

Development of Pose Estimation Algorithm for Quranic Arabic Word

Luqman Naim Mohd Esa, Malik Arman Morshidi, Syarah Munirah Mohd Zailani*

Department of Electrical and Computer Engineering, Kulliyyah of Engineering,
International Islamic University Malaysia, Malaysia

*Corresponding author, e-mail: e-mail: luqman.naim3@gmail.com¹, mmalik@iium.edu.my²,
syarazailani@gmail.com³

Abstract

The study carried out in this report proposes the best keypoint detection, description, and pose estimation algorithm combination for Quranic Arabic words. Oriented-FAST Rotated-BRIEF (ORB) and Accelerated-KAZE (AKAZE) are used as the keypoint detection and description algorithms while Random Sample Consensus (RANSAC) and Least Median Squares (LMEDS) are used to evaluate the homography for pose estimation algorithms. The algorithms are combined with each other to provide four different techniques to estimate the pose of Quranic Arabic words. The algorithms are tested on a limited dataset chosen from a phrase within the Quran. Performance of each algorithm is measured in real-time through inlier to keypoint ratio which determines pose accuracy.

Keywords: Pose estimation, ORB, AKAZE, RANSAC, LMEDS

Copyright © 2018 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Keypoint detection, description, and pose estimation are all sub-topics under a bigger subject called Computer Vision (CV). This subject deals with a computer's ability to see and understand objects in the real world through visual sensors like cameras. The two subtopics of keypoint detection and description enable a computer to understand an object's unique fingerprint while pose estimation allows the computer to gauge an object's position relative to it [1],[2].

Various studies have been done previously on these sub-topics for applications like algorithm comparisons, monocular visual odometry, and mobile augmented reality games [3]-[7]. Though, little study has been done with regards to Quranic Arabic words. There is no comprehensive reference sheet for keypoint detection, description, and pose estimation databases of Quranic Arabic words. This study aims to propose the best algorithm for it and become a stepping stone in the direction for future research in this area.

For the sub-topics of keypoint detection and description, the study utilizes the ORB and AKAZE algorithms since they provide decent performance along with minimal computational loads [8],[9]. This is ideal for real-time situations where speed and efficiency are needed. Furthermore, they are free from any patent protection claims since they are open source and widely available for public use. In the case of pose estimation, RANSAC and LMEDS algorithms are used because they are the most popular methods today [10]-[12]. These algorithms evaluate the homography of two similar objects at different viewing angles and estimate the pose of the objects relative to each other as illustrated in Figure 1 [13]. The remainder of this paper is organized as follows. Section 2 presents the research methodology where the experimental procedure is described. The experimentation and result of the studies are discussed in section 3. This paper is concluded in Section 4.

2. Research Method

The study uses three steps for algorithm development which are video data acquisition, keypoint detection, and description, and pose estimation as illustrated in Figure 2. Before starting the experiment, a test subject is defined for the algorithms to acquire and process data. The phrase "Bismillahi-rahmani-raheem" in the Quran is chosen for this study as shown in

Figure 3 [14]. It is one of the most common phrases in the Quran and also the most recognizable making it an ideal test subject. To note, since there are various writing styles for Quranic words, the study will only focus on text writings from Rasm Uthmani versions of the Quran. Rasm Uthmani is the most widely used version of the Quran today.

The phrase in Figure 3 contains four main elements which are physically independent of each other. The elements are "Bismi", "Allah", "Ar-Rahman", and "Ar-Raheem" as shown in Figures 4, 5, 6, and 7 respectively. The test subjects of this experiment are divided into these four basic elements.

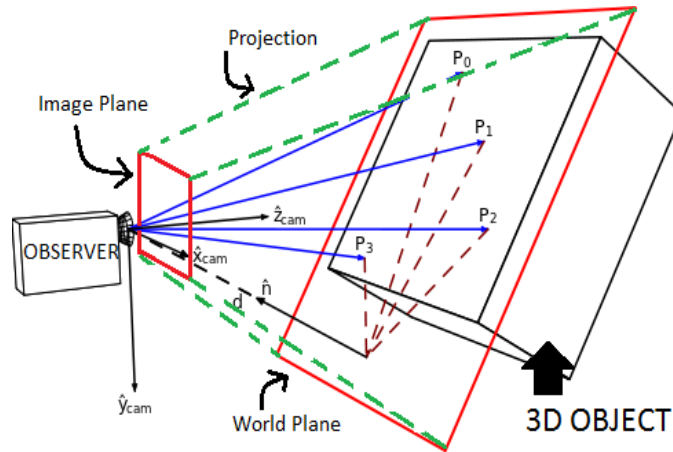


Figure 1. Homography evaluation for pose estimation [13]

