International Conference of Sciences, Engineering & Environmental Technology (ICONSEET), 5(20): 157– 163, 2019
ISSN 0794-9650
www.repcomseet.com

EVALUATION AND COMPARISON OF IMAGE MATCHING ALGORITHMS IN PROCESSING DIGITAL IMAGES ACQUIRED WITH MOBILE DEVICE.

Amoo Nureni Babatunde[1*], Odeyemi Felix Gbenga[1],Suru Whenayon Peter[1], Elufisan Paul Tomilola[2],

[1]Department of Surveying and Geo-informatics, The Federal Polytechnic Ede, Osun State, Nigeria.
[2]Department of Surveying and Geo-informatics, Federal University of Technology Akure, Ondo State, Nigeria.

[*]Corresponding author: olojaone78@yahoo.com; Tel: +2348069155169

Abstract: In this paper, we evaluate and compare the performance of three different image matching algorithms, which are, SURF, FAST and BRISK on images acquired with a mobile device. Based on the experimental results shown in this paper, quantitative comparison of each algorithm was done base on the number of points or features detected, number of features matched, total image matching time, efficiency and repeatability. It was shown that SURF is the most efficient algorithm with 68% efficiency and in the case of timings; FAST algorithm is faster in detection and matching of features while BRISK algorithm has the ability of detecting more features from images.

Keyword: **Algorithms, BRISK, FAST, Image matching, SURF**

1.0 INTRODUCTION

Nowadays more and more image acquisition systems based on digital sensors are developed and available at reasonable prices in the market. Mobile digital devices, such as mobile phones and tablet PCs, extend the digital cameras functionalities, allowing real-time exchange of images through their communication modems. The main advantage of this camera system is the possibility to acquire digital images and directly process them on a computer. They are used for close range photogrammetric applications. Extracting useful information from all these data, through efficient computational algorithms, is an important goal of the Computer Vision research community.

This research work is based on the evaluation and comparison of image matching Algorithms in processing digital images acquired with Mobile devices. In achieving this, SURF, FAST and BRISK were adopted for the matching of digital images acquired with Mobile devices and each of these algorithms was evaluated based on their results.

Each of these image matching algorithms adopted for this work have different methods, procedures and stages by which are they used in image matching and these different methods used by these were adopted for this work. Therefore, we present the evaluation of these algorithms with respect to each other which are widely accepted as a standard of comparison under common image transformations and this is done to see which of the algorithm the best is.

The programming and image processing aspect of this research work was achieved with the aid of MATLAB. MATLAB is a general-purpose and a vast programming language for computer vision and image processing. To this end, the language expresses a sense of balance between expressiveness, and performance.

