ANALOGY OF POINT FEATURE EXTRACTION TECHNIQUES

Mr Vivek A, Velayudhan A, Neeraj S, Nandakishore R Nair, Vishnu S

Assistant Professor, Amrita School of Engineering, Amrita Vishwa Vidyapeetham Amritapuri Kollam,

Student, Amrita School of Engineering, Amrita Vishwa Vidyapeetham Amritapuri Kollam

Student, Amrita School of Engineering, Amrita Vishwa Vidyapeetham Amritapuri Kollam

Student, Amrita School of Engineering, Amrita Vishwa Vidyapeetham Amritapuri Kollam

Student, Amrita School of Engineering, Amrita Vishwa Vidyapeetham Amritapuri Kollam

Email : viveka@am.amrita.edu, velayudhana@am.students.amrita.edu, neerajs@am.students.amrita.edu, nandakishorernair@am.students.amrita.edu, vishnu@am.students.amrita.edu

Contact: 9995019854, 8281129345, 8590642302, 8304912598

ABSTRACT- Image processing is a significant section of robotic and machine learning. The field of image processing is significant in terms of usage in the advanced world ranging from security, facial recognition to remote autonomous terrain mapping. The salient problem in image processing is how to detect, identify and recognize the feature points in an image. However, there are numerous feature point extraction techniques. In this paper, we are comparing the different available point feature extraction techniques currently in existence based on the scale changes, noise, rotation etc. The evaluation is based on the number of feature points from the reference image as well as matched points in the wild frame and elapsed time. The data and information from the study open up new research in the field of image processing based on point feature extraction.

Index terms **: Descriptors, feature extraction, interest point.**

I. INTRODUCTION

The human brain is capable of extracting, processing and manipulating data efficiently as information based on the neural pathways residing over our nervous system with the influence of neurons. However, compared to the machine point of view this concept of identification is relatively harder as they are trained to perform over an algorithm for fast computation. For example, computers can perform large amounts of calculation in milliseconds which would be a lifetime process for human brains. In contrast to that, humans can identify, differentiate, categorise the world around them which is harder from the machine's perspective. The artificial intelligence concept where the computers think like humans is the near future, but for the computer vision to attain the capability to identify, differentiate, categorise the world, it has to be trained under supervised or unsupervised learning. Image processing with feature extraction is an inevitable part of the machine training process and it is essential to find the most efficient feature extraction algorithm for every possible scenario. The concept of feature extraction works significantly in hand with image processing.

II. FEATURE DESCRIPTORS

1. BINARY ROBUST INVARIANT SCALABLE KEYPOINTS

The Binary Robust Invariant Scalable Keypoints algorithm is also known as BRISK is a feature point detection and description algorithm that is orthogonal in terms of scales and rotations. Brisk builds the binary feature descriptor of a given image by crossmatching the grayscale relation of different pair points in the image. An ideal interest point descriptor recognises the most prominent information content contained within the uncovered salient region and can recognise the same information content when encountered.

The BRISK algorithm can be grouped into three main processes, namely Scale-Space Keypoint Detection, Keypoint Description and Descriptor Matching[1]. In the Scale-Space Keypoint Detection, the saliency of the image is considered a continuous quantity throughout the scaling dimension and a sub-pixel and continuous scale refinement is performed for each of the detected maximum contrast points. In a keypoint description, the characteristic direction of each interest point is identified to achieve rotation invariance[2]. The brightness comparisons to aid the descriptiveness also helps in achieving robustness. The third process is Descriptor matching which can be expressed as a bitwise XOR followed by a bit count which can be achieved with great ease in today's computer architecture.

2. FEATURES FROM ACCELERATED SEGMENT TEST

Features from accelerated segment tests commonly known as FAST is a common corner point detection based feature extraction algorithm that gained momentum due to its computational efficiency.FAST corner detector was developed by Edward Rosten and Tom Drummond in 2006. Moreover, the FAST method produced superior performance in terms of computational time and resources when machine learning methods were applied, contributing to the ascension of this method. The FAST corner point detectors are also found to be efficient for real-time video processing implementations due to their rapid performance.

