

Article

A Descriptor-Based Advanced Feature Detector for Improved Visual Tracking

Kai Yit Kok ¹  and Parvathy Rajendran ^{1,2,*} 

¹ School of Aerospace Engineering, Universiti Sains Malaysia, Engineering Campus, Nibong Tebal 14300, Penang, Malaysia; kok901221kaiyit@yahoo.com

² Faculty of Engineering & Computing, First City University College, Bandar Utama, Petaling Jaya 47800, Selangor, Malaysia

* Correspondence: aeparvathy@usm.my or parvathy.rajendran@firstcity.edu.my

Abstract: Despite years of work, a robust, widely applicable generic “symmetry detector” that can parallel other kinds of computer vision/image processing tools for the more basic structural characteristics, such as a “edge” or “corner” detector, remains a computational challenge. A new symmetry feature detector with a descriptor is proposed in this paper, namely the Simple Robust Features (SRF) algorithm. A performance comparison is made among SRF with SRF, Speeded-up Robust Features (SURF) with SURF, Maximally Stable Extremal Regions (MSER) with SURF, Harris with Fast Retina Keypoint (FREAK), Minimum Eigenvalue with FREAK, Features from Accelerated Segment Test (FAST) with FREAK, and Binary Robust Invariant Scalable Keypoints (BRISK) with FREAK. A visual tracking dataset is used in this performance evaluation in terms of accuracy and computational cost. The results have shown that combining the SRF detector with the SRF descriptor is preferable, as it has on average the highest accuracy. Additionally, the computational cost of SRF with SRF is much lower than the others.



Citation: Kok, K.Y.; Rajendran, P. A Descriptor-Based Advanced Feature Detector for Improved Visual Tracking. *Symmetry* **2021**, *13*, 1337. <https://doi.org/10.3390/sym13081337>

Academic Editors: Moonsoo Ra and Whoi-Yul Kim

Received: 1 July 2021
Accepted: 23 July 2021
Published: 24 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: feature detection; descriptor; SURF; MSER; Harris detector; FAST; BRISK; SRF

1. Introduction

Various feature detectors and descriptors have been developed to solve computer vision problems over the last two decades [1,2]. Given its relative simplicity, the identification of rotation and translation symmetries and their skewed counterparts from processed images has been the main emphasis in computer vision. The main challenge of feature detection and description is to obtain features invariant to affine transformation, scale, rotation, and noise [3,4]. The ideal feature detector can extract similar feature points regardless of the changes of parameters, as mentioned previously. On the other hand, feature descriptors can characterize the feature points or regions with the most specific information. These same features can be matched together with other images [5].

SURF is a popular detector among the existing detectors that is partly inspired by the Scale Invariant Feature Transform (SIFT) [6,7]. It has higher efficiency, especially in computational cost than SIFT [8,9], making it more suitable for real-time applications. Still, there is a detector with comparable performance, such as BRISK [5]. MSER [10,11] is another example that is capable of obtaining good quality features.

The Harris detector is one of the most common corner detectors introduced by Harris and Stephens [12]. This detector's advantage includes simplicity and invariance of transformation, rotation, and lighting [13]. The Minimum Eigenvalue detector, another corner detector, was proposed by Shi and Tomasi [14], which has better stability than the Harris detector. However, the drawback of this method is that the computational cost is higher than the Harris detector. FAST, which was developed by Edward and Tom [15], has high computational efficiency. Additionally, the performance of this detector can

be increased with machine learning techniques, which makes it a good option for video processing applications.

The SURF descriptor is the preferred detector, and has a high matching rate and high-speed performance in terms of the descriptor. FREAK is another excellent example that was proposed by Alexandre Alahi et al. [3]. This descriptor mimics the saccadic search of the human eye, and the descriptor matrix is produced by comparing image intensities over a retinal sampling pattern.

Studies on the performance comparison among feature detectors and descriptors have been done [16,17]. Yet, the computational cost of the combination of detector and descriptor and accuracy still far from ideal performance. Thus, a new detector with a descriptor algorithm (SRF) is developed in this paper to tackle the accuracy and computational cost. It simplifies the process of obtaining feature points without non-maximal suppression. Besides forming a descriptor matrix with complex calculations, SRF characterizes features according to three aspects: location, grayscale intensity, and gradient intensity with weightage factors. This algorithm can achieve better accuracy with a lower computational cost than other existing approaches, which can be implemented in real-life applications, such as surveillance, traffic control, and video communication. In this paper, performance evaluation is done to justify the performance of SRF with other existing algorithms.

2. Related Work

This section presents the overviews of involved detectors and descriptors, including SURF, MSER, Harris, Minimum Eigenvalues, FAST, BRISK, and FREAK. Furthermore, the principle of SRF is introduced.

2.1. SURF

The fundamental concepts of SURF come from SIFT to improve computational efficiency [18]. In this algorithm, the image needs to be converted first to grayscale. Then, the integral image is formed [19] by cumulative summation for the rows and columns of the image matrix. The determinant of the Hessian response map is then built, and feature points are selected from scale-space through non-maximal suppression. Interpolations increase the sub-pixel accuracy of the feature points.

After obtaining the feature points, Haar wavelet responses can be calculated, as well as the dominant orientation of each feature point [20]. Figure 1 displays the orientation search and determination example, whereby the orientation angle is estimated based on the highest value of summed responses. The descriptor vector of feature points is then extracted according to the summation of wavelet responses.

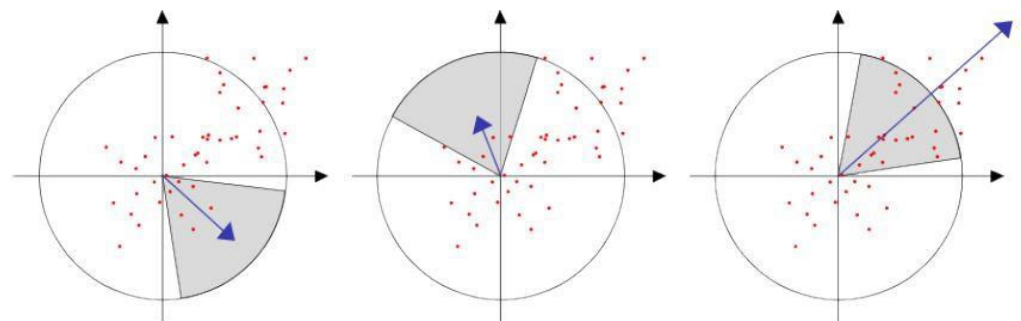


Figure 1. Orientation search and determination of SURF [19].

