Computational Speed and Qualitative Assessment of Real-Time Image Stitching Algorithm

Dhanesh Kumareswaran [1], Nazri Nasir [2], Muhammad Zulkarnain Abd Rahman [3], Abu Sahmah M. Supa'at [4]
[1,2] *School of Mechanical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia.*
[3] *Department of Geoinformation, Faculty of Built Environment and Surveying, Universiti Teknologi Malaysia.*
[4] *School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia.*
E-mail: dhanesh4@graduate.utm.my[1], mnazrimnasir@utm.my[2], mdzulkarnain@utm.my[3], abus@utm.my[4]

2021 International Conference on Communication, Control and Information Sciences (ICCISc) | 978-1-6654-0295-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCISc52257.2021.9484887

***Abstract* — In remote sensing and environmental mapping, Unmanned Aerial Vehicle (UAV) has been used extensively to capture images. For many years, digital maps are generated by using a method called image stitching. It is a method of combining multiple images to produce a segmented panorama. Since this method has been commonly used, many users produce an accurate map by using commercial software. However, a downside of this commercial software is a long computational time which is not appropriate for immediate mapping activities at chaos areas in particularly during rescue missions. This paper proposes a method to speed up the process of map generation by using a revised real-time image stitching algorithm including the qualitative assessments. In this research, the images are extracted from a video taken by a drone that flies in a dedicated flight path. These videos are then immediately transmitted to a ground station for further image processing and computation. The overlapping images are stitched and later undergoes features extraction process to identify the common features between the images. These common features are used to compute homography matrix which beneficial for image wrapping. The finding of this study suggests that ORB and AKAZE are the most suitable descriptors to be used in real-time image stitching because of their fast computational speed at adequate level quality. For instance, at 5 number of skip frame, ORB is at least 2-fold faster than AKAZE and goes up to 10-fold faster than BRISK.**

Keywords— image stitching, UAV, detection, real-time.

I. INTRODUCTION

Image stitching is a process of combining several sets of overlapping images into a single and broader image known as panorama [25]. This image stitching technique is not only limited in constructing panorama images but it is also used in other fields such as 3-D image reconstruction [3], architecture [6], remote sensing, underwater survey and mapping as well as forensics [18]. It is also known that image stitching is intensively involved in mapping the disaster-affected areas such as floods, landslides, and earthquakes [7, 9].

However, a major problem of these traditional methods of image stitching embedded in commercial software requires a lengthy processing and computation time [7]. Previous studies confirmed that the volatile processing time depends on the quantity of the image, size of the area, and precession level [14]. Earlier findings also found that these stitched images produced drift error and impair its horizontal positional accuracy [10]. Furthermore, there is a need for real-time image stitching for immediate rescue missions which requires elimination of the lengthy post-processing procedures with better accuracy [7, 9].

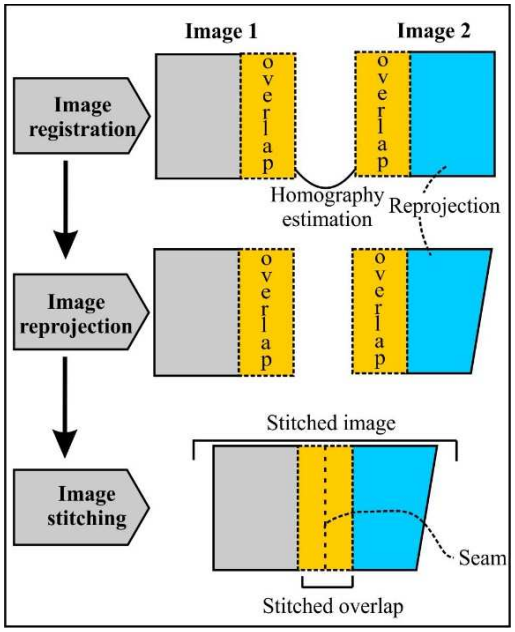

Fig. 1: The workflow of image stitching where the overlapping images are stitched into a large image containing seam.

This study discusses the background on feature-oriented image stitching in SECTION II. The methodology of feature-oriented real-time image stitching algorithm and procedures for the experimental setup in SECTION III. This study revised, reformulated the earlier real-time image stitching algorithm, and finally assessed the quality of stitched image using a quantifiable method. The experimental setup outcomes are discussed in SECTION IV and the conclusion remarks in SECTION V.