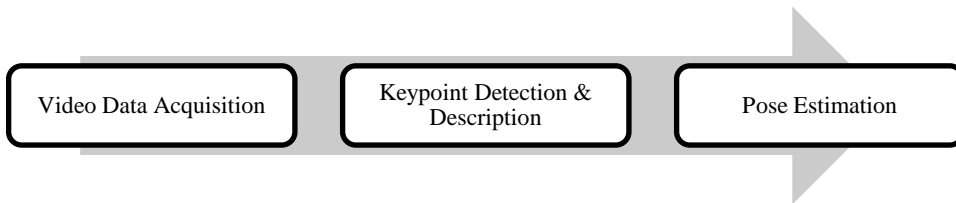


Figure 2. Experiment procedure



Figure 3. Rasm Uthmani version of "Bismillahi-rahmani-raheem" [14]

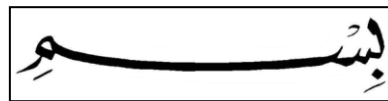


Figure 4. "Bismi"



Figure 5. "Allah"



Figure 6. "Ar-Rahman"



Figure 7. "Ar-Raheem"

3. Results and Analysis

The experimentation and results are discussed in detail following the steps of the algorithm; video data acquisition, keypoint detection and description, and pose estimation.

3.1. Video Data Acquisition

Although the study is intended for real-time video input, a pre-recorded video will instead be used to study algorithms. This is because, with a real-time video, the location variables of each keypoint differ from one real-time session to another. Hence, results will not be consistent with each real-time test iteration. With a pre-recorded video, each keypoint will be at the exact same location for every test iteration and this will produce dependable results for each test.

To record the video, an LG Nexus 5X smartphone is used with a 13-megapixel camera and autofocus enabled. The resolution of the video is 1920x1080 pixels at the time of recording with an aspect ratio of 16:9. Though, due to the large size of the pre-recorded video, a compression software is used to reduce the video resolution to 640x480 pixels which enable faster processing when it is run through the algorithm comparison software. Video length is exactly 1-minute with a rate of 23 frames per second. The video combines a series of camera movements comprising of zooming, tilting, rotating, and random motions around the four test subjects as shown in Figures 8 to 11.