The interest point detection in FAST can be explained as :

1. Selection of a pixel from the given image, considering the pixel has intensity I the pixel has to test if its an interesting point or not.
2. An approximate threshold value is selected for the image and a circle containing 16 pixels is considered around the selected pixel.
3. Now the selected pixel is an interesting point if the circle contains at least 12 continuous pixels that have intensity within a range of threshold from the intensity of the pixel to be tested.
4. To increase the efficiency of the method 4 pixels at a distance of 3 pixels from each other are compared first. If any 3 of these 4 pixels does not fall in the range of required intensity the given point is not a point of interest.
5. The above process is repeated on every pixel in the image[3].

One of the limitations of this method is the difficulty in finding adjacent maximum points and it is solved using non-maximal suppression(a method involving the assignment of score value to each of the detected interest points). Since the detected interest point must have a ring of darker or lighter pixel values around the centre FAST does not work well with crisp images and computer-generated images.

3. MINEIGEN

This method is a modified version of the Harris Corner Detector and was proposed by Shi and Thomas. Instead of using the score function as used in Harris Detector, MinEigen uses the Eigenvalues, which is used to calculate the score, to decide the corners[4].

The corner is calculated in MinEigen using the equation:

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

----- eqn[1]

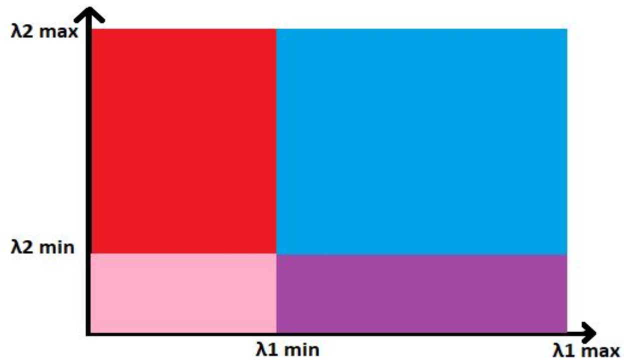

Figure 1. Min Eigen

Here, the blue region has both the eigenvalues greater than a certain value and is considered to be region accepted as corners.

The red and violet regions have either of the eigenvalue less than the minimum required and is considered an edge.

The pink area has both eigenvalues less than the minimum and is a flat area.

4. MAXIMALLY STABLE EXTREMAL REGION

MSER is used in the detection of blobs in images. This algorithm extracts several covariant regions called MSER(Maximally Stable Extremal Region). Its basic idea is to target regions that stay almost constant for a wide range of threshold values.

For a particular threshold value, the pixels with a value lower than the threshold are taken as white and pixels that have a value equal to or greater than the threshold are taken as black.

The following steps are involved in the extraction of MSER:

- The threshold intensity of the image is done by going back and forth from black to white which is called luminous thresholding.
- The related components are extricated, i.e. the extremal areas.
- Detect a threshold for which the extremal areas is Stable Maximally. A region is maximal even if the region above/below might be concurrent with the actual area.
- These MSER regions are then approximated within an ellipse.
- These region's descriptors are kept as features[5].

There are chances that a maximally stable extremal region is still rejected under the following conditions:

- ➔ The region is too big (according to the MaxArea parameter).
- ➔ The region is too small (according to MinArea).
- ➔ The region is too unstable (according to MaxVariation).
- ➔ The region is too similar to the parent MSER[6]

5. ORIENTED FAST AND ROTATED BRIEF

Oriented fast and Rotated Brief is a local feature descriptor having attracted major recognition from the existing SLAM scheme[7] with increased processing efficiency and having solid concurrent performance. It is a combined descriptor having a FAST keypoint detector and an enhanced variant of visual descriptor BRIEF(Binary Robust Independent Elementary Features). ORB was first presented by Ethan Rublee in 2011.

The ORB algorithm consists mainly of three major steps mainly feature point extraction, generating feature point descriptors and feature point matching. In the first step of the ORB image detection algorithm sequence the image feature point detection, feature point screening, image scale pyramid construction and sampling, feature point direction determination. The next step involves descriptor calculation from the ORB feature extracted points extracted from the previous step. Descriptor calculation[7] is the scale and rotation information regarding the image feature points. In the last step, the retrieved descriptors for feature extracted points need to be compared between two different time images to check if both of them matches.

6. SPEEDED UP ROBUST FEATURES

The normal steps of image cross-matching which are followed by the Speeded Up Robust Features(SURF) method are feature extraction, descriptor analysis and image cross-matching.

SURF extraction method uses a distribution-based descriptor and measure based on the Hessian matrix[8] for the detector. Irrespective of orientation the detector autonomously extracts features that must be extracted using a hessian based blob detector to seek out interest points. The detector is used to subtract the pyramid layers obtained using sub-sampling and Gaussian kernels smoothing[9].