II. . BACKGROUND

As shown in Fig. 1, image stitching is composed of image registration, reprojection, and stitching, which later can be classified further into 2 categories which are area-based method and feature-based method [8]. The overlapping images can be registered either in region-based or feature-based methods [13]. The region-based method tries to minimise the percentage of error based on the difference of overlapping in a respective image to a

reference image by iteration. Although this method is highly accurate, it is associated with high computational cost because of the iteration process in its algorithm. Meanwhile, feature based detection impose on the feature points which are established in the photos, however, the established corner points are widely used to stitch images [25]. In this study, we used feature-based stitching algorithm to combine the overlap regions.

A. Image Registration

For feature-based registration, previously there were several approaches used such as Kanade-Lucas-Tomasi tracker (KLT) [13], matching of Harris corners [8], Features from Accelerated Segment Test (FAST) [8], Scale Rotation Invariant (SIFT) [8, 17] and SURF [20]. If the overlapping images cannot be matched because of motion blur in pictures or poor homographic, Random Sample Consensus (RANSAC) algorithm is used to stitch these images [13]. In addition, an algorithm known as features detector will detect the feature available in the images. The detected features are described using feature descriptor (float or binary point descriptor).

In recent years, there is an increasing number of literatures describing some feature detectors and descriptor for real-time purposes. There are several descriptors invented by the pioneering studies such as;

i. Oriented FAST and Rotated Brief (ORB) has a binary feature detector. The application of ORB as a real-time feature detector has been extensively used by many researches for especially real-time image stitching [5]. Essentially, ORB is a modified and refined feature detector of Features from Accelerated Segment Test (FAST) and Binary Robust Independent Elementary Feature (BRIEF).

ii. Binary Robust Invariant Scalable Keypoints (BRISK) algorithm is an invariant to rotation and scaling. Even though most of the feature detection method uses float detectors, BRISK is a binary feature detector. BRISK and integrated FAST as filtering system can easily detect the corner points using AGAST algorithm. This method has been used intensively for real-time stitching in underwater [11].

iii. Scale-Invariant Feature Transformation (SIFT) was initially introduced by Lowe (2004) has 4 major steps which are scale-space extrema detection, keypoint localization, orientation assignment, and defining keypoint descriptors [17]. This method is invariant to scaling, rotation, and translation transformation [8]. It was used previously for real-time image stitching by Rizk and the co-workers [21].

iv. KAZE is a two-dimensional (2D) feature detection algorithm and not widely used because of its computational time and not suitable for real-time application [16]. Hence, the upgraded AKAZE algorithm was introduced, which is an accelerated form of KAZE. Besides, AKAZE is binary descriptor whereas KAZE is a float descriptor [24].

B. Image Reprojection

Image transformation can be divided mainly into 2 subcategories, which are two dimensional to two dimensional (2D-2D) and two dimensional to three dimensional (2D-3D) [23]. There are several types of 2D transformation which include translation, Euclidean, affine, and projective Fig. 2. To date, projective and affine transformation is the most common type of reprojection in image stitching [15]. For affine transformation, all parallel lines will be preserved whereas, for projective transformation, all straight lines will be kept [2, 15, 23]. It was decided that this study will focus on 2D transformation which valid for image stitching.

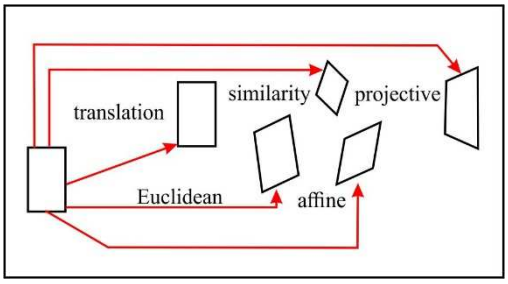

Fig. 2: Basic example of 2D image transformation [23]

III. METHODOLOGY

There are two subdivisions in this section, (A) prerequisites, which discuss in detail the material and setup of the experiment and (B) algorithm, which examine in-depth the algorithm development process for real-time image stitching.

