Triangular Regions Representations for Matching Images With Viewpoint Changes

IBRAHIM EL RUBE', (Senior Member, IEEE), AND SAMEER ALSHARIF
Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia

Corresponding author: Ibrahim El Rube’ (Ibrahim.ah@tu.edu.sa)

This work was supported by the Taif University Researchers Supporting Project through Taif University, Taif, Saudi Arabia, under Grant TURSP-2020/348.

ABSTRACT This paper proposes triangular region representations based on keypoints detected in images with viewpoint changes. The strongest keypoints in the reference and query images are allocated individually using a previously published contourlet-based approach to determine the keypoints. These selected keypoints serve as vertices of the triangular regions to be transformed into rectangular representations as simple numeric matrices. The suggested representation methods used to form rectangular matrices are full triangle representation (FTR) and a lighter representation called triangle medians and sides representation (TMSR). For the former, the intensity values along the lines connecting the anchor keypoint to the points between the other two triangular vertices form the rows of the representation matrix. These two triangle vertices are located in an allotted window around the anchor keypoint from the nearby keypoints. For the latter, the intensity values on the six triangular lines, medians, and sides formed the resulting rectangular matrix. The proposed representations are validated for image-matching applications using a descriptor-less matching method. Moreover, the performances of these algorithms are compared with those of traditional algorithms. The results confirmed the superiority of the proposed method over these algorithms.

INDEX TERMS Affine transformation, image matching, keypoints, triangular region representation.

I. INTRODUCTION

The detection and matching of keypoints are well-known problems in image-similarity judgment applications. For example, automated monitoring systems, such as monitoring the melting of ice at the poles due to global warming or observing geographical or environmental changes for a specific spot on the planet, need to maintain the image at a specific time to compare it with itself as time progresses. In addition, multimodality in biomedical imaging requires accurate registration based on reliable matching algorithms to achieve perfect visualization for diagnostic purposes. Feature extraction from targeted images is an essential element of image matching. In image processing, such features can be extracted as patterns, color discrepancies, or morphological datasets. Many algorithms have been developed to describe such features by considering image attacks, including noise, scaling, deformation, and image perspective. These descriptors can be classified based on shape or keypoint manipulation into edge-based descriptors [1]. Despite their high computational cost, keypoint-based descriptors are robust against many artifacts because they operate at the level of neighboring pixels, regardless of image rotation, light intensity, image perspective, or image scaling.

Existing methods, such as speeded up robust features (SURF), scale-invariant feature transform (SIFT), binary robust invariant scalable keypoints (BRISK), and KAZE (wind in Japanese), generally exhibit competitive performance and accuracy in similarity transformations. However, although these methods are invariant to changes in rotation and scale, they may suffer from a higher degree of viewpoint change, including skew and perspective transformations. Therefore, this paper presents a methodology based on triangular region representations to address the issues associated with affine transformation, which affects the performance of conventional methods.

This paper contributes to the body of knowledge as follows: Triangular regions with vertices based on keypoints are transformed into rectangular forms to address changing
point-of-view issues such as affine and weak perspective transformations.

Two representations are presented: full triangle representation (FTR) and triangle medians and sides representation (TMSR).

- A descriptor-less matching scheme based on triangular representations is introduced.

The remainder of this paper is organized as follows. The following section presents a literature review, followed by a detailed explanation of the proposed observations. Section 4 presents an image matching algorithm that uses two representations. The validation and evaluation of the FTR and TMSR methods are described in section 5. The conclusions of this study are significant.

II. RELATED WORKS

Recently, machine learning has attracted the attention of many researchers for matching images by using feature extraction. Furthermore, it was employed to determine the optimal number of keypoints and the size of their neighborhood windows. Although this technique may be helpful, particularly in real-time applications, it requires a large feature vector for the training. By contrast, keypoint-based descriptors require fewer keypoints, optimized depending on the method used.