Haar wavelet responses along with integral images are used efficiently with the descriptor. The area around the focal point is reoriented before the descriptor calculation.

7. HARRIS CORNER FEATURE EXTRACTION

Harris detector is a corner detection algorithm that is remarkably utilized in computer image processing algorithms to extricate the corners points and to extract the features of an image. The algorithm was improvised over the existing Moravec's corner detector. Compared to the previous algorithms, the Harris corner detector considers the differential of the corner score with relevance to direction directly, rather than mistreatment of shifting patches for each 45-degree angle. As the method is observed to be efficient in distinguishing between edges and corners, it has been improved and adopted in many algorithms to preprocess images for successive applications. This method considers a small window around each pixel in the given image and identifies all distinctive pixel windows[10]. Soleness is calculated by moving windows by a minute value in a particular direction and comparing the change that is detected in each of the pixel values. The sum squared difference(SSD)[11] is the sum of the squared difference of the values of each pixel score before and after the shift in all 8 directions. A change function for each image is defined as an aggregate of every sum squared differences(SSD).

Harris algorithm can be explicated in the following steps:

1. Transformation of the given image to a grayscale image

2. Application of a Gaussian filter to remove any noise or distortions.

3. Application of Sobel operator to obtain the x and y gradient values for each pixel in the grayscale image.

4. A 3×3 window is considered around every pixel in the grayscale image and the corner strength function is computed. This value is called the Harris value of the image.

5. all pixels that exceed a certain threshold are detected and is the local maxima within the window

6. For every pixel that meets the criteria of the threshold, a feature descriptor is computed[12].

8. KAZE

The KAZE method utilizes the non-linear scale-space through non-sequential diffusion filtering for feature point extraction and was put forward by P. F. Alcantarilla et al in 2012. KAZE method is built on scale normalized determinant of Hessian Matrix that is calculated at multiple scale levels[13]. The highest point of detector output is recorded as an interesting point by a mobile window. In the Feature description, the feature of orientation equability is introduced by detecting prominent orientation in the vicinity of a given radius around each of the recorded feature. Although this method requires high computational resources KAZE features are equable to rotation, scale, limited affine and are more distinctive at various scales. The equation representing the standard nonlinear diffusion can be expressed as.

$$\frac{\partial L}{\partial t} = div(c(x,y,t).\nabla L) \qquad \text{------- eqn[2]}$$

Where c = conductivity function

div = divergence,

∇ = gradient operator

L = luminance of the image.

III. EXPERIMENTS

This paper focuses on various feature detection algorithms. The implementation was done on Intel® core(TM) i7 10th gen processor with 16GB RAM and speed of 2.6-5GHz. The code was written in MATLAB R2020b on Windows 10 64 bits. The experiment consists of various tests by introducing effects like rotation, scale change and noise. The sample picture considered for all the tests is shown below having a size of 2.2Mb.

The experiment is divided into two sections.

The first section tabulates the time required for feature extraction of each point feature extraction methods namely BRISK, FAST, HARRIS, KAZE, MINEIGEN, MSER, ORB, SURF.

ALGORITHM	TIME OF FEATURE EXTRACTION
BRISK	0.313587
FAST	0.074147
HARRIS	0.0853465
KAZE	4.763558
MINEIGEN	0.91857
MSER	1.010411
ORB	0.148321
SURF	0.334866