Figure 8. Zooming

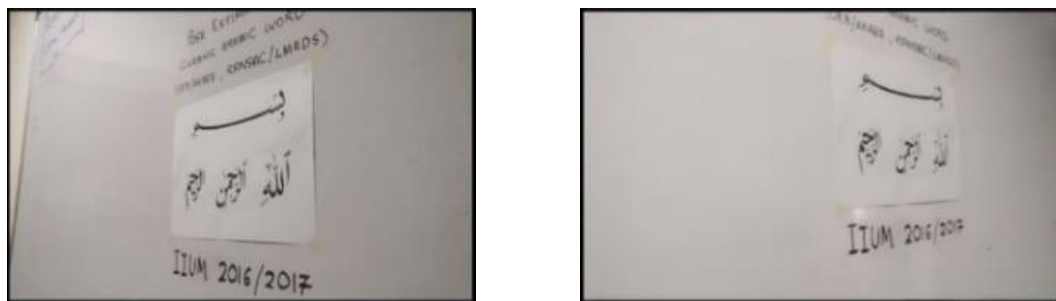


Figure 9. Tilting

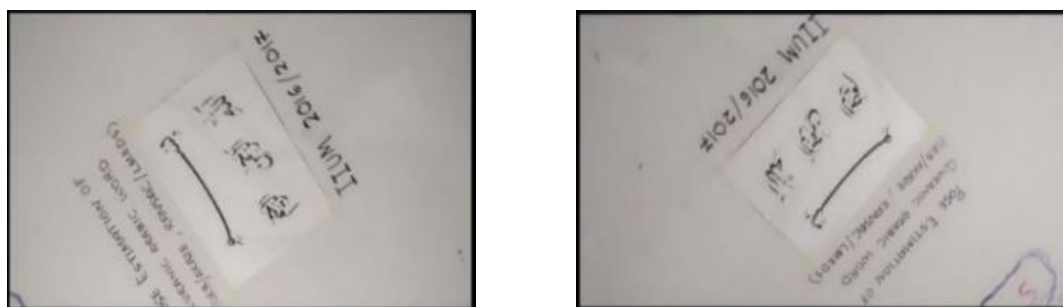


Figure 10. Rotating

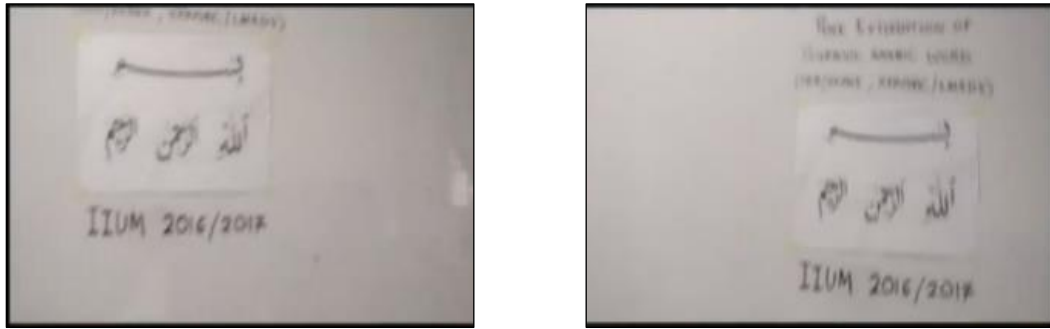


Figure 11. Random motions

The video is loaded into a compiled software program taken from [15] that compares the performance of four different algorithm combinations which are AKAZE+RANSAC, ORB+RANSAC, AKAZE+LMEDS, and ORB+LMEDS as shown in Figures 12 to 15. The software is compiled using Microsoft Visual Studio 2017 and OpenCV v3.2.0. In each comparison, the number of matches, inliers, and inlier ratio are updated frame-by-frame in real-time and logged into a text file. These three parameters are used as the evaluation metrics for performance analysis of each algorithm.