This study aims to represent and match images that have undergone affine or weak perspective transformations. Triplets are used as the simplest representative affinity. The algorithm used in this study is based on the local neighborhood of the keypoints. Therefore, we highlight up-to-date local pixel-wise work to develop a consistent comparative study.

Most classical image-matching algorithms are based on descriptors deduced from keypoint regions. Moreover, they depend on feature extraction and matching using similarity metrics such as the sum of the squared error and mutual information [2]. The procedure for such algorithms can be summarized as follows:

**Step 1:** Reading both images: Reference and query. It is preferable to preprocess these images to enhance their quality and ease the keypoint-detection process.

**Step 2:** Detect keypoints using known detection algorithms, and select the strongest keypoints.

**Step 3:** Compute the descriptor values in each image separately for the neighboring regions of the detected keypoints.

**Step 4:** Computing the similarity/distance between descriptors across both images.

**Step 5:** Matching and refining the correspondence of keypoints between two images.

The papers [3]–[5] attempted to match query and target images based on the one-to-one relationship between keypoints in both images as candidate points to form constraint triangles that contain features. Many studies have attempted to improve matching for such types of images. The paper [1] proposed an enhancement method for keypoint refinement. First, the best keypoint matches were selected based on the smallest distance between the keypoints in both images. Then, the triplets were constructed to be finally ranked and selected based on the best geometric similarities. Finally, the proposed method was compared with state-of-the-art keypoint refinement algorithms such as random sample consensus (RANSAC) [6], [7], SIFT [8], and symmetric SIFT (SSIFT) [9]. The results show a competitive performance against the proposed algorithm. Nevertheless, the proposed method considers only the last matching phase regardless of the transformation type.

Matching an affine-invariant region using triplets is not the moment technique. For instance, studies in [10] and [11] have developed keypoint triangular regions to implement descriptors of the internal features. First, they transformed the triangular region into an alternative region, such as an ellipse, circle, or parallelogram. They then nominated one of the well-known algorithms, SIFT, BRISK, or oriented fast and rotated BRIEF (ORB) (BRIEF: fast robust local feature detector), to describe the features inside the transformed triangle [12].

Researchers have developed various keypoint descriptors to match aerial and binary images and construct multimodal biomedical images. For example, the authors in [13] developed a binary descriptor based on the Harris detector and matched keypoints using the SIFT and ORB algorithms to estimate the satellite pose. The advantage of such a descriptor is the simplicity of its implementation, with less memory. In addition, [14] proposed a biological descriptor based on training and testing datasets to reduce the complexity of real-time applications. Finally, [15] adopted conventional descriptors for binary images to be preprocessed for noise removal, after which the edges were simple to find.

Consequently, the keypoints are apparent and easy to detect. Multimodality imaging is another application of keypoint matching for registration [16]. The study in [17] implemented a descriptor for multimodal images based on the MFI transform to extract features and match them with the Fréchet distance metric. In addition, in [18], the author developed a nonlinear SIFT descriptor to register nonmedical images. Research [19]–[21] has discussed comparative studies between different implementations of keypoint-based descriptors for image registration objectives.

The mentioned descriptors are not oriented toward resolving affine transformation issues. However, many attempts have been made to preserve rotation, translation, and scaling. The study in [22] used a wavelet transform, namely, the Harr transform, to obtain different scales and keep the rotation and translation invariant. The performance of such descriptors is comparable to that of SIFT, and the computational process is less complex. In [23], the authors considered pixels as patterns and stored their information as rotationally invariant. Statistical pixel operations were then applied to resolve the different transformation problems. The authors of [24] developed a new type of differentiable SIFT descriptor based on higher-order derivatives to create a higher-order scale. The advantage of this method is the extraction of additional
features using exhaustive computation. In addition, [25] suggested corner edge detection, where the contourlet transform was used to detect the edges with vector manipulation to determine the corners in images of different scales. The KAZE descriptor [26] was developed to overcome the problem of averaging noise at the boundaries owing to the Gaussian scale space.