2.0 RESEARCH METHODS

Workflow of the Research

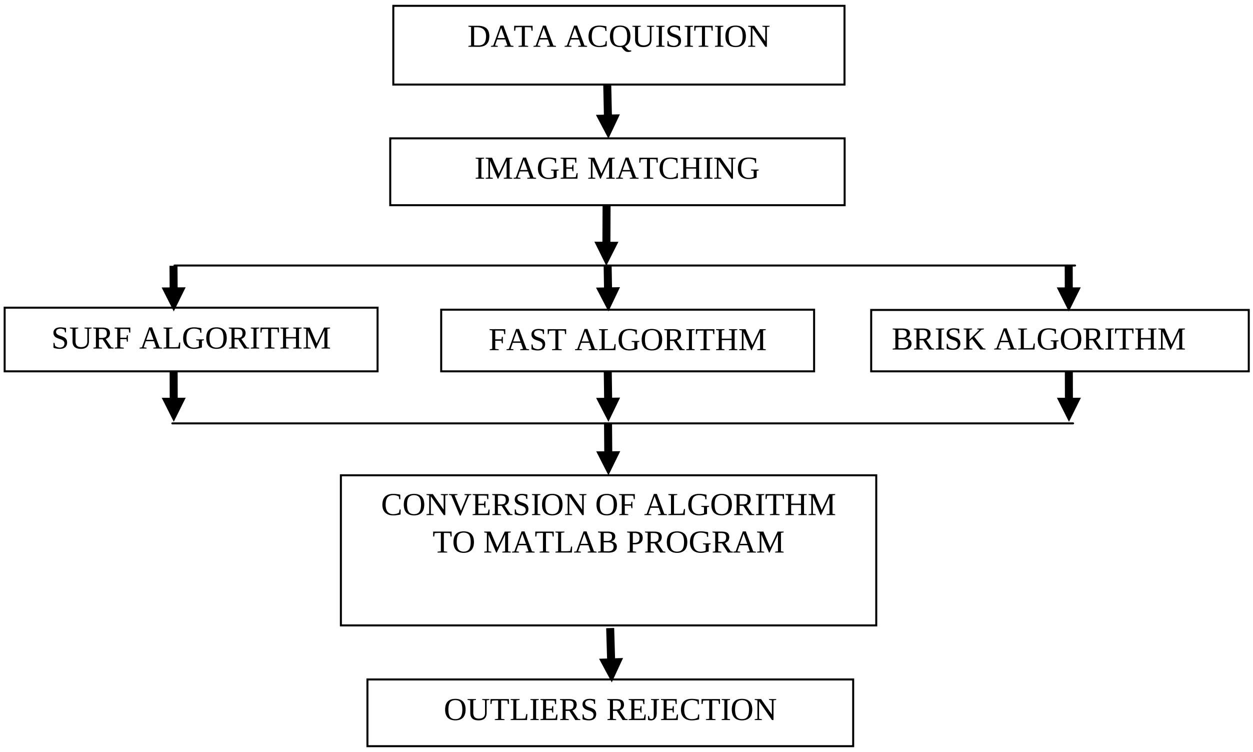


Figure 1.0: Workflow of the Research

As earlier said that the digital images were matched using three different image matching algorithms and the methods used by these algorithms in image matching are discussed below:

2.1 SPEED UP ROBUST FEATURES (SURF) ALGORITHM

SURF approximates the DoG with box filters. Instead of Gaussian averaging the image, squares are used for approximation since the convolution with square is much faster if the integral image is used. Also this can be done in parallel for different scales. The SURF uses a BLOB detector which is based on the Hessian matrix to find the points of interest. For orientation assignment, it uses wavelet responses in both horizontal and vertical directions by applying adequate Gaussian weights. For feature description also SURF uses the wavelet responses. A neighborhood around the key point is selected and divided into sub regions and then for each sub region the wavelet responses are taken and represented to get SURF feature descriptor. The sign of Laplacian which is already computed in the detection is used for underlying interest points. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse case. In case of matching the features are compared only if they have same type of contrast (based on sign) which allows faster matching.

2.2 FEATURES FROM ACCELERATED SEGMENT TEST (FAST) ALGORITHM

FAST is an algorithm proposed originally by Rosten and Drummond for identifying interest points in an image. An interest point in an image is a pixel which has a well-defined position and can be robustly detected. Interest points have high local information content and they should be ideally repeatable between different images. Interest point detection has applications in image matching, object recognition, tracking etc.

The reason behind the work of the FAST algorithm was to develop an interest point detector for use in real time frame rate applications like SLAM on a mobile robot, which have limited computational resources. Detection of multiple interest points adjacent to one another is one of the other problems of the initial version of the algorithm. This can be dealt with by applying non maximal suppression after detecting the interest points.

2.3 BINARY ROBUST INVARIANT SCALABLE KEYPOINTS (BRISK) ALGORITHM

BRISK is an image matching algorithm for high quality, fast keypoint detection, description and matching. The method is rotation as well as scale invariant to a significant extent, achieving performance comparable to the state-of-the-art while dramatically reducing computational cost. It achieves comparable quality of matching at much less computation time.
The key stages in BRISK are namely feature detection, descriptor composition and keypoint matching. They are described as follows:

2.3.1 ScaleSpace Keypoint Detection

It is performed at the first moment. The goal at this step is to identify salient points in the image that, ideally, could be uniquely differentiated from any other point. To do so, these points must be searched across the image and scale dimensions using a saliency criterion (Leuteneggeretal, 2011). The search through scale space is fundamental to achieve scale invariance and it is realized in BRISK by the use of a pyramid of images. This pyramid is formed by many layers which correspond to resample of the original image.

2.3.2 Keypoint Description

Given a set of keypoints (consisting of sub-pixel refined image locations and associated floating-point scale values), the BRISK descriptor is composed as a binary string by concatenating the results of simple brightness comparison tests.
This idea has been demonstrated by (Calonder et al, 2010) to be very efficient, however here we employ it in a far more qualitative manner. In BRISK, we identify the characteristic direction of each keypoint to allow for orientation-normalized descriptors and hence achieve rotation invariance which is key to general robustness. Also, we carefully select the brightness comparisons with the focus on maximizing descriptiveness.