A. Image processing device

In this study, we used a computer Intel i7-1070F, 16 GB RAM, and 8 GB memory graphic processor as an image processing tool. All the image processing is executed at a ground station where the computer is placed. Besides, we filmed the aerial images using a multicopter DJI Mini 2 at altitude 50 meters from the ground using its build-in camera (1/2.3" CMOS, 12 megapixel, 2720×1530 video resolution, mp4 format, SZ DJI Technology Co., Ltd., China). As the aim of this study is to formulate the real-time image stitching algorithm, the video transmission device was excluded. In this study, the image stitching is considered real-time when the algorithm produces near-instantaneous output with very short latency requirements while still continuously streaming the input images [4]. Multicopter flew at the specified altitude, recorded, and stored the aerial footages in a SD card, before manually transferred to the image processing computer (Fig. 3).

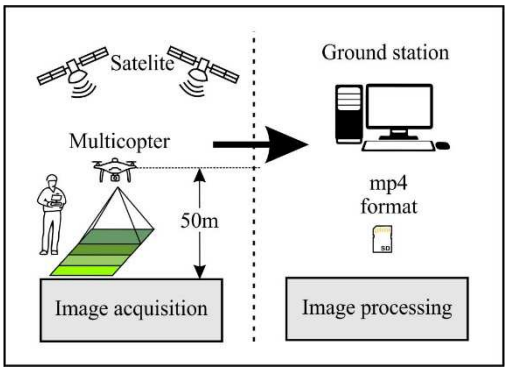

Fig. 3: Image stitching workflow

B. Development of revised algorithm

Fig. 4: Image stitching using the earlier versions of algorithm. See Fig. 5 to see the revised algorithm.

The algorithm used in this study was synthesized based on the pioneering studies, which include SIFT, ORB, BRISK, and AKAZE to detect features in every frame of prerecorded video [1, 4]. In this study, we used different descriptors to investigate and further identify the most appropriate feature detector for real-time image stitching. In the most recent studies, most of the researchers have tended to focus on ORB as a feature detector compared to KAZE and BRISK [4]. The process began with feature detection of the first frame I_K and second frame as I_{K+1} of the video, then followed with estimation of a homography using the RANSAC algorithm (exclude bold frame lines, Fig. 5). The homography matrix was computed using I_{K+1} as reference and image I_K was wrapped according to the computed homography into I_s (*s* stands for stitch). Then, both I_s and I_{K+1} were stitched together to form I_S. Finally, the stitched image, I_S was assigned as I_K then it continues the iterative process until all the frames in the video fully computed (See Fig. 4 for the final output image).

So far, it was discovered that earlier algorithm (unrevised SIFT) uses a longer computational time (2-fold slower compared to the present algorithm, see Fig. 6), and the final stitch image is in the perspective of the last frame of the video as shown in Fig. 4. The stitched image has poor quality with a longer computational time because of the new frame, I_{K+1} was chosen as the reference frame to compute the homography matrix. The earlier algorithm produced distorted images because it changed the perspective along with the video (Fig. 4 as an exemplary case). In addition, the first frame of the video is not much visible because the stitched image wraps according to the new frame respectively.