2.2. MSER

MSER claims that certain regions of grayscale intensity in an image are stable over a high range of thresholds, and these regions will be selected as features [21]. The implementation of MSER [10,11] is done through several steps. Initially, the threshold value is changed from maximum (255) to minimum (0) intensity and extracts the connected

pixels, known as Extremal Regions. Those with the minimum relative growth rate over the threshold range are ‘maximally stable,’ and these regions can be approximated into ellipse shapes. In other words, these ellipse regions are the features of the image.

There are some limitations in obtaining features, such as the minimum and maximum area of a region, area, and similarity. However, since the features are obtained using the intensity function, the features have several characteristics: high stability, invariance to the affine transformation of image intensities, and multi-scale detection (when overlap happens, both fine and large regions are detected).

2.3. Harris Detector

Harris detector is a common and simple corner detector [22]. In this algorithm, the gradient intensity in the x-direction, y-direction, and xy-direction for every single point of the image is calculated. Next, a correlation matrix for each image point is formed [23,24]. Then, the Gaussian function is used to smoothen the components in the matrix to enlarge the robustness of the detection. Eigenvalues are estimated from a hessian matrix to measure the response of each point using Equation (1) [25], whereby λ_1 and λ_2 are the eigenvalues of and k is the sensitivity factor which is normally within [0.04, 0.06]:

$$H = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (1)$$

Only the points with high values for both eigenvalues are selected as corner points or feature points [26]. Nevertheless, computing eigenvalues might lead to severe computational cost. Hence, an approximation, Equation (2) with determinant and trace of M , decreases the cost. Lastly, the response value is compared with the threshold value. The points with a higher value than the threshold are selected as feature points:

$$H = \det(M) - k * \text{trace}(M)^2 \quad (2)$$

2.4. Minimum Eigenvalue Detector

This detector has several similar concepts to the Harris detector. Firstly, the gradient intensity for every image pixel is calculated and forms the correlation matrix. The Gaussian function is used as well to smoothen the elements of the matrix.

The only difference is that the response equation is using eigenvalues as a benchmark, which is shown in Equation (3) [14]:

$$R = \min(\lambda_1, \lambda_2) \quad (3)$$

Again, the image points with higher response values than the threshold will be selected as feature points.

2.5. FAST

Some concepts from the FAST detector are referred to as the Smallest Univalued Segment Assimilating Nucleus (SUSAN) detector [15,27]. It identifies feature points using a circle of 16 pixels with radius 3 [28], shown in Figure 2. In this figure, assuming pixel P is selected for detection, the 16 pixels around pixel P are labeled from number 1 to 16 in a clockwise direction.

Suppose that the contiguous pixels in the circle are brighter than the intensity of pixel P plus a threshold value or all darker than the intensity of pixel P minus the threshold value. In that case, pixel P will be classified as a feature point [29]. Figure 2 illustrates three different states of the 16 pixels around pixel P: similar points (grey), darker points (black), and brighter points (white). This process will be repeated for every single pixel of the image to complete the detection.

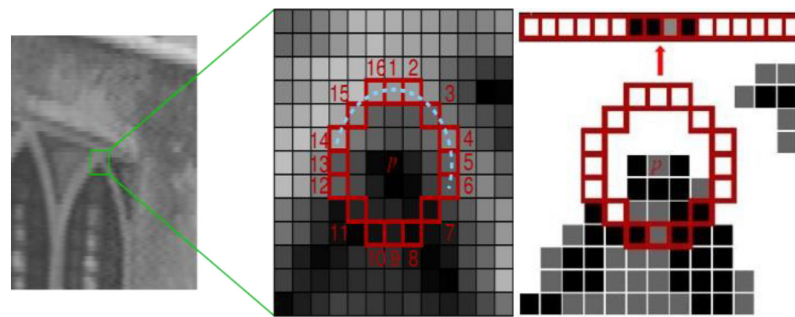


Figure 2. A processed pixel point with a demonstration of the 16 pixels surrounding the processed pixel point in vector form [16].

Multiple features adjacent to each other might be selected; non-maximal suppression will be carried out after detection to remove those not so ‘significant’ feature points. This is done by measuring the sum of the absolute difference between pixels in the contiguous arc and center pixel. Comparing this value with the adjacent feature point, the one with a smaller value will be neglected.

2.6. BRISK

BRISK is referred to as the FAST detector [30]. The scale-space pyramid layers of the image are formed with n octaves and intra-octaves, whereby usually $n = 4$ [3], as shown in Figure 3. The octave layers are produced by half-sampling the image progressively. The first intra-octave layer is formed by downsampling the image with a factor of 1.5, then the rest of the layers are formed using half-sampling.

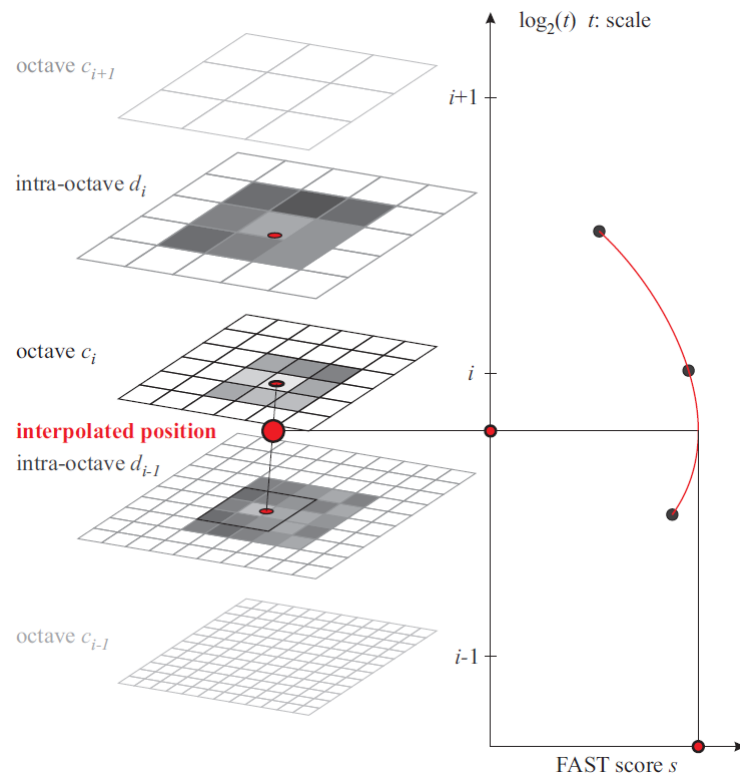


Figure 3. Visualization and interpolation between layers during scale-space features detection [5].

Next, potential regions of the feature are classified by applying the FAST method on all octaves and intra-octave layers individually with the same threshold value. Similar to FAST, non-maximal suppression is carried out on these feature regions for all layers.