2.3.3 Descriptor Matching

Finally, to perform the comparison between two or more keypoint descriptors, the Hamming distance is used. This distance measures the number of different bits between two binary strings and it represents the degree of inequality of the descriptors being compared.

3.0 EXPERIMENTS AND RESULTS

The method of image matching for each image matching algorithm were implemented and programmed using programming language called MATLAB 2017a. Specifications of the computer system used are: ***Intel(R) Core(TM) i3-4670 CPU @ 2.20 GHz,* and *4.00 GB RAM with windows 8.1 as an operating system.*** The dataset contains a sequence of two images.

(a) Image 1

(b) Image 2

Figure 2.0: Image datasets used for the experiment

We therefore evaluate the joint performance of all these stages in BRISK and compared it to SURF and FAST. Based on the experimental results shown in the table of results below, quantitative comparison of each algorithm was done base on the number of points or features detected, number of features matched, total image matching time, efficiency and repeatability.

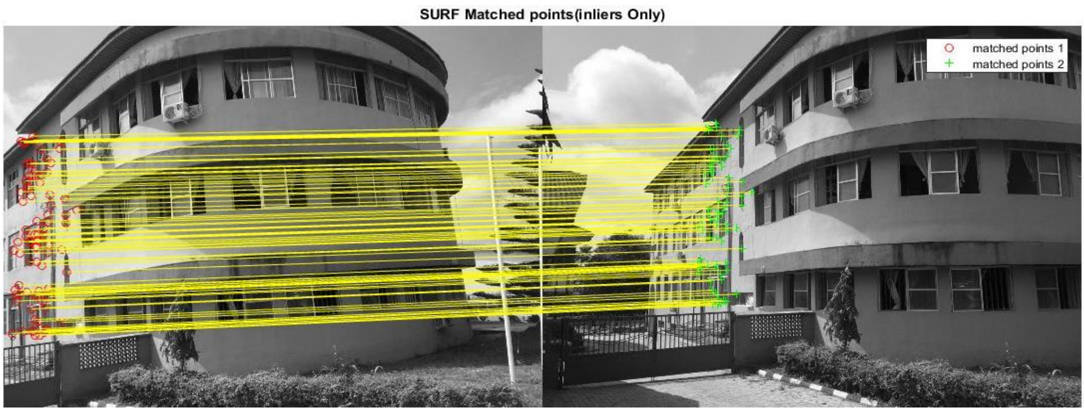
Table1: Shows The Results of Quantitative Comparison for Each Image Matching Algorithm.

Algorithm	Features Detected in Image Pairs		Features Matched	Inliers	Outliers	Mean Image Matching Time(seconds)	Repeatability (%)	Efficiency (%)
	Image 1	Image 2						
SURF	1660	1665	487	83	404	0.00171	20.54	58
FAST	2274	2051	115	98	17	0.00144	576.47	11
BRISK	3700	3405	188	51	137	0.00176	37.23	11

In table 1, it is observed that **1660** features are detected in image1 and **1665** features are detected in image2 using SURF algorithm. Also, **2274** features are detected in image1 and **2051** features are detected in image2 using FAST algorithm. While using BRISK algorithm it is also observed that **3700** features are detected in image1 and **3405** features in image2.

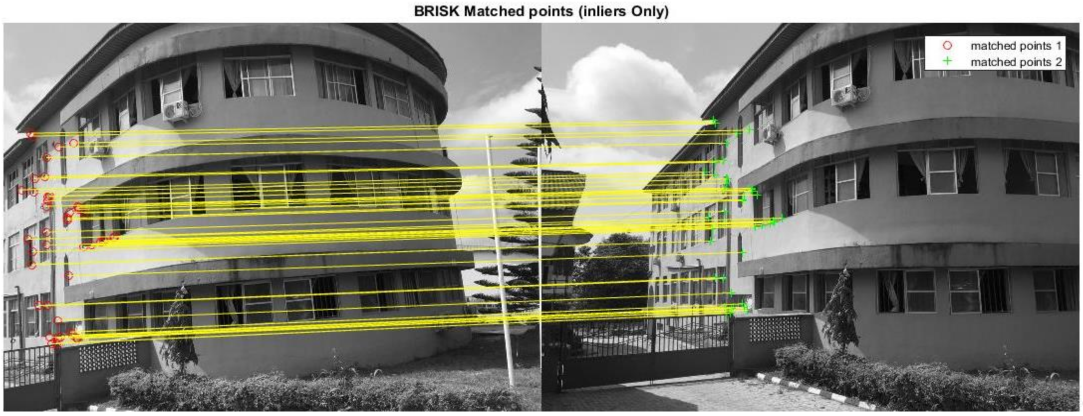
(a) (b) (c)