Table 1. Time of feature extraction for each algorithm

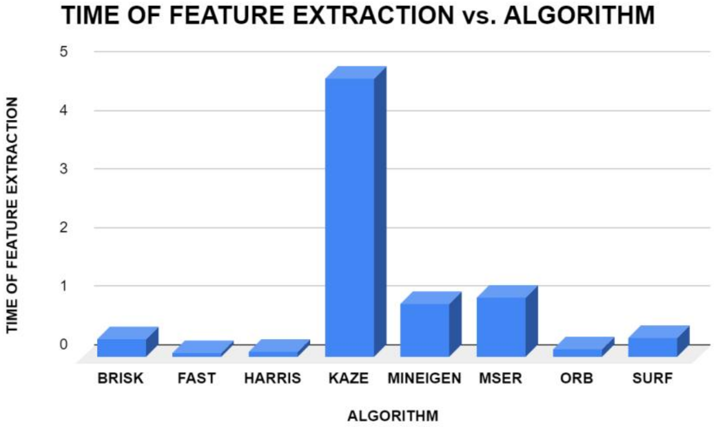

Figure 2. Graph for Feature Extraction time vs Algorithm

The graphs are plotted between different types of feature extraction algorithms and the time taken by these algorithms to extract the feature points from a given reference image. It is evident from the graph that the highest time for feature extraction was for KAZE and the lowest time was recorded for the FAST feature extraction method. Although KAZE required a higher computation time, the number of matched points was comparatively higher for KAZE, emphasising the linear relation between computation time and the number of matched points. From the above visualisation, it can be observed that the SURF feature extraction method requires comparatively less computation time in all variants of inputs and provides a relatively higher number of match points in most cases. Thus it can be concluded that SURF is befitting in a situation where the raw inputs are provided.

The second section is sub categorised into three parts.

The first part determines the time for evaluation of matching points and the number of matched points for images having angular distortion. For the experiment distortion angles of 0, 45, 90 degrees have been considered.

TILT - FEATURE EXTRACTION METHOD VS TIME			
ALGORITHM/ TILT ANGLE	45* TILT	90* TILT	0* TILT
BRISK	0.057429	0.046397	0.048213
FAST	0.064922	0.060837	0.067189
HARRIS	0.057784	0.055006	0.063457
KAZE	0.02712	0.027696	0.026556
MINEIGEN	0.053279	0.050198	0.05072
MSER	0.029002	0.031249	0.029311
ORB	0.06573	0.064043	0.05838
SURF	0.039339	0.03423	0.044163

Table 2. TILT - Feature Extraction Method and time taken

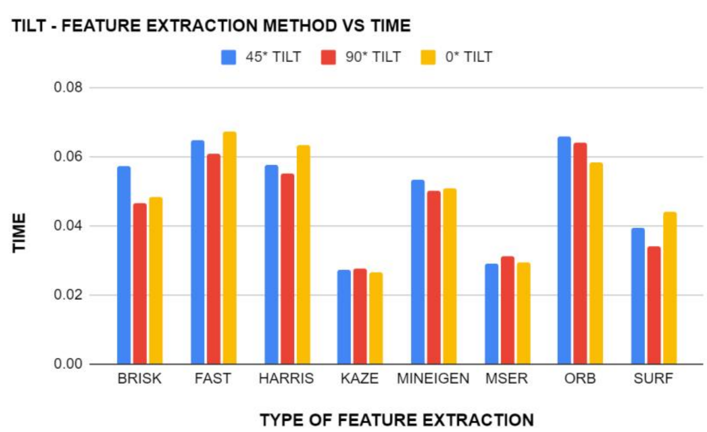

Figure 3. TILT - Feature Extraction Method VS Time

The above graph is plotted between the different feature extraction algorithms and the time taken by these algorithms to cross-match the interesting point in the reference image to the points in the variants of the reference image with different tilts of 0°,45°,90°. As visible from the graph, the KAZE algorithm was able to cross-match all three cases of tilt with the reference image within a time range of 0.026 to 0.027 seconds. It can also be observed that the time taken by KAZE to cross-match is almost consistent for all the three types of tilt image inputs whereas methods like ORB show large variation time taken for different tilts. Moreover, the highest time of computation for the cross-matching process was observed in ORB and Fast methods making these methods less suitable for scenarios that require high computational efficiency and quick outputs.

TILT - FEATURE EXTRACTION METHOD VS MATCHED POINT			
Algorithm	**45* Tilt**	**90* Tilt**	**0* Tilt**
BRISK	**35**	**47**	**496**
FAST	**25**	**118**	**483**
HARRIS	**3**	**31**	**498**
KAZE	**44**	**41**	**500**
MinEigen	**3**	**46**	**500**
MSER	**140**	**193**	**463**
ORB	**41**	**41**	**500**
SURF	**142**	**327**	**500**