For example, one of the feature points in the first-octave layer is compared with neighbor points. If the feature point meets the maximum requirement, it will then be compared with neighbor points in the above and below layers. However, the layers are formed with various discretization, and appropriate interpolation needs to be applied to ensure consistency.

2.7. FREAK

FREAK is a binary descriptor and uses a circular pattern in the form of a retinal sampling grid. The density varies from center to boundary. The points near the center have a high density, but the value drops exponentially for outer points, as shown in Figure 4 [3]. The local gradients of the chosen pairs are summed up to ensure the invariance rotation property.

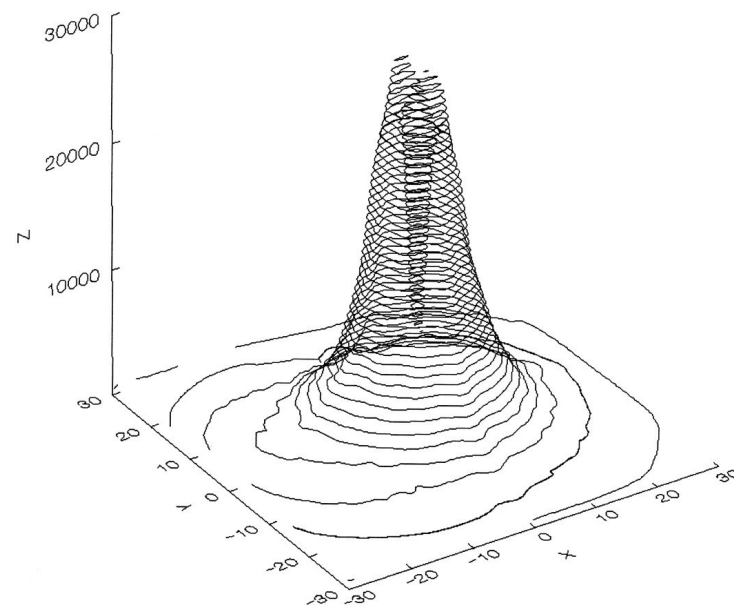


Figure 4. The density of ganglion cells over the retina [31].

Similar to the BRISK method, various kernel sizes are used for each sample point in this model. FREAK differs from BRISK in terms of overlapping receptive fields and exponential changes in size. It was observed that better performance could be archived by changing the size based on the log-polar retinal pattern and overlapping receptive fields.

3. Proposed Algorithm

SRF, the proposed algorithm in this paper, has some concepts inspired by the Harris detector. Compared to other existing feature detectors, the SRF algorithm is more efficient and faster in speed, which is more suitable for real-time applications.

3.1. Feature Points Detection

The first step of SRF is to obtain feature points from the image. This is done by a series of steps, as shown below:

- Conversion to grayscale. The SRF algorithm only works with a 2D image grayscale matrix, and any image format must be converted into a grayscale matrix format before proceeding to the next step. In this analysis, the grayscale conversion is done through the MATLAB built-in function, whereby the RGB to grayscale conversion is done using Equation (4):

$$\text{Grayscale} = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B \quad (4)$$

- Reduction of image resolution. This step aims to decrease the computational cost of the algorithm without affecting the output significantly. Thus, this step is not necessary if there is no restriction on the computational cost. In this stage, the image resolution is reduced to half by removing the even rows and columns. This kind of subsampling will not affect the performance of finding feature points, especially for those large images. Then, the image resolution is decreased to half again by combining four adjacent pixels into one and obtaining the average value. This process can decrease the computational load and image noise simultaneously with the drawback of the blurring effect.
- Calculation of the gradient intensity. This is used by the Harris detector, whereby the gradient intensity of every pixel is estimated in the x-direction, y-direction, and xy-direction using the convolution method with a mask matrix of $[-1 \ 0 \ 1]$ shown in Equation (5):

$$M = \begin{vmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{vmatrix} \quad (5)$$

The only difference compared to the Harris detector is that the Gaussian function is not used in SRF. Similar to a smoothing operation done using the Gaussian function, the SRF algorithm has reduced image resolution that has given the added advantage of the reduced computational time compared to other algorithms. Nevertheless, this process did not compromise accuracy when compared to other algorithms.

- Obtaining feature points. Only the points with large gradient intensity are selected as feature points. The average value from the three gradient intensities is measured with second-order derivatives (I_x^2, I_y^2, I_{xy}), and the threshold is set according to the highest average gradient intensity value to ensure consistent performance regardless of the light intensity of the image, as shown in Equations (6) and (7):

$$I_{average} = (I_{xx} + I_{yy} + I_{xy}) / 3 \quad (6)$$

$$Threshold = 0.1 \times I_{average}^{max} \quad (7)$$

- Filtering the feature points. Feature points within the same region should at least connect one adjacent point using the SRF algorithm. If the number of connected feature points is too small, it is assumed that these feature points are image noise and should be eliminated.

3.2. Features Description

Unlike existing feature descriptors, the SRF algorithm clusters connected feature points as one feature. This clustering concept is based on the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [32]. The points are grouped together according to the distance measurement and the minimum number of points. Other common clustering methods, such as the K-Means Clustering [33], were not used in this paper due to the number of clusters that need to be predefined. The generated clusters' region will be significantly affected by this parameter, which is not robust in dealing with various scenarios. In order to simplify and minimize the computational load, we proposed to set the distance between the points to be 1 (adjacent to each other), while the minimum number of points is set as 4. The feature points from the same region should be adjacent to each other, while the clusters with less than 4 points are considered image noise. Each cluster will form a rectangle boundary based on the outermost point of four sides, which is demonstrated in Figure 5. The characterization also does not use the convention method. A new descriptor matrix is proposed in this algorithm, formed by three elements: location, grayscale pixels, and gradient intensity.

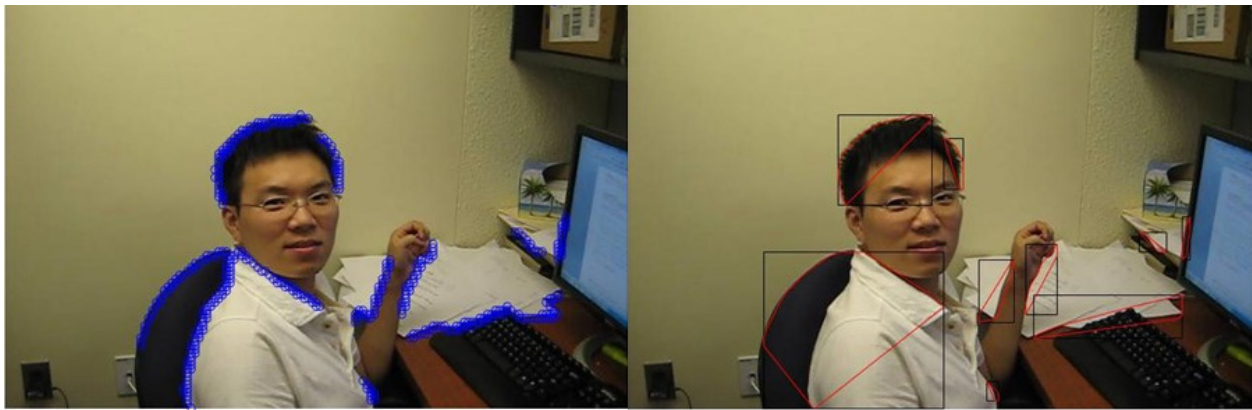


Figure 5. Feature points detection of SRF (left) and forming clusters from feature points in SRF (right).