The affine region can be detected and matched using different algorithms based on edge, Harris, and color intensity. In [27], different algorithms were compared to solve affine transformation using the aforementioned simple techniques. Later, many papers, such as [28]–[30], investigated the efficacy of some descriptors based on SIFT, BRISK, or ORB, at different forms of image affine transformation. However, studies have shown that these descriptors are suitable for some affine transformations but are not optimally accurate. By contrast, our current work is dedicated to improving affine-transformed image matching based on proposed representations and descriptor-less matching algorithms. We hypothesize that the proposed methodology improves the correspondence of geometrically transformed regions with competitive performance compared with other approaches.

III. THE PROPOSED REPRESENTATIONS

This study proposes representation methods based on triangular regions with vertices constituted by the detected keypoints. These representations transform the triangular regions into rectangular matrices. Fig. 1 shows the general procedure for obtaining rectangular matrices from the triangular regions.

For a given image I, first, the keypoints are detected by a keypoint detector similar to the non-sampled contourlet transform (NSCT) keypoint detector [31]. Then, only the most distinctive points are selected for further processing based on the detector’s response magnitude. These keypoints are categorized into two types; anchor and neighbor keypoints. Next, triangular regions are formed for each anchor keypoint by connecting it to two neighboring points within a particular window. After that, each triangle is compared to its regional triangles for redundancy removal. Triangular regions are then transformed into rectangular matrices, subsequently used in the image matching algorithm.

Triangular regions are represented in two ways in this paper. The first is a full triangle representation, in which all points encompassed by the formed triangle are considered in the transformation. The other representation only considers the triangle’s essential lines: the medians and the sides of the triangular region. Both representations are described as follows:

A. FULL TRIANGLE REPRESENTATION (FTR)

Image-matching researchers have long been interested in obtaining affine-invariant regions between images that have undergone a geometrical transformation. With three noncollinear vertices, the triangle is one of the geometric forms that can achieve that invariance. Therefore, this paper establishes triangular image regions with vertices formed from the extracted keypoints in each image. This formation of regions helps the matching algorithm recognize the affine-invariant regions across images that have suffered an affine transformation. For instance, as shown in Fig. 2, a triangle is formed with three keypoints that form its vertices (P0, P1, and P2), where P0 is an anchor keypoint, and P1 and P2 are neighboring keypoints. However, extracting and dealing with this triangular region has some difficulties arising from its non-quadrangular shape because images are defined and represented by rectangular forms. Therefore, transforming the image region from a triangular to rectangular form is required for the later processing and matching stages.

All triangular region pixels are mapped to a rectangular matrix in this representation. Thus, the triangle is scanned anticlockwise using lines starting from the P0 – P1 side and ending at the P0 – P2 side, as shown in Fig. 2. Typically, these lines are not equal in length and may produce irregular quadruple shapes. Therefore, all lines are sampled for a fixed number of points (N) to solve this problem. Thus, a matrix of \((M \times N)\) is produced using \(M\) lines with \(N\) samples each, representing a triangular region. In general, the length of the triangle side \((P1 – P2)\) may be used to compute \(M\), whereas the lengths of the two sides \((P0 – P1\) and \(P0 – P2\)) can be used to determine \(N\) for a given triangle. Nevertheless, to avoid the
difficulty of dealing with variable matrices sizes, the values of M and N should be unified for multiple triangular regions.

Ideally, the produced rectangular representation inherits the affine-invariant properties of triangular regions. Therefore, it is straightforward to compare and identify two similar triangular regions for images with affine transformation variations by using this representation. However, other issues, such as illumination variations, quantization, keypoint localization, and resampling may alter this comparison.