Table 3. TILT - Feature Extraction Method and number of matched points

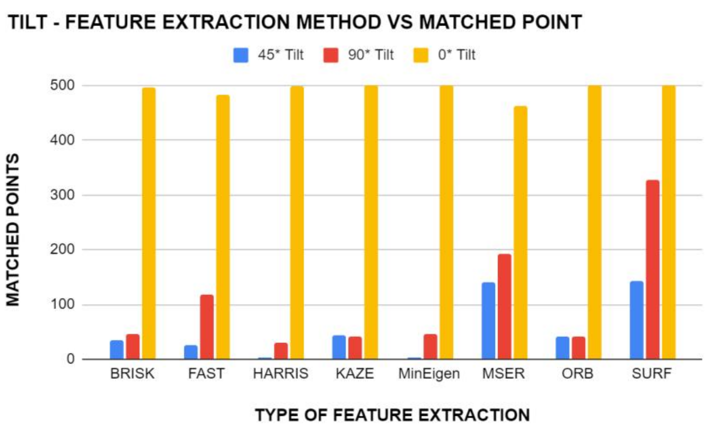

Figure 4. TILT - Feature Extraction Method vs Matched Points

The above graph is plotted between the different feature extraction algorithms and the number of matched points. As evident from the graph, the number of match points is maximum at 0° tilt for all the different types of feature extraction. Comparing the above two graphs of tilt input scenario it can be concluded that ORB requires higher computational infrastructure compared to methods like KAZE and MSER as the latter method was able to extract approximately the same number of interest points within one by the third fraction of time consumed by ORB and FAST methods. Also, contrary to the prior traits SURF has cross-matched the maximum number of points in all three type of tilt inputs.

The second part determines the time for evaluation of matching points and the number of matched points for images having noises. For the experiment, common noises like median, Gaussian and Dust and scratches are considered.

NOISE - FEATURE EXTRACTION METHOD VS TIME			
ALGORITHM	**MEDIAN NOISE**	**Gauss Noise**	**Dust and scratch**
BRISK	**0.048426**	**0.044872**	**0.041388**
FAST	**0.047351**	**0.041478**	**0.04614**
HARRIS	**0.054686**	**0.05249**	**0.05528**
KAZE	**0.025917**	**0.026505**	**0.027484**
MINEIGEN	**0.054124**	**0.051961**	**0.050279**
MSER	**0.030573**	**0.035674**	**0.037759**
ORB	**0.060046**	**0.065243**	**0.066454**
SURF	**0.037223**	**0.033108**	**0.038761**

Table 4. NOISE - Feature Extraction Method and time taken

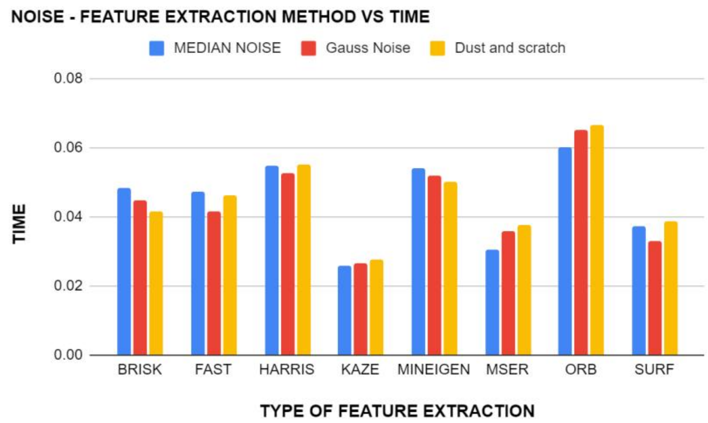

Figure 5. NOISE - Feature Extraction Method VS Time

The graph has been plotted for noise-feature extraction of three different types of noise against the time taken for these extractions by each method. From the graph, we can 0bserve that KAZE took the least time, between 0.0259sec (for median noise) and 0.0274sec (for Dust and Scratch) for feature extraction for all three types of noise, followed by MSER and SURF. The time taken by KAZE is also consistent when compared to the other methods where the time taken differs from each other by a greater value. We can also observe that ORB is the method that took the longest for extracting the features for all three types of noises and thus making it the least suitable among the bunch if the images to be cross-matched have a significant amount of noise.