Location: When features in images move from time to time, the distance traveled is mostly not far from the previous frame, since the time gap between frames is very small. The identification of location property makes matching easier and more accurate most of the time. In this algorithm, the center point coordinate of each cluster is stored in the descriptor matrix.

Grayscale pixels: Not only the distance traveled, but the average value of the grayscale pixels should be around the same for similar features from two frames, since changing of scale and rotation is not obvious within two frames in a limited time. By considering the variants of the scale, the rectangle box of each cluster is divided into 3×3 regardless of the rectangle size, and obtains the average value for each of the nine boxes.

Gradient intensity: As mentioned above, the features' movement is not significant, and thus, the gradient intensity will be changed slightly. An average gradient intensity is calculated from the clusters' rectangle and stored in the descriptor matrix.

In location, grayscale pixels and gradient intensity, images are autocorrelated based on overlap method to determine symmetry detection. These approaches assumes that the evaluated images is either perfectly symmetric or nearly symmetric. Lastly, a weightage factor is added to these three elements, since the impact of each element might not be the same to perform well in features matching. This descriptor can be formulated as shown in Equation (8):

$$D = c_1L + c_2G + c_3I \quad (8)$$

whereby c_1 , c_2 , and c_3 are the weightage factor for each element. Parameter optimization for each weightage factor is done based on the accuracy to obtain the suitable value in features matching. In this parameter optimization, the performance of the weightage factors is tested within a certain range of value, using various image samples from the visual tracker benchmark. The final optimized values of the weightage factors are: $c_1 = 2.4$, $c_2 = 0.09$, $c_3 = 0.8$. Figure 6 illustrates the flowchart of the SRF algorithm.

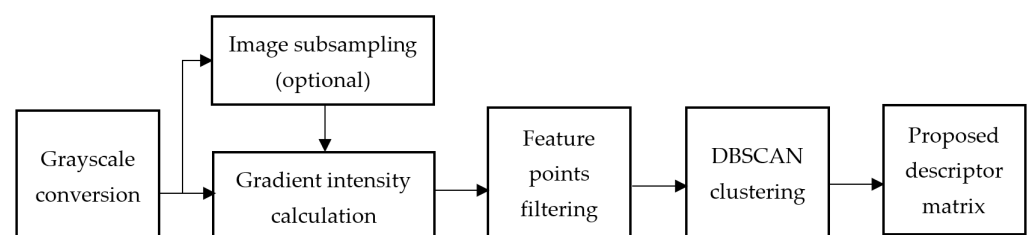


Figure 6. Process flow of the SRF algorithm.

3.3. Performance Evaluation

Two main parameters for the performance comparison of feature detection algorithms are accuracy and computational cost. When capturing image frames, the features of each

frame should be the same, ideally. This means that when matching between frames is done, the higher the matching features, the higher the accuracy. In any case, the number of detected features varies using different methods. Usually, the feature points are detected first in both frames. Then, the feature matching is done to obtain the correct matches between the frames. Next, the matching rate is estimated by determining the number of matched features divided by the average number of features between two frames [34,35]. However, in this work, instead of only two frames of each sample, 98 samples, which accounts for 58,613 frames, have been analyzed to determine the robustness of the proposed algorithm compared to other existing ones.

On the other hand, the computational cost in this research is the total run time of detection and features matching from two frames. The simulation analysis was conducted using MATLAB software, since it can complete tasks easier than other custom software, such as Python. The computer's processor is an Intel i5-4460 64-bit CPU @3.2 GHz, and the RAM is 16 GB. Prior simulation on the seven algorithms analyzed in this work was optimized to ensure that each algorithm performed at its optimized condition when analyzing all 98 image datasets. This ensured that each algorithm was hardware optimized.

4. Results and Discussions

A total of 98 sets of image frames with over 58,000 images in total were used to analyze the performance of SRF and the other six algorithms (i.e., SURF, MSER, Harris, Minimum Eigenvalue, FAST, and BRISK). The images were taken from the visual tracker benchmark online [36,37]. The reason for choosing this database is because of the large number of image frames and various types of images from low resolution to high resolution. Sufficient data can be collected in various ranges of light intensities, image blur, rotations, scales, and viewpoints. Figure 7 presents some of the images from the database.

