Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 455-460, Eylül 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 455-460, September 2020

Research Article

Performance and Trade-off Evaluation of SIFT, SURF, FAST, STAR and ORB feature detection algorithms in Visual Odometry

Abdullah Yusefi[1][1], Akif Durdu[2] and Cemil Sungur[2]

[1] Computer Engineering Department, Konya Technical University, Turkey
[2] Electrical-Electronics Engineering Department, Konya Technical University, Turkey

(1st International Conference on Computer, Electrical and Electronic Sciences ICCEES 2020 – 8-10 October 2020)

(**DOI:** 10.31590/ejosat.819735)

ATIF/REFERENCE: Yusefi, A., Durdu, A. & Sungur, C. (2020). Performance and Trade-off Evaluation of SIFT, SURF, FAST, STAR and ORB feature detection algorithms in Visual Odometry. *European Journal of Science and Technology*, (Special Issue), 455-460.

Abstract

In recent years there has been a great deal of research and study in the field of visual odometry, which has led to the development of practical processes such as visual based measurement in robotics and automotive technology. Direct methods, feature-based methods and hybrid methods are three common approaches in solving visual odometry problems and given the general belief that feature-based approach speeds are higher, this approach has been welcomed in recent years. Therefore, an attempt has been made in the present study to calculate the transformation matrix of two-dimensional sequential image sets using invariant features that can estimate the changes in camera rotation and translation. In the algorithm, two-steps of identifying keypoints and removing outliers are performed using five different local feature detection algorithms (SURF, SIFT, FAST, STAR, ORB) and RANdom SAmple Consensus algorithm (RANSAC), respectively. In addition, the impact of each of them, their intrinsic parameters and dynamic noise on the accuracy of the transformation matrix are evaluated and analyzed in terms of rotational MSE and computational runtime.

Keywords: Visual Odometry, Image Processing, Invariant Features, Local Feature Detection, Keypoints, RANSAC, Transformation Matrix

Introduction

In late decades, researchers have increased a lot of enthusiasm into visual odometry [1, 2] issues, and they have endeavored to give strategies and algorithms to make the process real-time, to improve the accuracy of the outcomes, increase the efficiency of the algorithms and reduce their computation and complexity each of which has its own characteristics [14, 16]. Hence, in this paper, feature-based technique as a general research approach has been chosen to test and study the impacts of common feature detectors and their intrinsic parameters and highlight the trade-off issues between these parameters.

The extraction of invariant local features, which is the method of detecting different small regions in the image such as corners and blobs[2, 5], as well as the selection of suitable algorithms to complement the points extracted[3, 4], play an important role in obtaining the correct results for the visual measurement process and its real-time efficiency.

There are different strategies for phases in the feature-based approaches such as defining points, matching them, and classifying them into two classes of keypoints and outliers. To this end, five common local feature detection algorithms (SURF, SIFT, FAST, STAR, ORB) are chosen to extract key points as well as the RANdom SAmple Consensus algorithm (RANSAC)[13] as a key point classifier to analyze the output of algorithms under various conditions such as dynamic noise. By conducting this comparison research and finally obtaining an acceptable precision transformation matrix, visual odometry application processes in the robotics and automotive technology industries can be followed.

[1] Corresponding Author: Konya Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Konya, Türkiye, ORCID: 0000-0001-7557-8526, e168129001005@ktun.edu.tr

This paper is structured as follows: Section 2 describes the associated visual odometry work [1] through the use of feature detectors. A brief overview of five specific feature detectors is given in Section 3. Section 4 records the findings of the analysis and the assessment. The article shall be concluded with section 5.

2. Related Works

2.1. Feature Detection

Different methods have been suggested to solve the issue of visual odometry using feature detectors. For example, in [1] using the Harris corner detector developed by Harris [9], local image features are identified and used by Nister in visual odometry. Nevertheless, there is a lack of scale-invariance and intensity-invariance [19] which fail when images vary in sizes or pixel intensities or become noisy [10]. A procedure for treating scale and rotation invariance using the principle of Differences of Gaussian (DoG) was introduced in [11]. This feature detector called Scale Invariant Feature Transform (SIFT) used blobs to detect image features. Nonetheless, this method suffered from a high computation issue. In order to resolve this downside, Bay et al. [12] proposed Speeded Up Robust Feature (SURF) which detects features based on the Hessian matrix. Agrawal et al. proposed CenSurE which later STAR detector was derived based on it in [15, 8]. This method has improved computational efficiency compared to SIFT and SURF. In addition, Features from the Accelerated Segment Test (FAST) were implemented in [6], which had the advantage of higher computational performance compared to previous methods [7].