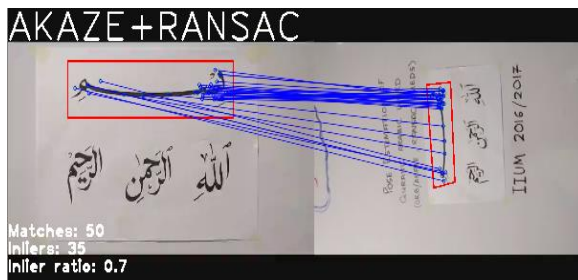


Figure 12. AKAZE+RANSAC

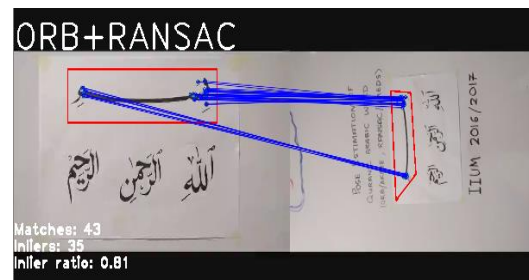


Figure 13. ORB+RANSAC

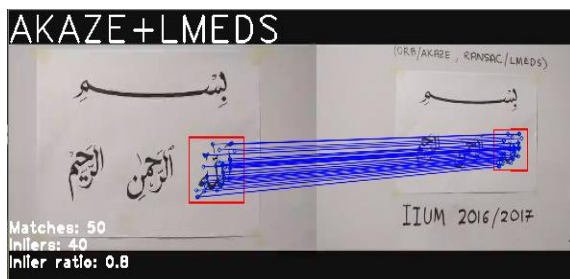


Figure 14. AKAZE+LMEDS

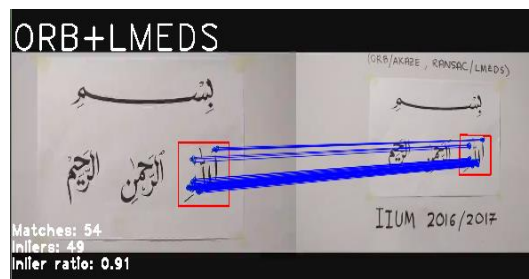


Figure 15. ORB+LMEDS

3.2. Keypoint Detection and Description

To quantify the keypoint detection and description performance of the algorithms, the average number of matches found in all frames is evaluated. In this evaluation, only the ORB and AKAZE algorithms are compared. ORB algorithm basically combines a modified version of a FAST keypoint detector with an also modified version of BRIEF keypoint descriptor. In FAST, keypoints are detected by means of scanning a candidate pixel p with its neighboring pixels by a radius r . Pixel p is detected as a keypoint when a significant amount of neighboring pixels inside

radius r are brighter or darker in intensity than candidate pixel p [8]. Meanwhile, AKAZE is branched from a parent algorithm called KAZE.

The name KAZE is Japanese for “wind” which the algorithm gets its inspiration from. In nature, the flow of wind is ruled by a non-linear process and that non-linearity concept is at the heart of the KAZE algorithm. On that basis, AKAZE was then developed by the same team of researchers as a faster iteration of KAZE [9]. AKAZE then improved upon the KAZE by applying a mathematical framework called Fast Explicit Diffusion (FED) which sped-up the non-linear scale-space computations by an order of magnitude [9]. The metrics are visualized in two forms which are real matches per frame and linear matches per frame. The results are shown in Figures 16 to 19.

Observing Table 1, AKAZE seems to perform better for simpler forms of Quranic words like “Bismi” and “Allah” with a noticeably higher number of matches found compared to ORB. On the other hand, when more complex words are tested like “Ar-Rahman” and “Ar-Raheem”, ORB performs slightly better than AKAZE. This difference in performance may be associated with AKAZE’s use of non-linear scale space method for keypoint detection [15]. This method does not use Gaussian Blurring which preserves image quality and because of this, keypoints in simple objects are better detected by AKAZE compared to ORB [17].