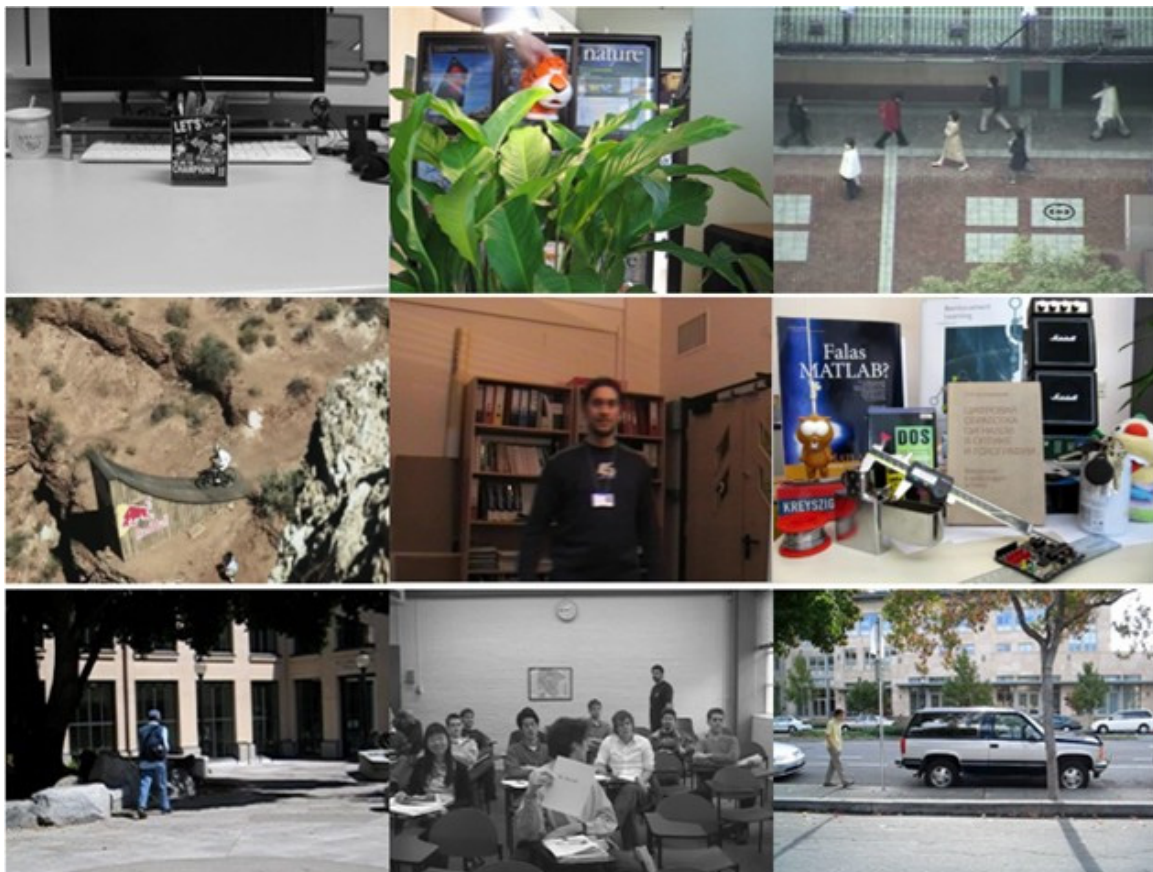


Figure 7. Examples from visual tracking datasets.

Harris, Minimum Eigenvalue, FAST, and BRISK detectors use FREAK for features description, while SURF and MSER detectors use the SURF descriptor in this performance comparison analysis. The SRF detector, however, is using its own descriptor. Figure 8 plotted the accuracy and computational cost of the algorithms in various sets of image frames. In this analysis, the detected features are matched between consecutive frames in each image dataset. Based on the figures, the accuracy and computational cost are not constant for all involved algorithms in different case studies. The image data and the number of detected features are not the same in different images.

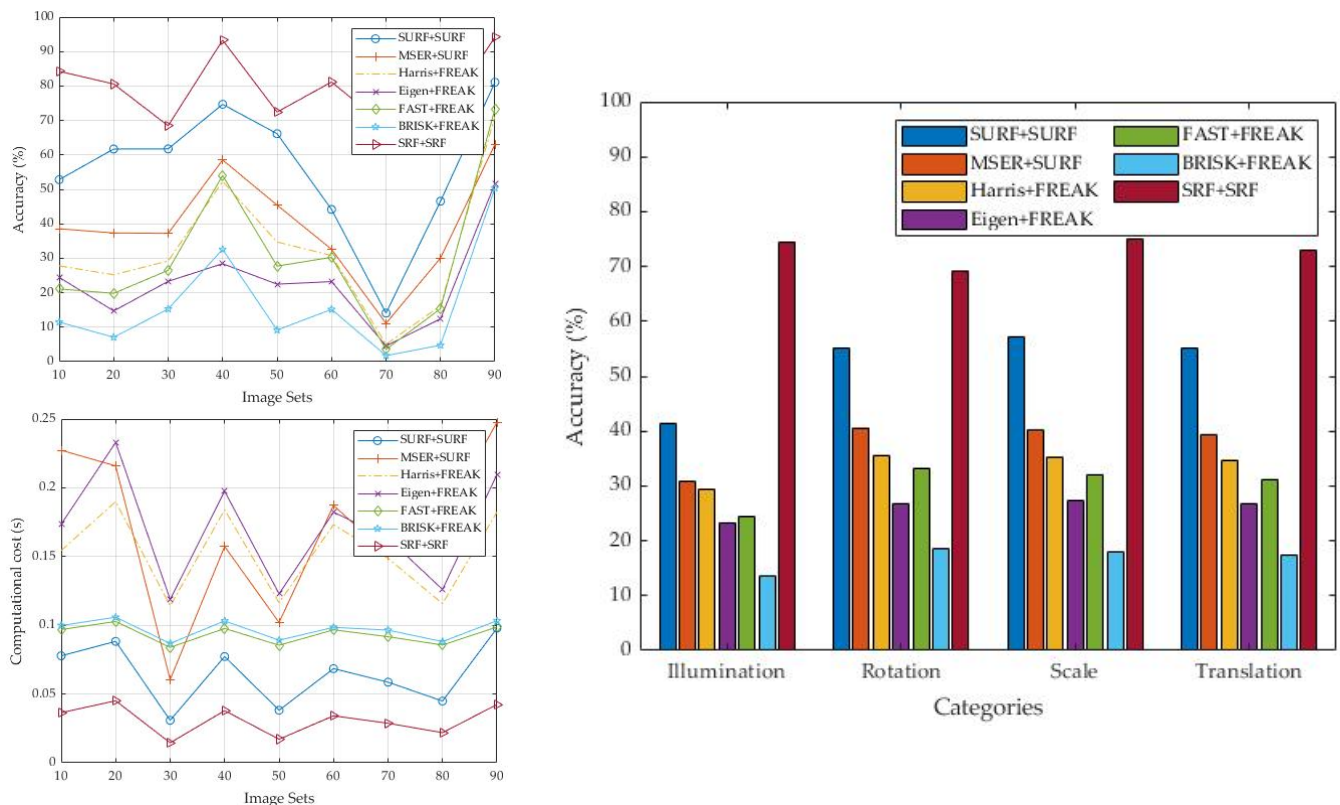


Figure 8. Accuracy and Computational Cost for various image datasets.

It can be observed that the SRF algorithm mostly has the largest accuracy and smallest computational cost. In the SRF algorithm, corner feature points are extracted and the feature points within same region are grouped together to form a cluster. Moreover, those clusters with a number of feature points less than 4 are considered as image noise and will be removed, which is different to the methods in this study. Not only can the accuracy be improved by eliminating the image noise, but the computational cost can also be reduced through forming the clusters, since less work is needed for the feature matching. Based on Figure 8, it can be observed that the accuracy of other approaches is much lower than the SRF method in the 70th image dataset. This is because there are large illumination changes between frames in the image dataset. Overall, the accuracy of SRF is more than 60%, and the majority of the computational cost is less than 0.05 s. The bar graph illustrates the average accuracy of the tested algorithms in these image datasets, based on changes in illumination, rotation, scale, and translation, using the visual tracker benchmark image datasets. Compared to other approaches, the SRF algorithm is more robust to illumination, rotation, scale, and translation changes, with at least 70% of average accuracy. This is mainly because changes in the pixel location are considered in the SRF algorithm, in which the correct match can be found more easily, especially when there are only small changes between frames. Moreover, pixel intensity and gradient intensity with suitable weightage are included in the proposed algorithm, which can improve the features

matching further without burdening the computational load. The combination of these three factors improves the feature matching between two frames significantly, since the closer cluster with high similarity of pixel intensity and gradient intensity is more likely to be matched.