2.2. Feature-based Visual Odometry

In addition to standard SfM, various robust and reliable visual odometry systems have been developed that provide loop detections. Loop detection allows the visual odometry system to track loops and modify the previous trajectories on that basis. The mechanism is outlined in more depth in [20]. They combine the feature detection with the optical flow for visual odometry. The same task was aimed in [21] by looking for repeatable features. It is proposed in [23] that an efficient method of noise removal has had a high impact on feature-based visual odometry approaches such as [24]. In [22], certain robust features are applied to the main system in order to achieve a robot motion calculation. It will assist the application of the Simultaneous Localization and Mapping (SLAM). ORB-SLAM2 is one of the most important SLAM methods based on the ORB features. This technique was developed not only to track the position and detect the loop, but also to reconstruct the 3D model by generating a point cloud of the observed features. The latter function is particularly useful for simultaneous mapping of the world when executing a localization task [19].

2.3. Feature-based Visual Odometry

Numerous publications are available in the literature on the subject of the comparison of features. A contrast of SIFT and SURF was made in evaluating their efficiency [27] but not in visual odometry. The efficiency of Harris, SIFT, SURF and KLT for structure from motion (SfM) was evaluated in [24]. BRISK was suggested in [25] and was compared with feature detectors SIFT and SURF. A drift and translation error comparison of the visual odometry features of SIFT, SURF, ORB and A-KAZE was evaluated in [26].

3. Feature Detectors

3.1. SIFT

SIFT recognizes blobs as local features and has the advantage of scale invariance and rotation invariance [11]. The scale invariance is done by a process called Difference of Gaussians and dividing the scale-space which is a function of *L (x, y,* $\boldsymbol{\sigma}$*)* into different smaller images. Images get smaller every time by half of the previous one and the operation of Gaussian kernel or blurring.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \qquad (1)$$

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \qquad (2)$$

Where *I* is the image, *G* is the Gaussian blur, *x* and *y* are the coordinates and $\boldsymbol{\sigma}$ is the scale factor.

3.2. SURF

SURF is composed of two general steps: First one is the feature extraction that detects the keypoints in the image and the second one is the feature description that is used for feature matching [12]. In order to speed up the computation time SURF uses Hessian matrix-based interest points which after adapting it to scale and rotation invariance is defined as:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \qquad (3)$$

Where *H* is the Hessian matrix and *L(x, σ)* is the Gaussian blur discussed in the previous section.

3.3. FAST

This method considers 16 pixels around the processing pixel to decide whether it's a keypoint or not. This way it doesn't need to process the whole image pixels and thus results in faster computation and real-time detection of features [6, 7].

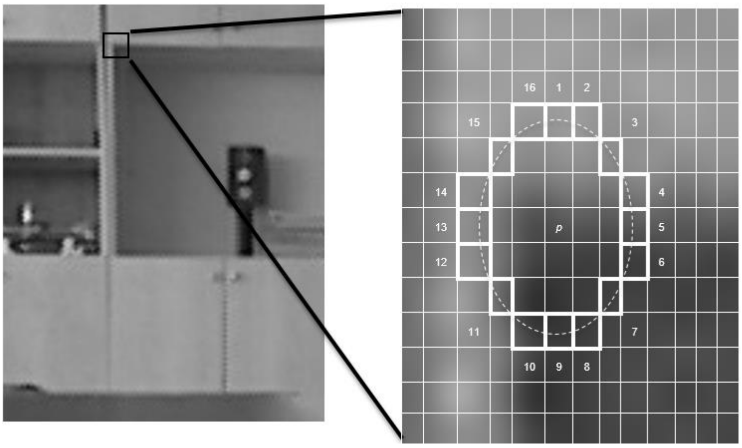

Fig. 1 FAST looks at the 16 pixels around the pixel p to detect features.

3.4. STAR

This feature detector is a derived version of CenSurE and was developed by the OpenCV library. STAR too is a scale and rotation invariant detector that uses Laplacian of Gaussians (LoG) and instead of complete circle masking it takes account two 45-degree difference overlapped squares as an approximation of it [8, 15].

3.5. ORB

Oriented FAST and Rotated BRIEF (ORB) is a combined version of Features from Accelerated Segment Test (FAST) Feature detector and Binary Robust Independent Elementary Features (BRIEF) feature descriptor [17].