NOISE - FEATURE EXTRACTION METHOD VS MATCHED POINT			
Algorithm	**Median Noise**	**Gauss Noise**	**Dust and scratch**
BRISK	**15**	**103**	**13**
FAST	**0**	**20**	**1**
HARRIS	**4**	**9**	**2**
KAZE	**28**	**162**	**49**
MinEigen	**2**	**3**	**3**
MSER	**18**	**12**	**14**
ORB	**34**	**8**	**34**
SURF	**94**	**259**	**94**

Table 5. NOISE - Feature Extraction Method and number of matched points

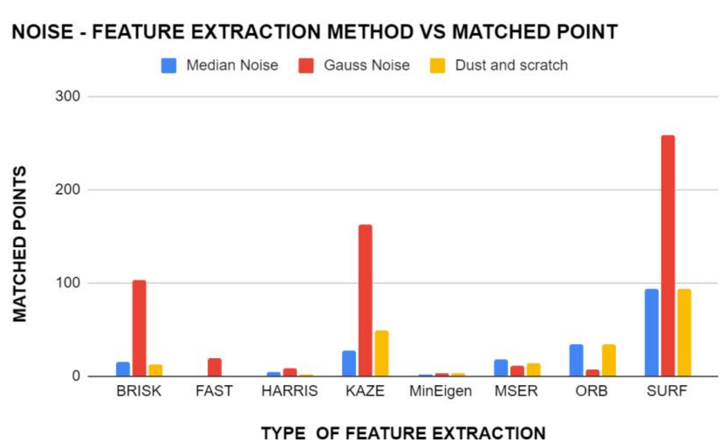

Figure 6. NOISE - Feature Extraction Method vs matched points

The graph has been plotted for noise-feature extraction of three different types of noise against the number of matched points they were able to come up with for the different method. It can be inferred from the graph that SURF has the highest number of cross-matched points with Gaussian noise having the highest number among the different noises. KAZE and BRISK are the methods that have some comparable stats and all the other methods falling behind them by a large margin. So when it comes to extracting the feature points from the images with noise, SURF is the best. If we are to combine time taken and the number of points they were able to come up with in that period, SURF is the best available choice for cases where noise is there.

The third part determines the time for evaluation of matching points and the number of matched points for images having scaled distortion. For the experiment distortion angles of 120, 150, 170, 190 percentages of the reference image has been considered.

SCALED - FEATURE EXTRACTION METHOD VS TIME				
ALGORITHM/SCALED FACTOR	120% SCALE	150% scale	170% scale	190% scale
BRISK	0.0439	0.04487	0.049317	0.042299
FAST	0.046347	0.043665	0.047194	0.047015
HARRIS	0.059786	0.052826	0.053238	0.053117
KAZE	0.026761	0.028547	0.026327	0.025562
MinEigen	0.051124	0.049958	0.052885	0.051355
MSER	0.042234	0.041882	0.030799	0.030086
ORB	0.062399	0.063155	0.070053	0.064498
SURF	0.037279	0.034314	0.038728	0.035153

Table 6. SCALE - Feature Extraction Method and time taken

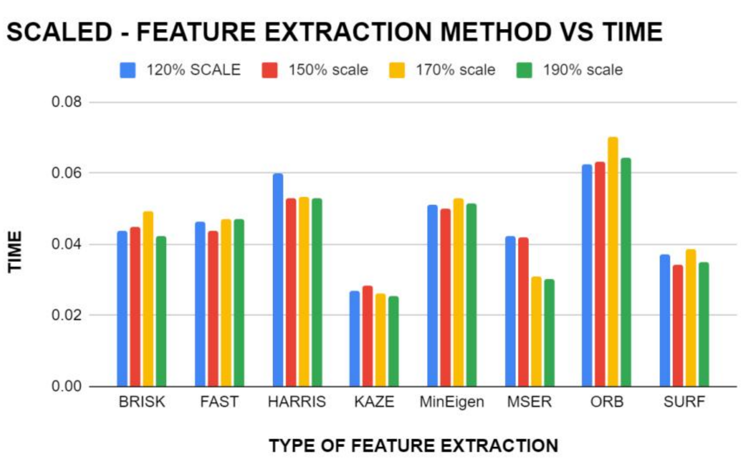

*Figure 7.*SCALE - Feature Extraction Method and time taken

The above graph is plotted between different feature extraction algorithms and the time taken by these algorithms for feature extraction. The scaled variants of the Preference image was provided as the input for plotting the graphs. It is evident from the graph that the KAZE method has the least computation time ranging from 0.026 to 0.028 whereas the ORB method requires the maximum computation time ranging from 0.062 to 0.070. It is also noteworthy that the SURF method maintained its trait by requiring a computation time slightly above the method with the least computation time.