Detailed data are shown in Tables 1 and 2 to estimate the algorithms' average accuracy and computational cost in various sets of images. Table 1 shows that SRF has the highest accuracy, which is 73% on average. SURF has the second-highest accuracy, but this was still almost 20% lower than SRF. The rest of the algorithms have poor accuracy, as they do not even exceed 40%, especially since the BRISK algorithm only has an accuracy of 17%.

Table 1. Estimation of the average accuracy for each of the algorithms.

Algorithm	Accuracy (%)						
	1st Set	2nd Set	3rd Set	...	97th Set	98th Set	Average
SURF	60.7	83.1	32.9	...	87.2	63.8	55.4
MSER	40.1	65.8	27.0	...	82.8	47.7	39.6
Harris	28.9	75.9	12.0	...	85.8	53.8	34.7
Eigen	24.1	66.7	9.5	...	85.3	42.2	26.7
FAST	22.9	71.5	9.8	...	87.8	48.1	31.3
BRISK	7.0	48.8	5.2	...	80.1	26.6	17.4
SRF	64.4	95.5	73.2	...	93.0	82.1	73.1

Table 2. Estimation of average computational cost for each of the algorithms.

Algorithm	Computational Cost (s)						
	1st Set	2nd Set	3rd Set	...	97th Set	98th Set	Average
SURF	0.03	0.04	0.06	...	0.08	0.11	0.06
MSER	0.07	0.10	0.11	...	0.22	0.36	0.14
Harris	0.11	0.11	0.15	...	0.17	0.16	0.14
Eigen	0.11	0.11	0.15	...	0.25	0.19	0.16
FAST	0.08	0.08	0.09	...	0.10	0.10	0.09
BRISK	0.08	0.09	0.10	...	0.10	0.10	0.09
SRF	0.01	0.02	0.02	...	0.04	0.05	0.03

In terms of computational cost, again, SRF has the best performance, with only 0.01 s. Using SRF as the benchmark, SURF is at least 100% longer than SRF, while FAST and BRISK are more than 200% longer. The MSER, Harris, and Minimum Eigenvalue algorithms have a much higher computational cost, which is at least 400% longer than SRF. Hence, regardless of the accuracy or computational cost, SRF has the best performance among SURF, MSER, Harris, Minimum Eigenvalue, FAST, and BRISK. Figures 9 and 10 illustrate some examples of features detection and matching throughout this analysis. In order to provide a reliable indicator of the algorithm performance, image transformation, such as rotation and translation, and scale and luminance are demonstrated in Figures 9 and 10, respectively.

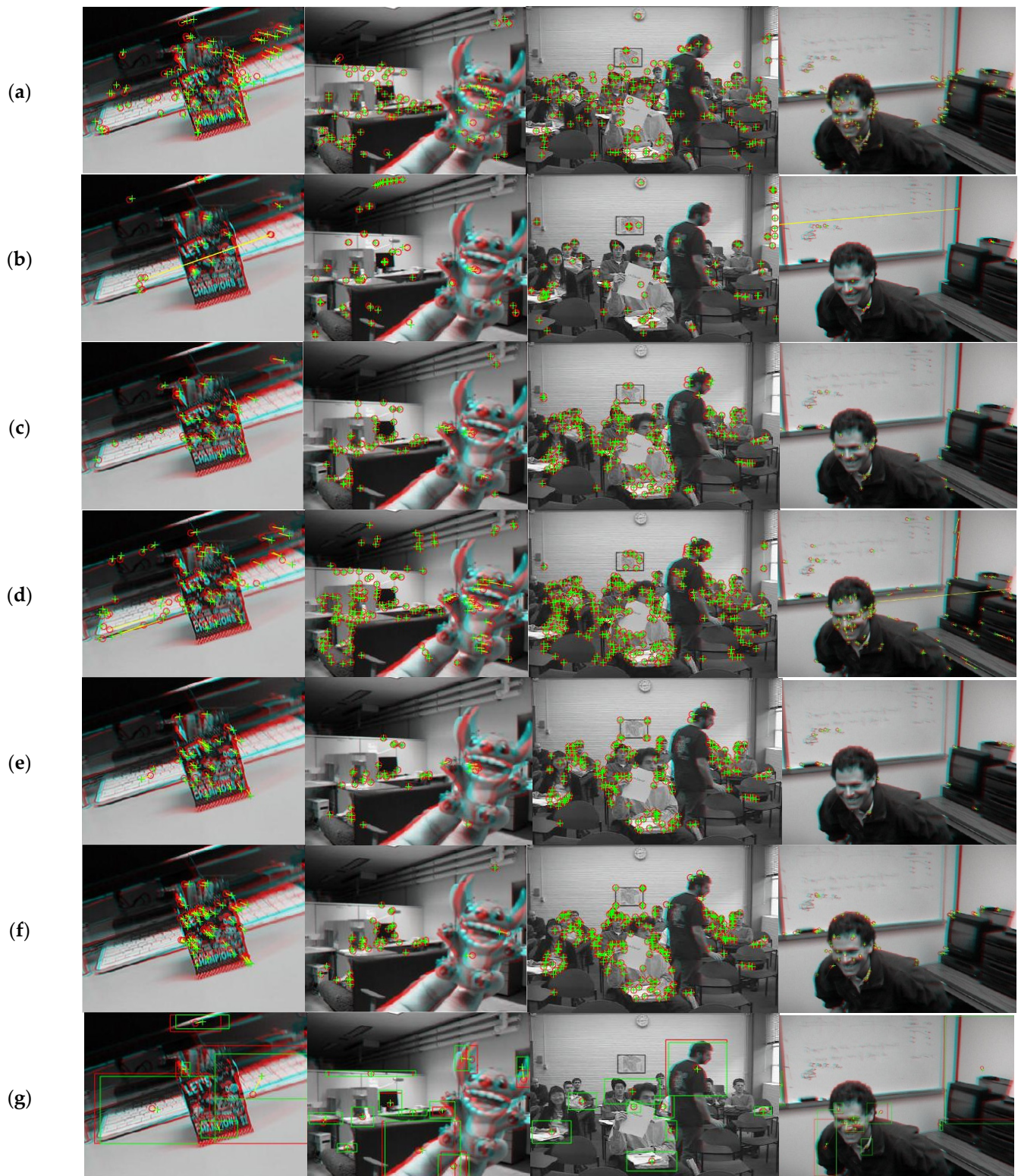


Figure 9. Features matching (changes in rotation and translation) (a) SURF, (b) MSER, (c) Harris, (d) Eigen, (e) FAST, (f) BRISK, (g) SRF.