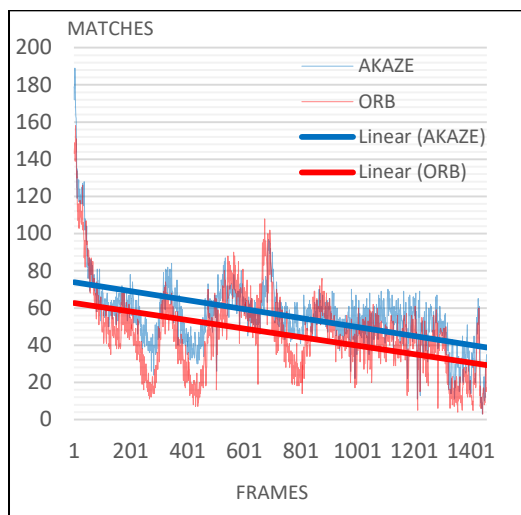


Figure 16. "Bismi" Matches

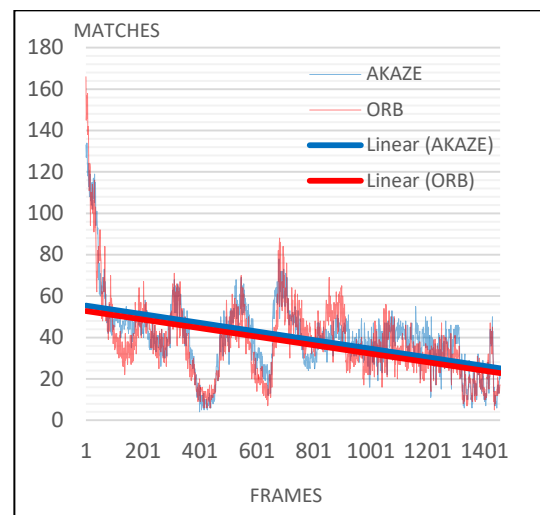


Figure 17. "Allah" Matches

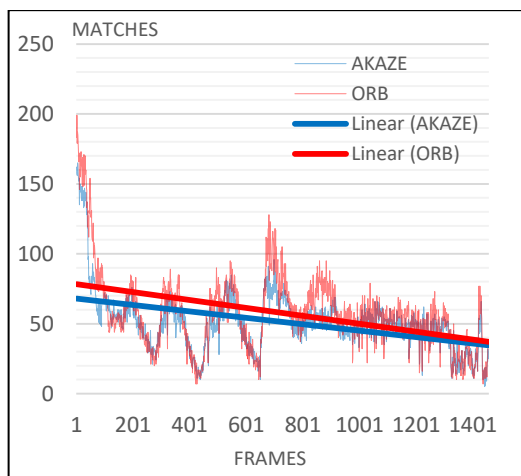


Figure 18. "Ar-Rahman" Matches

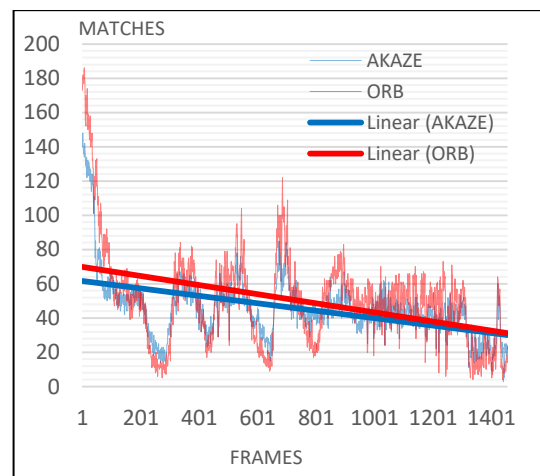


Figure 19. "Ar-Raheem" Matches

Table 1. Average Matches

DATASET	AVERAGE MATCHES	
	AKAZE	ORB
"BISMI"	56	46
"ALLAH"	40	37
"AR-RAHMAN"	51	57
"AR-RAHEEM"	45	50

3.3. Pose Estimation

For pose estimation, the homography matrix equation is used which; given an image with a set of coordinates P0, P1, P2, and P3, transforms the image from the observer's perspective on an image plane and projects it onto a world plane in a three-dimensional environment. This definition can be illustrated graphically in Figure 1 [13].

Mathematically, the two-dimensional coordinates of an input image are extracted and multiplied with a homography matrix containing nine entries to obtain the projection of the image in. The nine entries inside the homography matrix enable the image to have up to eight degrees of freedom as illustrated in equation 1.

$$\begin{bmatrix} \rho'_i x'_i \\ \rho'_i y'_i \\ \rho'_i \end{bmatrix} = \tilde{x}'_i = H \tilde{x}_i = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (1)$$

Where:

$$\begin{bmatrix} \rho'_i x'_i \\ \rho'_i y'_i \\ \rho'_i \end{bmatrix} \text{ are the } (x, y) \text{ coordinates of the resulting image in world plane.}$$

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \text{ is the homography transformation matrix.}$$

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \text{ are the } (x, y) \text{ coordinates of source image in the image plane.}$$

From equation 1 the study observes how RANSAC and LMEDS algorithms compare with each other when combined with both ORB and AKAZE keypoint detection algorithms. To gauge the performance of each algorithm combination, its mean homography accuracy ratio across all frames is determined by equation 2:

$$\bar{\alpha} = \frac{\sum_k^n \left(\frac{I_k}{M_k} \right)}{n} \quad (2)$$

Where:

$\bar{\alpha}$ = mean homography accuracy ratio.

n = total number of frames

I = number of inliers.

M = number of keypoint matches.

$k = 1, 2, 3, \dots$

RANSAC and LMEDS algorithms compare keypoint descriptions from both reference frame and real-time frame to determine the number of inliers. Inliers can never be higher than the maximum number of keypoints detected for each frame [11],[12]. Hence, an inlier to keypoint ratio of '0.9' means 90% accuracy. This accuracy ratio is considered as the evaluation metric for pose estimation. Figures 20 to 24 show the cumulative number of inliers detected per frame for each Quranic word. These Figures generally demonstrate each algorithm performance in calculating inliers.