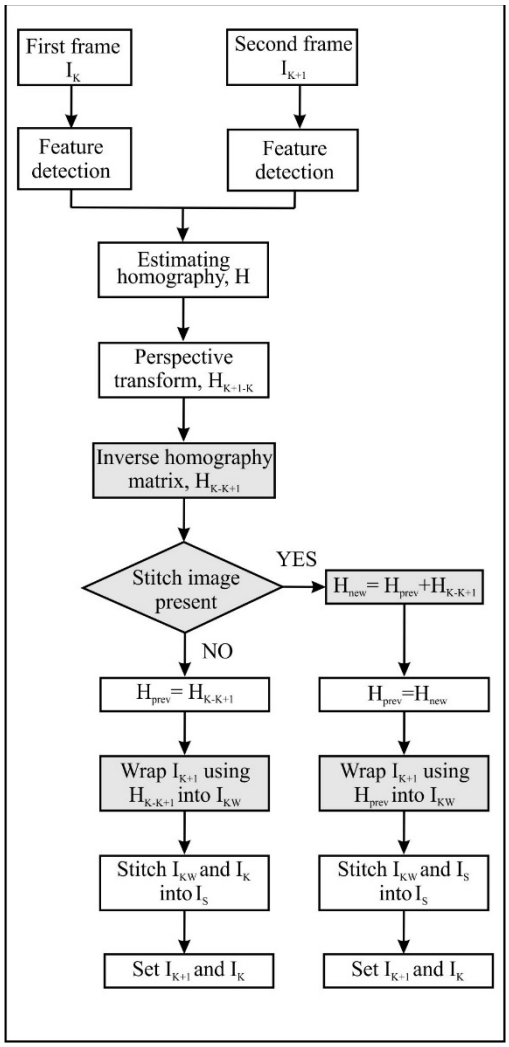

Fig. 5: Improvised image stitching flowchart. The grey blocks describe the additional and revised process.

Therefore, the earlier algorithm was later improved whereby the homography matrix was computed using frame I_{K+1} to I_K (Fig. 5). Then, the homography matrix was used to wrap I_{K+1} in respective I_K to form I_{KW}. After that, the wrapped image I_{KW} was stitched with I_K and formed I_S. I_{K+1} was then set as I_K and finally, the homography matrix was added continuously to ensure the new frame wraps according to the initial reference frame of the video.

C. Qualitative assessment

SSIM is a measurement of structural similarity between the reference image and target image [22]. The value of SSIM ranges from 0 to 1 (1 indicates that both images have the same structural similarity and 0 the vice versa). In addition, SSIM can detect the loss of information between images, but it requires a reference image to identify the structural similarity [22]. The SSIM is calculated using various windows x and y in the target image and reference

image. The following is the mathematical equation of SSIM:

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (1)$$

Where,

$\mu_x\mu_y$ Mean value for the window x and y

σ_x, σ_y Variance value of window x and y

σ_{xy} Covariance of window x and y

c_1, c_2 Product of dynamic range of pixel value (8-bit image pixel value is 255) with a coefficient (c_1 0.01 and c_2 0.03)

PNSR is a ratio between the maximum signals to the noise present [9, 27]. If the value of PNSR is high, it indicates less noise is present in the image and vice versa if the value is low. PNSR does not have a set of range values like SSIM. Its value can range from 0 up to infinity [27]. However, PNSR requires a reference image to evaluate the target image. Following is the formula of PNSR:

$$PNSR=10\log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (2)$$

Where MAX is the maximum pixel value in both images. For example, an 8-bit image will have a maximum pixel value of 255. MSE is the means square error between two images. The following is the formula for MSE:

$$MSE=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(I(x,y)-K(x,y)\right)^2 \quad (3)$$

Where MN refers to the size of image. I refer to the reference image, and K is the target image that contains noise.

RMSE is derived from the formula of MSE. RMSE indicated the root-mean-square error in between the target image and reference image. Similarly, RMSE requires a reference image to evaluate the target image. Following is the formula for RMSE:

$$\text{RMSE}=\sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(I(x,y)-K(x,y)\right)^2} \quad (4)$$

'MN' refers to the size of the image, 'I' refers to the reference image and 'K' refers to the target image.

BRISQUE is preferred to evaluate the 2D plane image quality because it is specifically designed for 2D image features [12]. Vaidya and co-workers found that BRISQUE has one of the suitable methods to identify distortion in the image [26]. BRIQUE model uses natural scene statistics to determine the image quality. If there is a presence of distortion in the target image the value of BRISQUE would be high and the lower BRISQUE score will represent no distortion [19]. The model of BRISQUE score has been trained with a distorted image with such as blur and noise [19, 26].

IV. RESULT AND ANALYSIS

A suitable feature detector is needed to be identified for this real-time image stitching algorithm. Thus, the researchers performed the computation using the improvised algorithm explained in Fig. 5 to detect the detector. The experiment was executed by varying the number of skipped frames versus the average computation (seconds) using 5 different detectors.