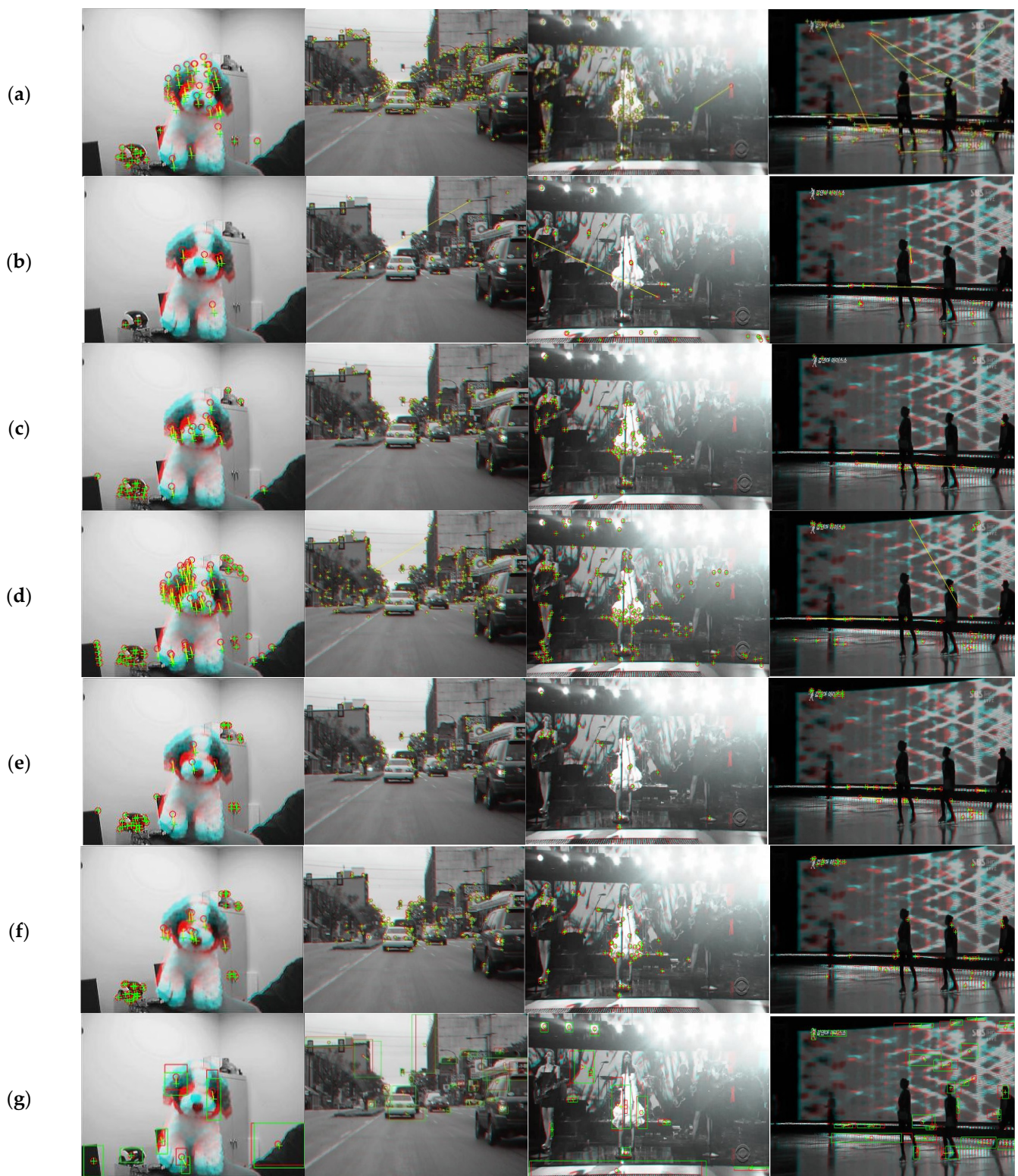


Figure 10. Features matching (changes in scale and luminance) (a) SURF, (b) MSER, (c) Harris, (d) Eigen, (e) FAST, (f) BRISK, (g) SRF.

5. Conclusions

In conclusion, SRF has a 73% average accuracy, which is the highest as compared to the SURF, MSER, Harris, Minimum Eigenvalue, FAST, and BRISK detectors. The accuracy is nearly 20% higher than SURF and more compared to other detectors. In terms of

computational cost, SRF is only half of SURF and even less than the other detectors. Thus, it can be expected that SRF has a higher performance than the others even when converted to other languages for applications or analyses. Still, the accuracy of the SRF method might be affected significantly when there is a large asymmetrical changes in rotation, translation, scale, or lamination change between frames. This is because it is designed to perform symmetry detection detect and match feature points between consecutive frames, where they are expected to show only small changes in pixel location, intensity value, and gradient intensity.

Author Contributions: Conceptualization, K.Y.K. and P.R.; methodology, K.Y.K. and P.R.; software, K.Y.K.; validation, K.Y.K. and P.R.; formal analysis, K.Y.K. and P.R.; investigation, K.Y.K. and P.R.; resources, K.Y.K.; data curation, K.Y.K. and P.R.; writing—original draft preparation, K.Y.K.; writing—review and editing, K.Y.K. and P.R.; visualization, K.Y.K. and P.R.; supervision, P.R.; project administration, P.R.; funding acquisition, P.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Universiti Sains Malaysia, Grant No. 1001/PAERO/8014120, and the APC was funded by the Universiti Sains Malaysia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The authors confirm that the data supporting the findings of this study are available within the article. The dataset(s) supporting the conclusions of this article is (are) available in the Berkeley Segmentation Data Set repository http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/BSR/BSR_bsds500.tgz, (accessed on 20 October 2020).