Finally, the accuracy rate for each algorithm combination is determined by equation 2. To visualize the metric, a 'box and whiskers' graph plot is chosen. This type of graph specializes

in showing error distributions for large datasets and is particularly good in showing outliers and inliers. In Figures 24 to 27, each algorithm combination is shown side-by-side with their respective average accuracy rates recorded in Table 2.

From Table 2, the study finds that LMEDS provides the best average accuracy rate in estimating the pose of a Quranic Arabic word regardless of the type of keypoint detection algorithm used. In both AKAZE+LMEDS and ORB+LMEDS combinations, the results achieved higher average accuracy rates than their respective RANSAC counterparts. Also, ORB+LMEDS algorithm combination provided the best overall results with all tests achieving more than 80% average accuracy rate.

The superior accuracy obtained by LMEDS can be attributed to its unique parameter estimation method. LMEDS automatically sets the error-rejection parameters from the dataset whereas, in RANSAC, the user has to set them manually [11]. This automated system enables LMEDS to provide the best possible parameters to minimize errors for pose estimation. Hence, a combination of ORB+LMEDS algorithms is suggested for pose estimation of Quranic Arabic words limited to within the mentioned datasets.

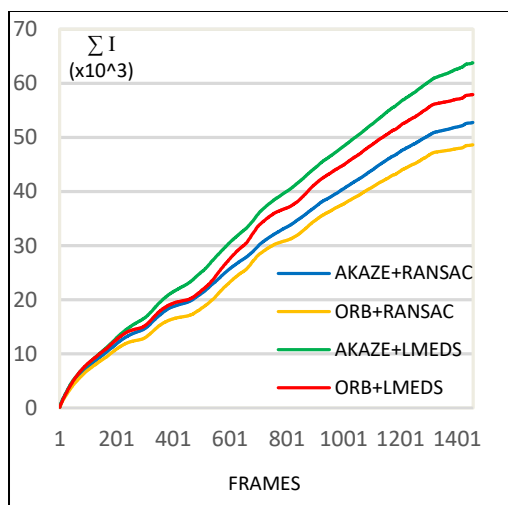


Figure 20. "Bismi" Cumulative Inliers