Figure 3.0: Feature detection of image 1 using: (a) SURF (b) FAST (c) BRISK



(a)

(b)



(c)

Figure 4.0: Matching of image 1 with image 2 using: (a) SURF (b) FAST (c) BRISK

We can see that BRISK algorithm detected the highest number of features in both images. SURF algorithm is the runner-up as it detects more number of features than FAST algorithm. SURF features are detected in a scattered form generally all over the image and SURF features are denser. BRISK features are more concentrated on corners.

The feature matching strategy used here is Normalized Cross Correlation technique which was used in matching SURF, FAST and BRISK features. From the table above, it was observed that SURF algorithm have the highest number of matched features from both images. This indicates that SURF algorithm has the ability of matching more corresponding features or points detected in two or more images than any other image matching algorithm. BRISK algorithm is the runner-up as it was able to match **188** features from both images more than FAST algorithm. FAST algorithm was able to match **115** features from the images.

3.1 Outlier Rejection

Outliers are those errors that do not show a systematic character and do not fit to the considered error distribution. A statistically robust M-estimator Sample Consensus (MSAC) algorithm, which is a variant of the Random Sample Concensus (RANSAC) algorithm, was used to find a transformation corresponding to the matching point pairs. (MSAC) algorithm with **500 iterations** and **99.5% confidence** has been applied to removes outliers while computing the transformation matrix. You may see varying results of the transformation computation because of the random sampling employed by the MSAC algorithm. Table 1 above shows the number of outliers rejected and inliers accepted as given by MATLAB.

3.2 Timings

Tests were performed to measure the relation between execution time in the steps of detection, description and matching of the both algorithm. The time was recorded with the aid of a stopwatch and the result in time is shown in table 1 above. Each timing value presented in the table is the average of **50 measurements** (to minimize errors which arise because of processing glitches).

3.3 Efficiency

Based on the experimental result, the efficiency of each algorithm was evaluated and it is calculated using a relation given by:

$$\% \, Efficiency = \frac{2 \times NoofMatches}{\text{Total number of features } detected}$$

The efficiency of each algorithm was calculated and the result is shown in table1 above. The results showed that the number of matches is significantly lesser than the number of features detected. Therefore, it establishes that the number of matches is by themselves not very good. What is needed is to evaluate and compare that how much of features that are detected by the algorithm are actually useful in the process of matching then to their correspondents. So, the most efficient algorithm based on the experimental result is SURF with 68% efficiency.

3.3 Repeatability

The detector repeatability score as defined by (Mikolajczyk et al, 2005) is calculated as the ratio between the corresponding keypoints and the minimum total number of keypoints visible in both images. The correspondences are identified by looking at the overlap area of the keypoint region in one image (i.e. the extracted circle) and the projection of the keypoint region from the other image (i.e. ellipse-like): if the region of intersection is larger than 50% of the union of the two regions, it is considered a correspondence. Note that this method is largely dependent on the assignment of the keypoint circle radius, i.e. the constant factor between scale and radius. We choose this such that the average radii obtained with the BRISK detector approximately match the average radii obtained with the SURF and FAST detectors.
The assessment of repeatability scores is performed using constant BRISK detection thresholds across one sequence. For the sake of a fair comparison with the SURF detector, we will adapt the respective Hessian threshold such that it outputs approximately the same number of correspondences in the similarity based matching setup. From Figure 5 below, the degree of viewpoint changes, increases gradually from the image 1 to image 2 in each image set. The image 1 is considered as the original image in both sets.

3.4 Recall and Precision Evaluation

Recall and precision are two relevant metrics used to show the effectiveness of a descriptor method. Recall measures the proportion of correct positives matches, considering all the possible correct positive matches between two images. While precision measures the proportion of false positives matches, considering all the performed matches between those images (Alexander et al, 2013). Mathematically, **Recall** =

$$\frac{True\ Positive}{True\ Positive + False\ Negative}$$

Precision = $\frac{True\ Positive}{True\ Positive + False\ Positive}$

These curves in figure 4 represent the algorithm capacity to keep a small number of false positives (precision) while it associates the maximum number of keypoints between two images (recall).