SCALED - FEATURE EXTRACTION METHOD VS MATCHED POINT				
Algorithm	120% scale	150% scale	170% scale	190% scale
BRISK	35	23	19	7
FAST	45	50	14	6
HARRIS	25	8	2	0
KAZE	76	23	7	13
MinEigen	28	5	4	4
MSER	67	29	41	28
ORB	61	42	71	25
SURF	200	228	177	172

*Table 7.*SCALE - feature extraction method vs matched point

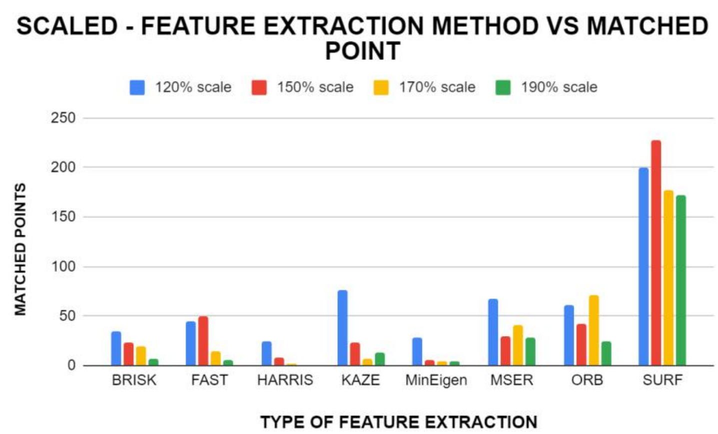

*Figure 8.*SCALE - feature extraction method vs matched point

The above figure shows the graph between different feature extraction algorithms and the maximum number of matched points of these algorithms. Scaled image variants of the reference image were provided as the input in the above graph and the results were as expected with SURF uncovering the maximum number of match points. Although the KAZE method required the least amount of computation time (0.026-0.028) it was only able to locate relatively very few matchpoints.

As evident from the above two graphs, SURF was able to locate the maximum number of match points, within a comparatively lesser amount of time. Thus it can be concluded that for scaled input variants SURF has the maximum computational efficiency.