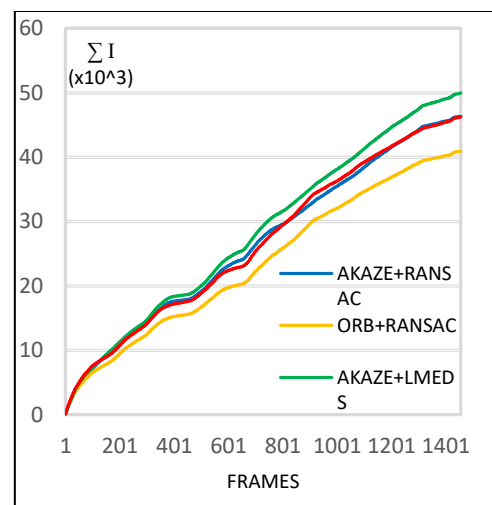


Figure 21. "Allah" Cumulative Inliers

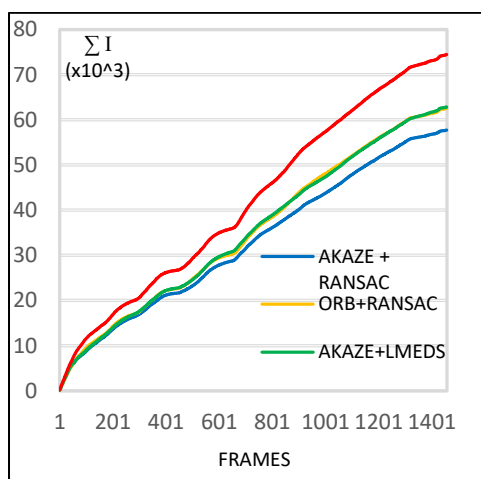


Figure 22. "Ar-Rahman" Cumulative Inliers

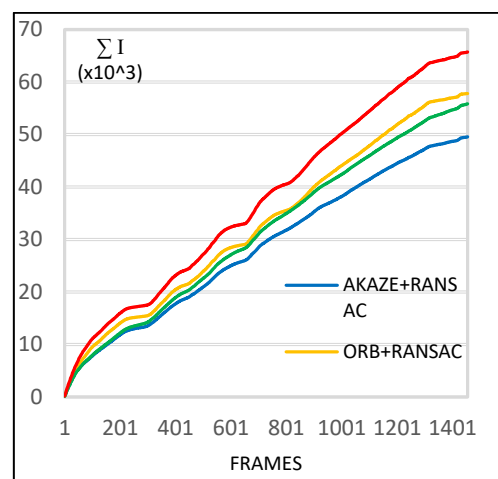


Figure 23 "Ar-Raheem" Cumulative Inliers

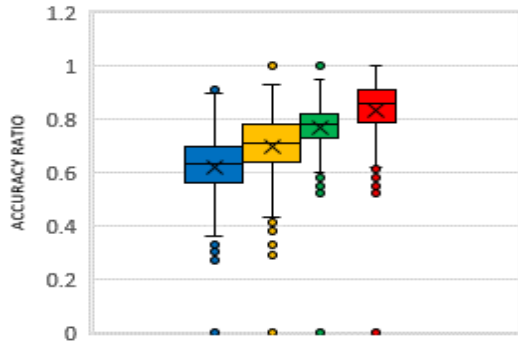


Figure 24. "Bismi" Accuracy

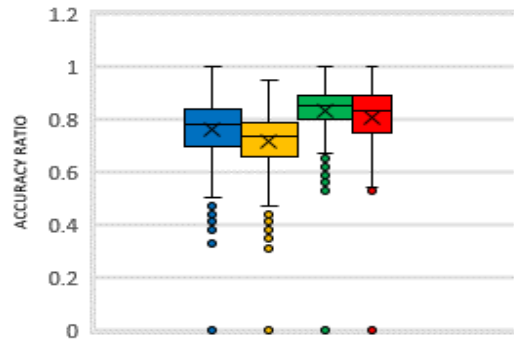


Figure 25. "Allah" Accuracy

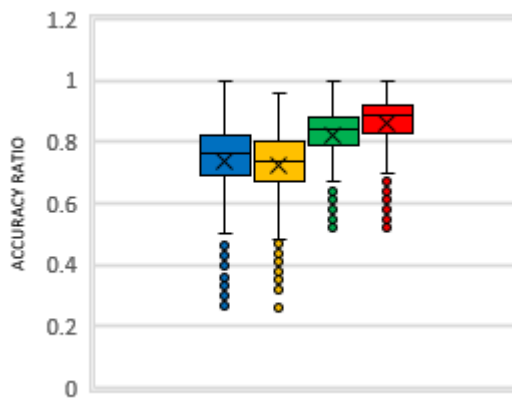


Figure 26. "Ar-Rahman" Accuracy

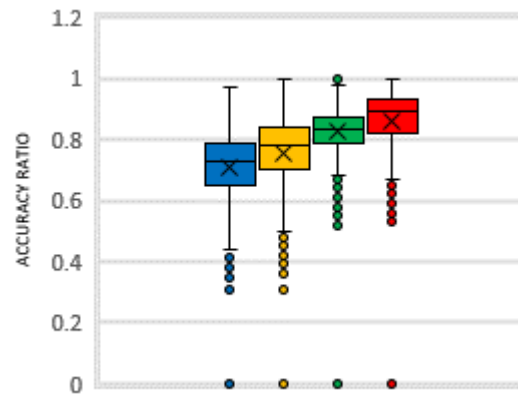


Figure 27. "Ar-Raheem" Accuracy

*(Figure 24, 25, 26, 27 Legend: ■KAZE+RANSAC ORB+ ■ANSAC AKAZE+LM ■DS ORB+LMEDS) ■

Table 1. Mean Accuracy Rate

DATASET	MEAN ACCURACY RATIO, $\bar{\alpha}$			
	AKAZE + RANSAC	ORB + RANSAC	AKAZE + LMEDS	ORB + LMEDS
"BISMI"	0.617	0.699	0.769	0.836
"ALLAH"	0.759	0.716	0.831	0.808
"AR-RAHMAN"	0.738	0.722	0.824	0.861
"AR-RAHEEM"	0.709	0.756	0.823	0.857

4. Conclusion

This study has conducted an extensive investigation into the selected keypoint detection, description and pose estimation algorithms for Quranic Arabic words. The ORB and AKAZE algorithms have been combined with RANSAC and LMEDS to produce four different algorithm combinations that have been tested on four unique test subjects taken from a selected Quranic word. Taking the results and analysis that have been made into consideration, the study showed that ORB+LMEDS algorithm combination had the best average accuracy rate out of all four algorithms used for pose estimation.