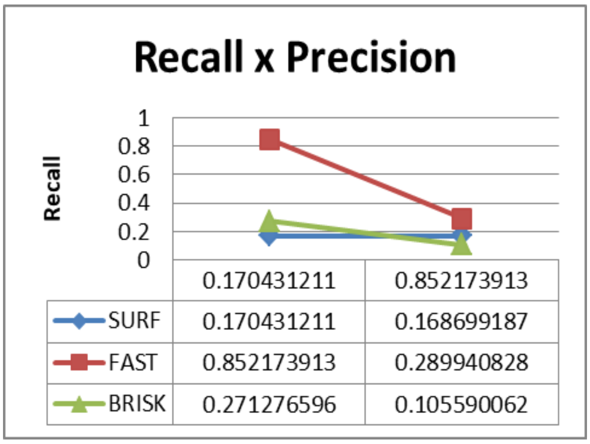


Figure 5.0: Recall and Precision line

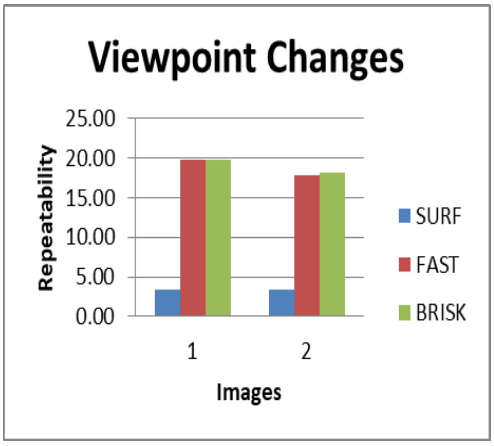


Figure 6.0: Repeatability test for images

4.0 CONCLUSION AND RECOMMENDATION

4.1 Conclusion

This paper evaluates and compares the image matching Algorithms (SURF, FAST and BRISK) in processing digital images from Mobile devices. Experiment was carried out based on various methods of each image matching algorithm using the digital images acquired from mobile phone and the result was gotten with the aid of a MATLAB**.** From the results, we can see that BRISK algorithm has the ability of detecting more features from images and BRISK features are more concentrated on corners. Also, SURF features are detected in a scattered form generally all over the image and SURF features are denser. In the aspect of matching features, SURF algorithm has the ability of matching more corresponding features or points detected in two or more images than any other image matching algorithm.

Furthermore, based on the experimental result, I hereby conclude that SURF algorithm is the most efficient algorithm with 68% efficiency. But in the case of timings, result showed that a FAST algorithm is faster in detection and matching of features. Its detection and descriptor computation is typically an order of magnitude faster than the one of SURF and BRISK, which is considered to be the fastest rotation and scale invariant features currently available. BRISK is easily scalable for faster execution by reducing the number of sampling-points in the pattern at some expense of matching quality – which might be affordable in a particular application. Moreover, scale and/or rotation invariance can be omitted trivially, increasing the speed as well as the matching quality in applications where they are not needed.

4.2 Recommendation

For future work, we hereby recommend that more image matching algorithms together with the ones used in this paper using the same dataset, so evaluate and compare those algorithm based on number of feature detected and matched, efficiency, timing, etc.

REFERENCE

Alexander, G., Camila, T., Rodrigo, Bem, and Diana, Adamatti (2013). Performance Evaluation of BRISK Algorithm on Mobile Devices. *In Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 5-11.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust Independent Elementary Features. In Proceedings of the 11th European Conference on Computer Vision (ECCV 2010), pages 778-792.

Leutenegger,S.,Chli, M.,and Siegwart, R. (2011). BRISK: Binary Robust Invariant Scalable Keypoints. *In Proceedings of the 13ht International Conference on Computer Vision (ICCV 2011).*

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A. J., Matas, F., Schaffalitzky, T. Kadir, and Gool, L. (2005). A Comparison of Affine Region Detectors. *International Journal of Computer Vision* (IJCV), 65(1):43–72, 2005. 2, 5

Mikolajczyk, K. and Schmid, C. (2005). A performance Evaluation of Local Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10):1615 –1630.