B. TRIANGLE MEDIANS AND SIDES REPRESENTATION (TMSR)

Rather than considering all triangle points, as in the previous representation, only six triangle lines are used and converted into a rectangular matrix in this representation. These lines represent the three sides of the triangle and its three medians, as shown in Fig. 3. The triangle in Fig. 3 is transformed into a rectangular matrix by mapping the points along the lines (P₀ − P₁, P₀ − P₂, P₁ − P₂, P₀ − P₀m₀, P₁ − P₁m₁, and P₂ − P₂m₂). P₀ is the anchor keypoint; P₁ and P₂ are neighboring keypoints; and P₀m₀, P₁m₁, and P₂m₂ are the side midpoints of the triangle (P₀P₁P₂). Unfortunately, these six lines typically do not have equal lengths (points), making it challenging to handle this region and compare it with the other regions. This issue can be solved by sampling all the lines with a fixed number of samples (N). Hence, this representation produces a rectangular matrix (6xN) by taking a fixed number of samples (N) for the mentioned triangular region’s six lines.

IV. DESCRIPTOR-LESS IMAGE MATCHING

In order to demonstrate the robustness of the two introduced representations, a descriptor-less matching algorithm is used with these representations. Instead of utilizing a keypoint descriptor, a distance measure is directly applied to the rectangular matrices to measure the similarity between the keypoints across two images. So, this method represents an image-matching scheme that depends on the introduced representations regardless of any descriptor calculation. The operational steps of this matching scheme are illustrated in Fig. 4 and detailed afterward.

For two images, I₁ and I₂, the matching starts by producing the rectangular matrices either by FTR or TMSR representations, as shown in Fig. 4, using the following steps:

1. After detecting keypoints in both images, triangles in the support regions around the anchor keypoints with angles greater than 35° and sides with lengths more than five pixels are considered for the representations.

2. Although no general preprocessing method is applied to the input images, local normalization is applied separately for each rectangular matrix. The normalization process is performed by subtracting the matrix average and dividing it by the (maximum-minimum) range value.

3. A proximity matrix is generated after (level 1) matching all matrices for the anchor keypoints across the two images. The average of the three best-matched rectangular matrices of the anchor keypoints across the two images determines the proximity matrix values.

4. Subsequently, (level 2) matching is performed based on the computed proximity matrix by adopting the assignment problem algorithm [32].

5. Finally, the RANSAC method excludes outliers, refines the matching, and produces the correspondences.

V. VALIDATION AND RESULTS

A. DATASET

Six groups [33], each comprising six images, are used in the experiment to test the performance of the proposed representation. Fig. 5 shows these images with their groups, which have various degrees of viewpoint changes compared
to their reference images. All images are, first, converted to gray levels before any computations. All tested groups have images with the exact pixel resolution as the reference image, except “Bird” and “Bricks”.

**B. KEYPONTS DETECTION AND SELECTION**

Keypoints are detected using an NSCT-based method, as previously stated. This technique is appropriate because it generates dense keypoints and has other desirable properties, such as high repeatability compared to traditional algorithms [31].

A significant number of keypoints in each support region is needed for the proposed representations. However, the computation time of the representation will be significantly influenced by the number of keypoints included in calculations. Consequently, only a set of the strongest keypoints is defined as neighbors, and then a subset of these neighbors is selected as primary (anchor) keypoints. Thus, the anchor keypoints in an image are a subset of the neighbor keypoints. The strongest keypoints are selected based on the highest values of the metrics used in the detector. In the following experiments, the neighbors in the reference images are set to 3000 keypoints, whereas the anchors are set to 1500 keypoints. For the query image, these numbers are weighted by the ratio of total keypoints in the query image to total keypoints in the reference image. Fig. 6 shows samples of images in each image group with the detected keypoints. The blue diamonds in Fig. 6 indicate the anchor keypoints, whereas the remaining keypoints marked with yellow stars are additional neighboring keypoints. Anchor keypoints are also considered neighbors for other adjacent anchor keypoints within an annulus-shaped window with an inner radius of 5 pixels and an outer radius of 48 pixels.

**C. EVALUATION OF THE PROPOSED REPRESENTATIONS**

The proposed representations are validated through a comparative study of the keypoints that matched the six groups, as shown in Fig. 5. First, the representations are compared to validate and analyze their properties and performance in various image groups. Subsequently, a comparison with the conventional keypoint-based methods is conducted.