4. Performance Evaluation and Trade-Offs

4.1. Performance Evaluation

In evaluating the experimental results, two comparable criteria were the rotational mean square error (MSE) and computational time. This includes the error rate of camera rotation, feature detection techniques and finally how long it took to perform visual odometry. The feature detectors, dataset sequences and their respective rotational MSE is shown in Table I. In the experiments, threshold, number of feature detections and intrinsic parameters were configured as give the best possible translational performance with least rotational MSE. Table II depicts the computational runtime of each sequence of dataset on all feature detectors. The impacts of each feature detector and their tradeoffs are discussed in the next section.

In order to evaluate the performance of the feature detectors in visual odometry, the famous KITTI's grayscale dataset with resolution of 1241 x 376 pixels was used on a system with Intel's 9th generation core i7 that had 16GB RAM.

As can be seen in Table I, the most rotational error has happened in seq_00 of the Kitti dataset and the cause of this high error rate is the movement of other vehicles while the camera was waiting in the traffic light. The reason for this is that traditional visual odometry cannot handle such situations and there was no loop detection or other optimization algorithms implemented in our experiments. Please note that, the same situation happens in seq_07 too. As can be observed, in both times FAST had the best performance with 2.3082 and 0.1434 respectively. However, the least performance was different in two dataset sequences. In the first one the highest error rate was with the ORB feature detector and the in the seq_07 the least performance was with the SURF.

Table 1. MSE

Seq	*Frames*	*SIFT*	*SURF*	*STAR*	*FAST*	*ORB*
00	4540	3.5692	4.59843	4.52235	2.3082	7.90480
01	1100	0.0353	0.13101	0.28460	0.0215	0.03479
02	4660	0.0927	0.08051	0.11423	0.0406	0.08904
03	800	0.0182	0.01759	0.1422	0.0225	0.03328
04	270	0.0006	0.00068	0.0009	0.0005	0.0006
05	2760	0.8082	1.26559	0.94318	0.4613	1.10614
06	1100	0.0185	0.02107	0.01381	0.0312	0.01693
07	1100	3.5233	3.26835	7.19671	0.1434	2.77622
08	4070	0.1634	0.15540	0.11701	0.0394	0.10918
09	1590	0.0322	0.02263	0.13907	0.0223	0.01656
10	1200	0.0636	0.34598	0.19013	0.0510	0.25826

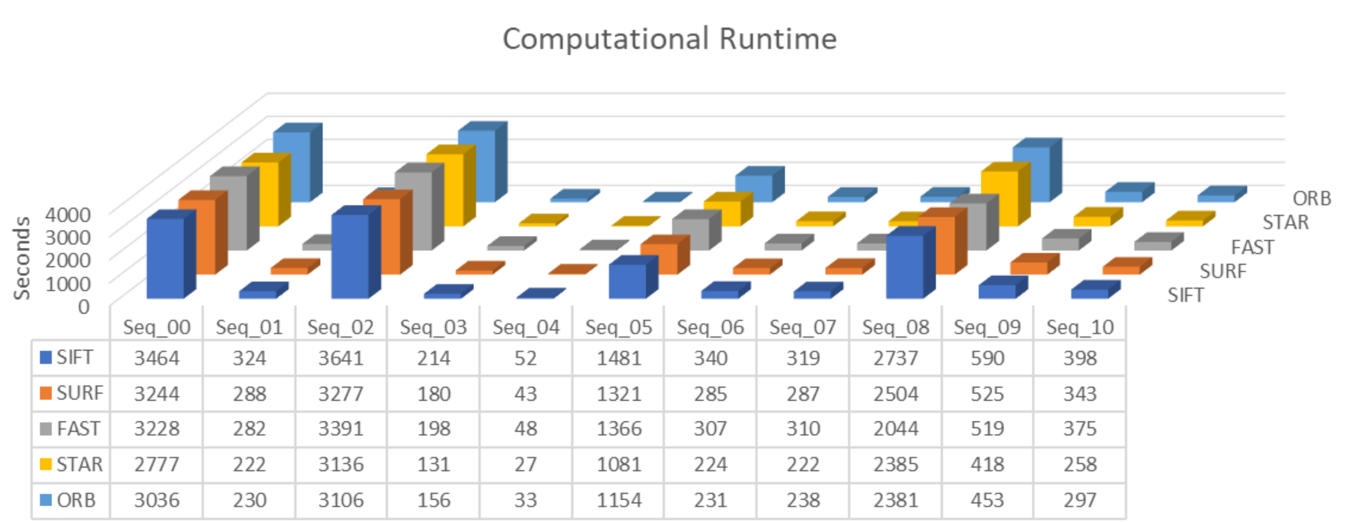

Fig 2. Computational Runtime of visual odometry using five different feature detection algorithms

Moreover, the trajectory diagram of visual odometry of seq_02 using five feature detectors has been given as an example in figure 3. In this figure, all five feature detectors had a reasonable performance when there is less noise and velocity deviation in the dataset. Similarly, as previous sequences the best performance was in FAST and SURF, ORB, SIFT and STAR subsequently.

