer vision, robotics, video compression, deep learning and AI. E-mail: phucnx.isi@lqdtu.edu.vn



Van-Xiem Hoang is the Deputy Head, in charge of the Department of Robotics Engineering, Faculty of Electronics and Telecommunications, Vietnam National University - University of Engineering and Technology (VNU-UET). He received the Ph.D. degree from Lisbon University, Portugal, in 2015, the M.Sc. degree from Sungkyunkwan University, South Korea, in 2011, and the B.E degree from Hanoi University of Science and Technology, Vietnam, in 2009, all in Electrical and Computer Engineering. He is an executive committee member of VNU-UTS Joint Innovation and Technology research center. His research interests are machine learning, image processing, computer vision, video communications, robot vision and signal processing. E-mail: xiemhoang@vnu.edu.vn

Quang-Lam Bui received his Ph.D. degree at Sungkyunkwan University, South Korea in 2014. He is currently researcher at Automotive Engineering Technology, Phu Xuan University. His research interests include computer vision, simultaneous localization and mapping (SLAM), sensor fusion and medical robotics. E-mail:bquanglam@gmail.com

Quang-Hieu Dang received his Ph.D. degree at Le Quy Don Technical University in 2022. He is currently senior researcher of the Institute of System Integration, Le Quy Don Technical University. His research interests include radar navigation, telecommunications and software development. E-mail: hieudq.isi@lqdtu.edu.vn

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), deep learning and AI.E-mail: hungnh.isi@lqdtu.edu.vn

CẢI THIỆN HIỆU SUẤT CỦA ĐO HÌNH ẢNH TRỰC QUAN CAMERA ĐÔI BẰNG CÁCH LOẠI BỎ ĐẶC TRƯNG KHÔNG ỔN ĐỊNH

Vũ Anh Đức, *Nguyễn Xuân Phục*, *Hoàng Văn Xiêm*, *Bùi Quang Lam*, *Đặng Quang Hiệu*, *Nguyễn Hữu Hùng*(*)

Tóm tắt

Đo hình ảnh trực quan (Visual Odometry) bao gồm hai giai đoạn quan trọng: 1) trích xuất đặc điểm và 2) ước tính tư thế. Hiệu suất của Visual Odometry phụ thuộc vào chất lượng của các đối tượng bao gồm số lượng đối tượng, tỷ lệ phần trăm khớp chính xác và vị trí của các đối tượng được phát hiện. Thông thường, phương pháp RANSAC đã được sử dụng trong ước tính tư thế để loại bỏ các ngoại lệ và chọn một tập hợp các tính năng tốt mang lại độ chính xác cao hơn. Tuy nhiên, trong trường hợp số kết quả sai cao hơn, RANSAC dường như đang thất bại. Bài báo đề xuất phương pháp loại bỏ đặc trưng không ổn định bằng phát hiện đối tượng dựa trên học sâu. Phương pháp đề xuất đánh giá trên bộ dữ liệu KITTI cho độ chính xác cao hơn 6 - 8% so với phương pháp thông thường.

Từ khóa

Stereo visual odometry, ma trận cơ sở, phát hiện đối tượng, lựa chọn điểm đặc trưng không ổn định.