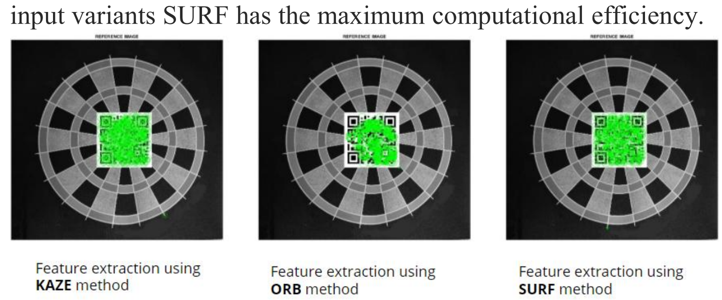

*Figure 9.*KAZE, ORB, SURF reference image with feature extraction implemented

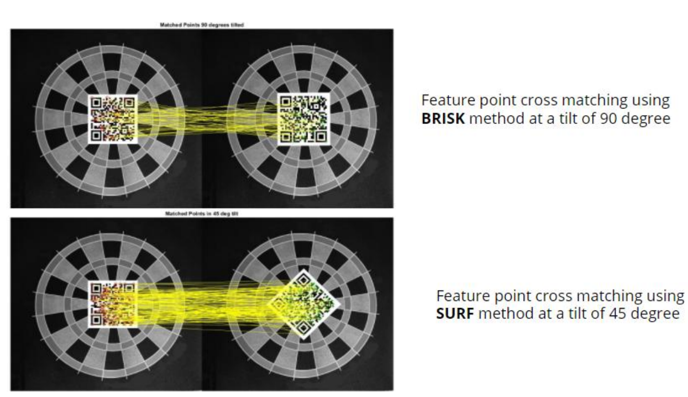

*Figure 10.*BRISK, SURF cross matching examples

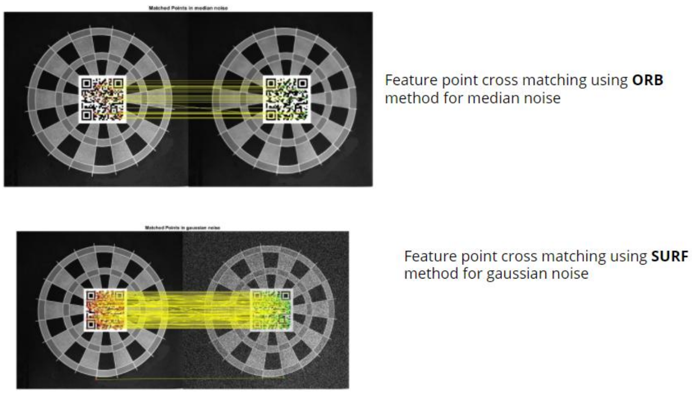

Figure 11.ORB and SURF cross matching examples

IV. CONCLUSION

In summary, the SURF algorithm performs the best in cases of noise, rotation and scaling with maximum feature points extracted in a relatively minimal time. Brisk is the method that performed moderately well in all of the test scenarios. FAST has the least time for feature extraction and KAZE has the most time among them. ORB performs well in scaled case but at the cost of time.

In the case of noise and scaled condition, SURF outperforms every other method with the highest number of feature points matched with the best time to points matched ratio. Even though KAZE has the least time taken, the number of points matched is way less.

In the case rotation, SURF and MSER performs the best among them with a maximum number of extracted points matched in a relatively less time frame. All the features matched all the points when presented with ZERO degrees tilt.

V. REFERENCES

1. BRISK: Binary Robust Invariant Scalable Keypoints Stefan Leutenegger, Margarita Chilli and Roland Y. Siegwart Autonomous Systems Lab, ETH Zurich
2. Leutenegger, Stefan, M.Chli, and R.Siegwart(2011), “BRISK: Binary robust invariant scalable keypoints.”, In Proceedings of “IEEE International Conference on Computer Vision (ICCV)”, pp. 2548-2555.
3. Wu, M. Research on optimization of image fast feature point matching algorithm. *J Image Video Proc.* 2018, 106 (2018). https://doi.org/10.1186/s13640-018-0354-y
4. Zeng, Yonghong & Liang, Ying-Chang. (2007). Maximum-Minimum Eigenvalue Detection for Cognitive Radio. 1 - 5. 10.1109/PIMRC.2007.4394211.

5. P.-E. Forssen(2007), “Maximally stable colour regions for recognition and matching,” In Proceedings of “IEEE Conference on Computer Vision and Pattern Recognition”, Minneapolis, USA.
6. Donoser, Michael & Bischof, Horst. (2006). Efficient Maximally Stable Extremal Region (MSER) Tracking.IEEE Comp. Soc. Conf. Comp. Vis. Pattern Recogn. 1.553-560.10.1109/CVPR.2006.107.
7. ORB: an efficient alternative to SIFT or SURF, Ethan Rublee Vincent Rabaud Kurt Konolige Gary Bradski
8. Speeded-Up Robust Feature Extraction and Matching for Fingerprint Recognition, Umma Hany; Lutfa Akter; 2nd Int'l Conf. on Electrical Engineering and Information & Communication Technology (ICEEICT) 2015 Jahangirnagar University, Dhaka-1342, Bangladesh, 21-23 May 2015
9. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool(2008), “SURF: Speeded Up Robust Features, “Computer Vision and Image Understanding, vol. 110(3), pp. 346–359.
10. Harris and M. Stephens (1988), “A combined corner and edge detector”, (PDF) , In: Proceedings of Alvey Vision Conference, pp. 147–151.
11. Evaluating Harris Method in Camera Calibration OUYANG Cheng, WANG Guangzhi, ZHANG Quan, KANG Wei, DING Hui, Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference Shanghai, China, September 1-4, 2005
12. Kok, Kai Yit & Rajendran, Parvathy. (2018). Validation of Harris Detector and Eigen Features Detector. IOP Conference Series Materials Science and Engineering. 370. 10.1088/1757-899X/370/1/012013.
13. Alcantarilla P.F., Bartoli A., Davison A.J. (2012) KAZE Features. In: Fitzgibbon A., Lazebnik S., Perona P., Sato Y., Schmid C. (eds) Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7577. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33783-3_16
14. Comparison and Study of Classic Feature Point Detection Algorithm Daguang Jiang, Junkai Yi 2012 International Conference on Computer Science and Service System
15. Performance Evaluation of Various Feature detection Algorithms in VSLAM, S.SriVidhya, Prakash S, ISSN - 2250-1991 | IF 5.215 | IC Value: 79.96, PARIPEX - INDIAN JOURNAL OF RESEARCH
16. J. Bauer, H. Bischof, A. Klaus, K. Karner(2004), “Robust and fully automated image registration using invariant features,” ISPRS, DVD Proc, pp. 12–23.