However, as shown in figure 2 the rates in terms of computational runtime was different that rotational MSE. In computational runtime, the best number was with STAR almost in all dataset sequences and the least performance was with SIFT. In the experiments, it was observed that even if FAST feature detector performed slower compared to STAR in terms of computational runtime, it had the best overall performance when rotational error also was taken in account.

4.2. Trade-Offs

As it is observed from the results in figure 2 the error rate is somewhat related to the amount of dataset. This is because visual odometry is an accumulative method that error rates get accumulated in every frame step. Please note that, the computation runtime gets better in each method through the developed phases as shown in figure 3. The fastest one was STAR and then FAST, SURF, ORB and SIFT subsequently.

In terms of accuracy and reliability, as can be seen in figure 2 the most accurate and reliable one is the FAST algorithm. However, it takes more runtime compared to ORB feature detection. ORB feature detectors are faster compared to other methods but at the cost of a little accuracy loss. In addition to that, all the above methods were observed to be vulnerable against dynamic movement while stable state of the camera. Moreover, the accuracy and the computation runtime of the final visual odometry results are highly dependent on configured parameters such as threshold and number of features detected. For example, the number of features detected in the above experiments were set to 10000 to get the best accuracy. It was observed that, as we minimized the number of features the computation runtime was noticeably decreased however the accuracy of the system decreased too.

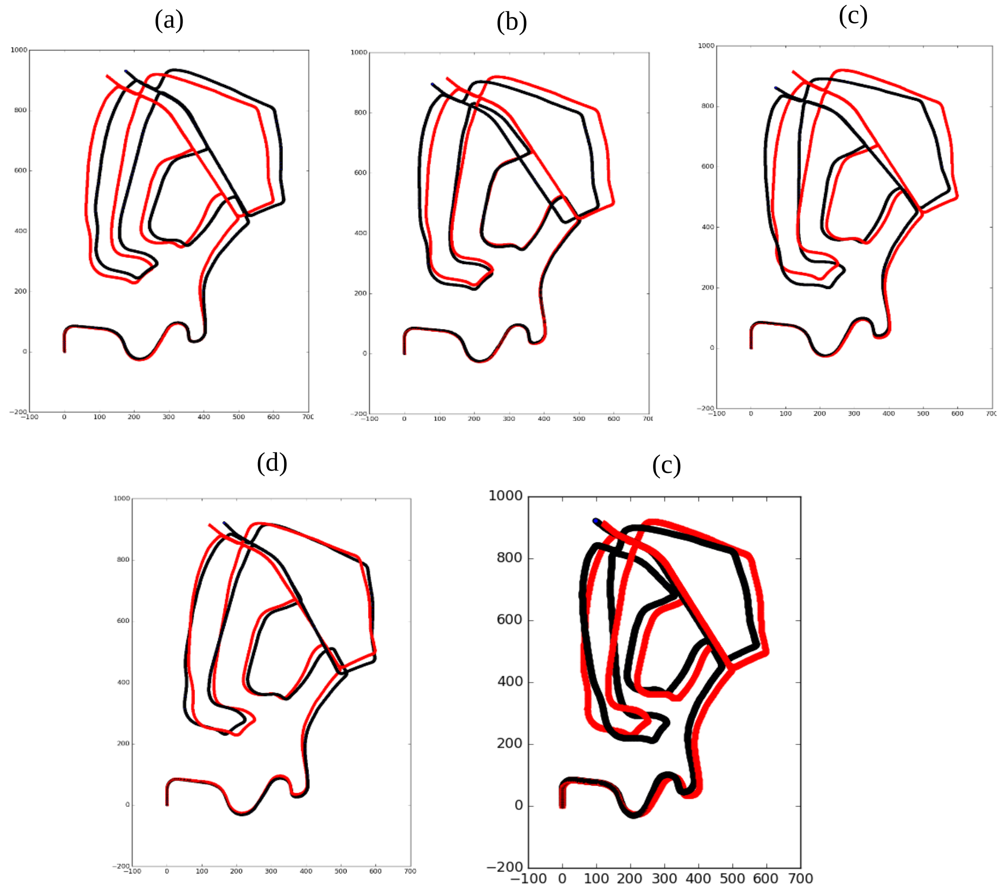

Fig. 3 Visual Odometry Trajectories: (a) STAR, (b) SIFT, (c) ORB, (d) SURF, (e) FAST.