Furthermore, the study also found out that in the case of keypoint detection and description, ORB algorithm proved superior for complex words while AKAZE performed better for simpler words. However, room for improvements is still available to produce more comprehensive results. Since this study extracted only one phrase from the Quran, future studies can include more phrases for experimentation which would equate to more accurate pose estimation results of Quranic words. Secondly, other keypoint detection, description, and pose estimation algorithm combinations can be included in the experiment that might produce

better results than ORB+LMEDS. Finally, future experiments can be done on more powerful devices that are able to process full resolution videos. The compression done on the video in this study might have had an effect in keypoint and inlier calculations.

References

- [1] Ashken S. What is Computer Vision-Post 5: A Very Quick History. *blippAR*, [Internet]. 2016. Available from: <https://blippar.com/en/resources/blog/2016/10/04/what-computer-vision-post-5-very-quick-history/>.
- [2] Zhao Y, Lei J. A Camera Self-Calibration Method Based on Plane Lattice and Orthogonality. *Telecommunication Computing Electronics and Control (TELKOMNIKA)*. 2013; 11(4): 767-74.
- [3] Işık Ş, Özkan K. A Comparative Evaluation of Well-known Feature Detectors and Descriptors. *Int. J. Appl. Math. Electron. Comput.* 2014; 3(1): 1.
- [4] Li Z, Selviah DR. Comparison of Image Alignment Algorithms. :2-5.
- [5] Choi S, Yu W. Performance Evaluation of RANSAC Family. 2009. :1-12.
- [6] Chien H, Chuang C, Chen C, Klette R. *When to Use What Feature ? SIFT, SURF, ORB, or A-KAZE Features for Monocular Visual Odometry*. Image and Vision Computing New Zealand (IVCNZ). International Conference. 2016; (1): 0-5.
- [7] Bang J, Lee D, Kim Y, Lee H. *Camera Pose Estimation Using Optical Flow and ORB Descriptor in SLAM-Based Mobile AR Game*. Platform Technology and Service (PlatCon), 2017 International Conference. 2017: 8-11.
- [8] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: *An efficient alternative to SIFT or SURF*. Proc. IEEE Int. Conf. Comput. Vis. 2011: 2564-2571.
- [9] Alcantarilla P, Nuevo J, Bartoli A. *Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces*. Proceedings Br. Mach. Vis. Conf. 2013: 13.1-13.11.
- [10] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*. 1981; 24(6): 381-395.
- [11] Meer P, Mintz D, A Rosenfeld, DY. Kim. Robust regression methods for computer vision: A review. *Int. J. Comput. Vis. Apr.* 1991; 6(1): 59-70.
- [12] Divya G, Sekhar CC. Image Mosaicing for Wide Angle Panorama. *International Journal of Electrical and Computer Engineering*. 2015; 5(5).
- [13] Chum O, Pajdla T, Sturm P. The geometric error for homographies. *Comput. Vis. Image Underst.* Jan. 2005; 97(1): 86-102.
- [14] The Power Of Bismillahirrahmanirrahim. *ASMAUL HUSNA*, [Internet]. 2015. Available from: <http://www.asmaul-husna.com/2016/02/the-power-of-bismillahirrahmanirrahim.html>.
- [15] OpenCV. OpenCV: AKAZE and ORB planar tracking. *OpenCV*, [Internet]. 2016. Available from: http://docs.opencv.org/3.2.0/dc/d16/tutorial_akaze_tracking.html.
- [16] Alcantarilla PF, Bartoli A, Davison AJ. KAZE Features. *European Conference on Computer Vision*, 2012: 214-227. Springer, Berlin, Heidelberg.
- [17] Rachmawati E, Suwardi IS, Khodra ML. Review of Local Descriptor in RGB-D Object Recognition. *Telecommunication Computing Electronics and Control (TELKOMNIKA)*. 2014; 12(4): 1132-41.