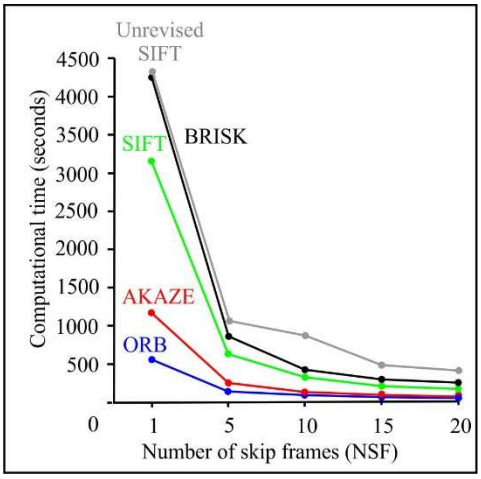

Fig. 6: Computational time at a different number of the skipped frame. BRISK, SIFT, AKAZE, and ORB are revised-algorithm whereas unrevised SIFT is the earlier version of algorithm. The number of skipped frames (NSF) was calculated by dividing the total frames of drone video with frames used. For example, the NSF value is 1, which means all frames were used accordingly.

Typically, the computational time of all detectors was significantly reduced as the number of the skipped frame numbers increase (Fig. 6). ORB had the fastest computational time compared to other detectors. ORB detectors have binary descriptors with a faster computational time, while SIFT detectors use float descriptors during computation. Meanwhile, AKAZE which also use binary descriptor demonstrated fast computation equivalent to ORB. However, this fast computational time has an undesirable implication in reducing the stitching quality which might fail to meet the mapping requirement.

Image quality evaluation has a major problem because human eyes-based assessment is very subjective and tends to differ from person to person [9]. Until recently, little attention has been paid to discuss the stitch image quality assessment [22]. In this study, we adopted the assessment methods for the stitched images invented by the pioneering research [1, 22]. During the assessment, the last frame was used as a reference image to evaluate the cropped part of the

stitched image. Then, the images were evaluated in the prospect of Root Mean Square Error (RMSE), Signal-to-Noise Ratio (PNSR), and Structural Similarity Index Measurement (SSIM). In addition, the BRISQUE score is also identified as one of the methods to access the quality of image even without a reference image.

Fig. 7: The representative snapshot of a stitched image using AKAZE, BRISK, ORB and SIFT when NSF is 5. See https://bit.ly/2SySv85 for the output of the improvised algorithm.

According to the plotted results (Fig. 8), RMSE values increase with the increasing values of NSF. According to RMSE values, the most efficient stitching quality occurred when the NSF value ranged from 1 to 5. By compromising the computational time and the noise present in the stitched image, the maximum number to skip frame is 5, because most of the intersection point is in between 1 to 5 NSF values. Despite having a 5% more RMSE value than other detectors, ORB showed a high-quality stitch image with a faster computational time compared to other feature detectors (ORB 6 -fold faster than SIFT, 10-fold faster than BRISK, and 2-fold faster than AKAZE). The PNSR values reduce as the number of skipped frames (NSF) increased and the highest PNSR value achieved by the BRISK algorithm. BRISK formulation produces relatively good quality images with minimum distortion or visible seams. Stitched images using ORB formulation have the lowest PNSR value compared to other feature detectors. Notably, there was no significant difference in the plotted SSIM values except for ORB because it has the least structural similarity compared to the other detectors (1% lower than SIFT, and 2% lower than AKAZE and BRISK), and it has the least structural similarity compared to the other detectors. ORB algorithm, on the other hand, has the lowest computational time and high structural similarity. According to the BRISQUE score, ORB has the most distortion in its stitched image compared to other detectors (13% higher than AKAZE, 21% more than BRISK, and 12% higher than SIFT), but it computation time is relatively low compared to other detectors.