Acknowledgments: The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Thakur, N.; Han, C. An Ambient Intelligence-Based Human Behavior Monitoring Framework for Ubiquitous Environments. *Information* **2021**, *12*, 81. [CrossRef]
2. Mouats, T.; Aouf, N.; Nam, D.; Vidas, S. Performance Evaluation of Feature Detectors and Descriptors Beyond the Visible. *J. Intell. Robot. Syst.* **2018**, *92*, 33–63. [CrossRef]
3. Alahi, A.; Ortiz, R.; VanderGheynst, P. FREAK: Fast Retina Keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 510–517.
4. Zhu, Y.; Tong, M.; Jiang, Z.; Zhong, S.; Tian, Q. Hybrid feature-based analysis of video's affective content using protagonist detection. *Expert Syst. Appl.* **2019**, *128*, 316–326. [CrossRef]
5. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary robust invariant scalable keypoints. Proceedings of Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
6. Varytimidis, C.; Rapantzikos, K.; Avrithis, Y.; Kollias, S. α -shapes for local feature detection. *Pattern Recognit.* **2016**, *50*, 56–73. [CrossRef]
7. Lee, M.H.; Cho, M.; Park, I.K. Feature description using local neighborhoods. *Pattern Recognit. Lett.* **2015**, *68*, 76–82. [CrossRef]
8. Wang, Y.; Fan, J.; Qian, C.; Guo, L. Ego-motion estimation using sparse SURF flow in monocular vision systems. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 1729881416671112. [CrossRef]
9. Kang, T.-K.; Choi, I.-H.; Lim, M.-T. MDGHM-SURF: A robust local image descriptor based on modified discrete Gaussian–Hermite moment. *Pattern Recognit.* **2015**, *48*, 670–684. [CrossRef]
10. Nistér, D.; Stewénius, H. Linear Time Maximally Stable Extremal Regions. In Proceedings of the Transactions on Petri Nets and Other Models of Concurrency XV; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2008; pp. 183–196.
11. Obdržálek, D.; Basovník, S.; Mach, L.; Mikulík, A. Detecting Scene Elements Using Maximally Stable Colour Regions. In Proceedings of the Communications in Computer and Information Science; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2010; pp. 107–115.
12. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; Volume 15, pp. 147–151.
13. Parks, D.; Gravel, J.-P. Corner detection. *Int. J. Comput. Vision* **2004**, 1–17.
14. Shi, J.; Tomasi, C. Good Features to Track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, DC, USA, 21–23 June 1994; pp. 593–600.

15. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In Proceedings of the Transactions on Petri Nets and Other Models of Concurrency XV; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2006; pp. 430–443.
16. Işık, Ş. A Comparative Evaluation of Well-known Feature Detectors and Descriptors. *Int. J. Appl. Math. Electron. Comput.* **2014**, *3*, 1–6. [[CrossRef](#)]
17. Loncomilla, P.; Ruiz-Del-Solar, J.; Martínez, L. Object recognition using local invariant features for robotic applications: A survey. *Pattern Recognit.* **2016**, *60*, 499–514. [[CrossRef](#)]
18. Wu, Y.; Su, Q.; Ma, W.; Liu, S.; Miao, Q. Learning Robust Feature Descriptor for Image Registration with Genetic Programming. *IEEE Access* **2020**, *8*, 39389–39402. [[CrossRef](#)]
19. Jurgensen, S.M. *The Rotated Speeded-Up Robust Features Algorithm (R-SURF)*; Naval Postgraduate School: Monterey, CA, USA, 2014.
20. Chen, L.-C.; Hsieh, J.-W.; Yan, Y.; Chen, D.-Y. Vehicle make and model recognition using sparse representation and symmetrical SURFs. *Pattern Recognit.* **2015**, *48*, 1979–1998. [[CrossRef](#)]
21. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767. [[CrossRef](#)]
22. Bok, Y.; Ha, H.; Kweon, I.S. Automated checkerboard detection and indexing using circular boundaries. *Pattern Recognit. Lett.* **2016**, *71*, 66–72. [[CrossRef](#)]
23. Příbyl, B.; Chalmers, A.; Zemčík, P.; Hooberman, L.; Čadík, M. Evaluation of feature point detection in high dynamic range imagery. *J. Vis. Commun. Image Represent.* **2016**, *38*, 141–160. [[CrossRef](#)]
24. Elfakiri, Y.; Khaissidi, G.; Mrabti, M.; Chenouni, D.; El Yacoubi, M. Word spotting for handwritten Arabic documents using Harris detector. In Proceedings of the 2016 International Conference on Information Technology for Organizations Development (IT4OD), Fez, Morocco, 30 March–1 April 2016; pp. 1–4.
25. Orguner, U.; Gustafsson, F. Statistical Characteristics of Harris Corner Detector. In Proceedings of the 2007 IEEE/SP 14th Workshop on Statistical Signal Processing, Madison, WI, USA, 26–29 August 2007; pp. 571–575.
26. Hassaballah, M.; Abdelmgeid, A.A.; Alshazly, H.A. Image Features Detection, Description and Matching. In *Econometrics for Financial Applications*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2016; pp. 11–45.
27. Rosten, E.; Porter, R.; Drummond, T. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 105–119. [[CrossRef](#)]
28. Sharif, H.; Hölzel, M. A comparison of prefilters in ORB-based object detection. *Pattern Recognit. Lett.* **2017**, *93*, 154–161. [[CrossRef](#)]
29. Fularz, M.; Kraft, M.; Schmidt, A.; Kasiński, A. A High-Performance FPGA-Based Image Feature Detector and Matcher Based on the FAST and BRIEF Algorithms. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 141. [[CrossRef](#)]
30. Persson, A.; Loutfi, A. Fast Matching of Binary Descriptors for Large-Scale Applications in Robot Vision. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 58. [[CrossRef](#)]
31. Garway-Heath, D.F.; Caprioli, J.; Fitzke, F.W.; Hitchings, R. Scaling the hill of vision: The physiological relationship between light sensitivity and ganglion cell numbers. *Investig. Ophthalmol. Vis. Sci.* **2000**, *41*, 1774–1782.
32. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. Density-based spatial clustering of applications with noise. *Int. Conf. Knowl. Discov. Data Min.* **1996**, *240*, 6.
33. Yuan, C.; Yang, H. Research on K-Value Selection Method of K-Means Clustering Algorithm. *J* **2019**, *2*, 226–235. [[CrossRef](#)]
34. Liu, J.; Bu, F. Improved RANSAC features image-matching method based on SURF. *J. Eng.* **2019**, *2019*, 9118–9122. [[CrossRef](#)]
35. Setiawan, A.; Yunmar, R.A.; Tantriawan, H. *Comparison of Speeded-Up Robust Feature (SURF) and Oriented FAST and Rotated BRIEF (ORB) Methods in Identifying Museum Objects Using Low Light Intensity Images*; IOP Conference Series: Earth and Environmental Science; IOP Publishing: South Lampung, Indonesia, 2020; Volume 537, No. 1; p. 012025.
36. Wu, Y.; Lim, J.; Yang, M.-H. Online Object Tracking: A Benchmark. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
37. Luo, M.; Zhou, B.; Wang, T. Multi-part and scale adaptive visual tracker based on kernel correlation filter. *PLoS ONE* **2020**, *15*, e0231087. [[CrossRef](#)] [[PubMed](#)]