5. Conclusion

This paper studies the impacts of five common feature detectors (SIFT, SURF, STAR, FAST, ORB) and their intrinsic parameters on the performance of transformation matrix extraction in terms of rotational MSE and computation runtime in visual odometry. After the experimental evaluation and analysis of results it was proved that FAST outperformed other feature detectors in the accuracy of visual odometry and the ORB and STAR had better computational runtime with the cost of degradation in accuracy. In addition to that, it was observed that the intrinsic parameters such as configured threshold and number of features detected in the image highly affects the performance and computational efficiency of the visual odometry.

References

[1] Nistér, David, Oleg Naroditsky, and James Bergen. "Visual odometry." Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.. Vol. 1. Ieee, 2004.

[2] Scaramuzza, Davide, and Friedrich Fraundorfer. "Tutorial: visual odometry." IEEE Robotics and Automation Magazine 18.4 (2011): 80-92.

[3] Fraundorfer, Friedrich, and Davide Scaramuzza. "Visual odometry: Part ii: Matching, robustness, optimization, and applications." IEEE Robotics & Automation Magazine 19.2 (2012): 78-90.

[4] Civera, Javier, et al. "1-point RANSAC for EKF-based structure from motion." 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009.

[5] Scaramuzza, Davide. "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints." International journal of computer vision 95.1 (2011): 74-85.
[6] Rosten, Edward, and Tom Drummond. "Machine learning for high-speed corner detection." European conference on computer vision. Springer, Berlin, Heidelberg, 2006.
[7] Rosten, Edward, and Tom Drummond. "Fusing points and lines for high performance tracking." Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. Vol. 2. Ieee, 2005.
[8] Konolige, Kurt, Motilal Agrawal, and Joan Sola. "Large-scale visual odometry for rough terrain." Robotics research. Springer, Berlin, Heidelberg, 2010. 201-212.
[9] Harris, Christopher G., and Mike Stephens. "A combined corner and edge detector." Alvey vision conference. Vol. 15. No. 50. 1988.
[10] Wei, Lijun, et al. "GPS and stereovision-based visual odometry: Application to urban scene mapping and intelligent vehicle localization." International Journal of Vehicular Technology 2011 (2011).
[11] Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.
[12] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." European conference on computer vision. Springer, Berlin, Heidelberg, 2006.
[13] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24.6 (1981): 381-395.
[14] Lindeberg, Tony. "Feature detection with automatic scale selection." International journal of computer vision 30.2 (1998): 79-116.
[15] Agrawal, Motilal, Kurt Konolige, and Morten Rufus Blas. "Censure: Center surround extremas for realtime feature detection and matching." European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2008.
[16] Poddar, Shashi, Rahul Kottath, and Vinod Karar. "Evolution of visual odometry techniques." arXiv preprint arXiv:1804.11142 (2018).
[17] E. Rublee, et al. "ORB: An efficient alternative to SIFT or SURF." 2011 International conference on computer vision. Ieee, 2011.
[18] Klette, Reinhard. Concise computer vision. Springer, London, 2014.
[19] Mur-Artal, Raul, and Juan D. Tardós. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.
[20] Corke, Peter, Dennis Strelow, and Sanjiv Singh. "Omnidirectional visual odometry for a planetary rover." 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). Vol. 4. IEEE, 2004.
[21] Scaramuzza, Davide. "Performance evaluation of 1-point-RANSAC visual odometry." Journal of Field Robotics 28.5 (2011): 792-811.
[22] Nistér, David. "An efficient solution to the five-point relative pose problem." IEEE transactions on pattern analysis and machine intelligence 26.6 (2004): 756-770.
[23] Tardif, Jean-Philippe, Yanis Pavlidis, and Kostas Daniilidis. "Monocular visual odometry in urban environments using an omnidirectional camera." 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008.
[24] Govender, Natasha. "Evaluation of feature detection algorithms for structure from motion." (2009).
[25] Leutenegger, Stefan, Margarita Chli, and Roland Y. Siegwart. "BRISK: Binary robust invariant scalable keypoints." 2011 International conference on computer vision. Ieee, 2011.
[26] Chien, Hsiang-Jen, et al. "When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry." 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ). IEEE, 2016.
[27] Bauer, Johannes, Niko Sünderhauf, and Peter Protzel. "Comparing several implementations of two recently published feature detectors." IFAC Proceedings Volumes 40.15 (2007): 143-148.