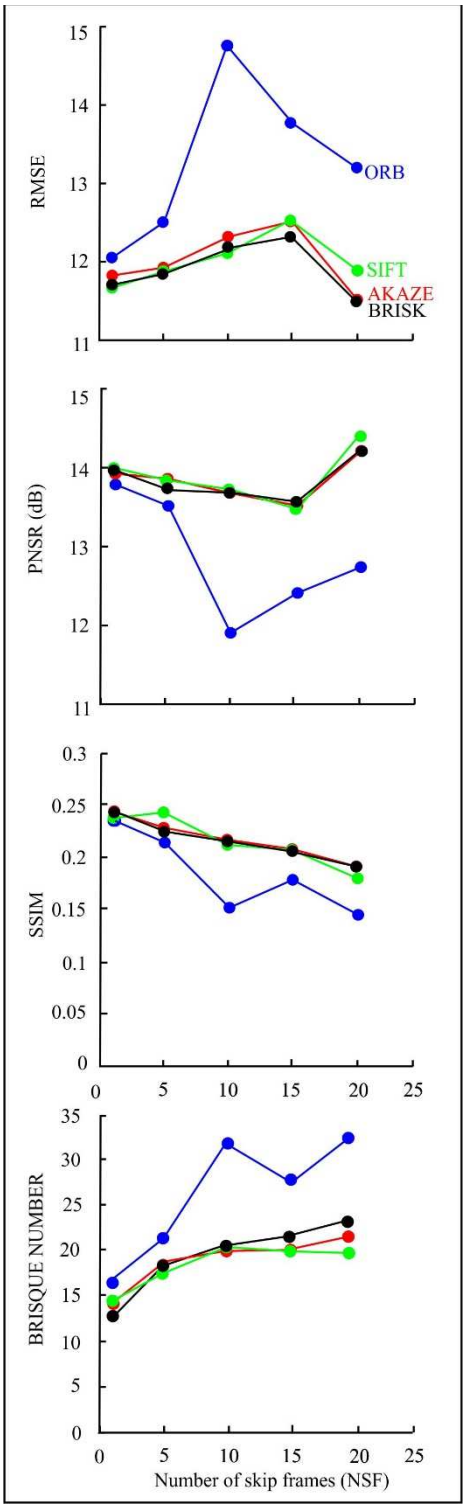

Fig. 8: The number of skip frames versus RMSE, PNSR, SSIM, and BRISQUE SCORE. A higher value of PNSR values explains less noise is present in the image compared to the reference image. The SSIM value ranges from 0 to 1, to describe the structural similarity between both images where 1 denotes a similar image is detected. Besides, higher values of BRISQUE score denotes more distortion is present in the stitched image. A higher PNSR value reveals less noise existed in the image compared to the reference image.

V. CONCLUSION

In conclusion, real-time image stitching is a needed solution, especially in remote sensing and rescue mission purposes. ORB and AKAZE are suitable descriptors to be used in real-time image stitching because of their fast computational speed. As a future recommendation, parallel processing is suggested to reduce the computational time. In addition, high-resolution aerial images are suggested to be used in image stitching to improve the spatial quality of the stitched image. Finally, the geometrical quality of the stitched image can be improved by applying geo-referencing procedures which progressively warp and fits the image to a geographic location.

ACKNOWLEDGEMENTS

This work was funded by a scholarship from the Universiti Teknologi Malaysia (UTM) and the Ministry of Education of Malaysia (MOHE) to the first author. This project is also supported by Aeronautic Laboratory (Aerolab), UTM, and UTM High Impact Research Grant (Reference number: PY/2019/02778). We would like to thank Sarah Azreen Muhamad for critically reading the manuscript and the two anonymous referees for their constructive comments.

REFERENCES

[1] N. Abdelkrim, K. Atmane, and S. Houari, "Quantitative Analysis of Real-Time Image Mosaicing Algorithms," in *International Conference on Systems, Signals, and Image Processing*, vol. 2018-June,

[2] D. Avola, L. Cinque, G. L. Foresti, and D. Pannone, "Homography vs similarity transformation in aerial mosaicking: which is the best at different altitudes?," *Multimedia Tools and Applications,* Article vol. 79, no. 25-26, pp. 18387-18404, 2020.

[3] R. Battulwar *et al.*, "A Practical Methodology for Generating High-Resolution 3D Models of Open-Pit Slopes Using UAVs: Flight Path Planning and Optimization," (in English), *Remote Sensing,* vol. 12, no. 14, p. 2283, Jul 2020.

[4] R. de Lima, A. A. Cabrera-Ponce, and J. Martinez-Carranza, "Parallel hashing-based matching for real-time aerial image mosaicing," *Journal of Real-Time Image Processing,* Article 2020.

[5] R. De Lima and J. Martinez-Carranza, "Real-time aerial image mosaicing using hashing-based matching," in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS 2017*, pp. 144-149.

[6] C. Eschmann, C. M. Kuo, C. H. Kuo, and C. Boller, "Unmanned Aircraft Systems for Remote Building Inspection and Monitoring," 01/01 2012.

[7] P. Fanta-Jende, D. Steininger, F. Bruckmüller, and C. Sulzbachner, "A versatile UAV near real-time mapping solution for disaster response - Concept, ideas and implementation," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 43, B1 ed., pp. 429-435.

[8] D. Ghosh and N. Kaabouch, "A survey on image mosaicing techniques," *Journal of Visual Communication and Image Representation,* vol. 34, pp. 1-11, 2016/01/01/ 2016.

[9] D. Ghosh, S. Park, N. Kaabouch, and W. Semke, "Quantitative evaluation of image mosaicing in multiple scene categories," in *2012 IEEE International Conference on Electro/Information Technology*, 6-8 May 2012, pp. 1-6.

[10] D. Hein, T. Kraft, J. Brauchle, and R. Berger, "Integrated UAV-based real-time mapping for security applications," *ISPRS International Journal of Geo-Information,* Article vol. 8, no. 5, 2019, Art no. 219.

[11] D. A. Jatmiko and S. U. Prini, "Study and Performance Evaluation Binary Robust Invariant Scalable Keypoints (BRISK) for Underwater Image Stitching," *IOP Conference Series: Materials Science and Engineering,* vol. 879, p. 012111, 2020/08/07 2020.

[12] H. Jiang *et al.*, "Cubemap-Based Perception-Driven Blind Quality Assessment for 360-degree Images," *IEEE Transactions on Image Processing,* vol. 30, pp. 2364-2377, 2021.

[13] M. S. Kamel and A. Campilho, "Image Analysis and Recognition, 4th International Conference, ICIAR 2007, Montreal, Canada, August 22-24, 2007, Proceedings," 01/01 2007.

[14] E. Khoramshahi, R. A. Oliveira, N. Koivumäki, and E. Honkavaara, "An image-based real-time georeferencing scheme for a UAV based on a new angular parametrization," *Remote Sensing,* Article vol. 12, no. 19, pp. 1-27, 2020, Art no. 3185.

[15] J. Kim, T. Kim, D. Shin, and S. H. Kimb, "Robust mosaicking of UAV images with narrow overlaps," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 2016-January, pp. 879-883.

[16] H. Liu and G.-F. Xiao, "Remote Sensing Image Registration Based on Improved KAZE and BRIEF Descriptor," *International Journal of Automation and Computing,* vol. 17, no. 4, pp. 588-598, 2020/08/01 2020.

[17] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004/11/01 2004.

[18] M. A. Mat Amin, S. Abdullah, S. N. Abdul Mukti, M. H. A. Mohd Zaidi, and K. N. Tahar, "Reconstruction of 3D accident scene from multirotor UAV platform," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 43, B2 ed., pp. 451-458.

[19] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain," *IEEE Transactions on Image Processing,* vol. 21, no. 12, pp. 4695-4708, 2012.

[20] J. Qi *et al. Image Stitching Based on Improved SURF Algorithm, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11744 LNAI, pp. 515-527, 2019.

[21] M. Rizk, A. Mroue, M. Farran, and J. Charara, "Real-Time SLAM Based on Image Stitching for Autonomous Navigation of UAVs in GNSS-Denied Regions," in *Proceedings - 2020 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2020*, pp. 301-304.

[22] S. K. Sharma, K. Jain, and M. Suresh. *Quantitative evaluation of panorama softwares, Lecture Notes in Electrical Engineering*, vol. 500, pp. 543-561, 2019.

[23] R. Szeliski, "Image Alignment and Stitching: A Tutorial," *Foundations and Trends® in Computer Graphics and Vision,* vol. 2, no. 1, pp. 1-104, 2007.

[24] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 3-4 March 2018, pp. 1-10.

[25] O. S. Vaidya and S. T. Gandhe, "The Study of Preprocessing and Postprocessing Techniques of Image Stitching," in *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, 8-9 Feb. 2018, pp. 431-435.

[26] O. S. Vaidya and S. T. Gandhe, "The Study of Preprocessing and Postprocessing Techniques of Image Stitching," in *2018 International Conference On Advances in Communication and Computing Technology, ICACCT 2018*, pp. 431-435.

[27] Z. B. Wang and Z. K. Yang, "Review on image-stitching techniques," *Multimedia Systems,* vol. 26, no. 4, pp. 413-430, Aug 2020.