The representations in this work, FTR and TMSR, are evaluated in the first experiment, and their outcomes are compared using the matching method specified in Fig. 4 with...
FIGURE 6. Examples of detected keypoints for pair of images (1 & 4) in each group. For each image, the blue diamond represents an anchor keypoint, while the yellow star is a neighbor keypoint.

FIGURE 7. Examples of triangular regions with (a) nearby (adjacent) centroid coordinates and equivalent triangle areas (b) different centroid coordinates but similar triangle areas (c) similar centroid coordinates with different areas (d) different centroid coordinates and triangle areas.

the dataset depicted in Fig. 5. In order to facilitate calculations, all matrices created by the FTR are configured to contain eight lines with eight samples per line (i.e., M = 8 and N = 8), yielding 8 × 8 matrices. However, due to the computation technique used for TMSR, the number of lines is always 6, with eight samples per line (N = 8) producing 6 × 8 matrices for the TMSR representation. Lower sampling values may speed up the calculations but could produce more significant quantitation errors. On the other hand, selecting a higher sampling value enhances the performance of both representations at the expense of computational complexity. The correct keypoints match the reference, and the query images are obtained when the correspondences between the images are within a circle of three pixels after rectifying the reference keypoints using the provided homography matrices found with the dataset.

Calculating the two representations for every potential keypoint can be time-consuming, mainly if many of these keypoints are detected in the image. Therefore, a filtration criterion is necessary to lower the overall calculations cost of representation and matching operations. The primary goal is to reduce the number of triangles rather than the number of nearby keypoints, resulting in fewer matrices for each anchor keypoint. This filtration is accomplished based on the values of the areas of the triangles and their centroid coordinates.

For triangles A and B, formed by the same anchor keypoint, as shown in Fig. 7, the centroid-based distance (d1) and distance (d2) based on the values of the triangular areas are computed using the following equations:

\[
d_1 = \frac{d(P_{ca}, P_{cb})}{\text{area(Triangle } B)}
\]

\[
d_2 = \frac{|\text{area (Triangle } A) - \text{area (Triangle } B)|}{\text{area(Triangle } B)}
\]

where \(P_{ca}\) and \(P_{cb}\) are the centroids of triangles A and B, respectively, and \(d(P_{ca}, P_{cb})\) is the Euclidean distance between these two centroids. The area of triangle B weighs distances \(d_1\) and \(d_2\) to ensure invariance of the two measures to scale the change between the images. Filtration is performed by skipping triangle B from the representation process if both measures \(d_1\) and \(d_2\) (in (1) and (2)) are less than the predefined threshold. For example, in Fig. 7, only triangle B in (a) is discarded from further processing among the four cases in that figure. For the dataset used in this study, the average area of the triangles attached to the anchor keypoints is approximately 500, with the numbers of keypoints and support region sizes defined in the previous section. Therefore, for small centroid distances (e.g., < 3), the threshold should be less than \(3/500 = 0.006\). Accordingly, we chose a threshold value of 0.005 for all experiments in this study. It is worth noting that there is no optimal choice
for determining the best threshold value since this is a trade-off process. The higher the threshold value, the faster the representation and matching with lower performance and vice versa.

The performance of our proposed algorithm is measured using the F1 score, which combines the precision and recall measures into one measure, defined as:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \%$$  

(3)

where precision measures the correctly detected matches to all detected matches, and recall is the percentage of correct matches to all possible matches between images. Fig. 8 shows the F1 score for both representations, FTR and TMSR, applied to the adopted dataset using the matching algorithm presented in Fig. 4. Although the two representations are calculated differently, the performance of the lighter representation, TMSR, is similar to that of FTR in most cases.

To further investigate the distinctions between the two representations, the difference of the F1-score between every two implementations is computed. The difference of F1-scores is given by

$$dF_1(a, b) = F_1^b - F_1^a$$  

(4)

where $F_1^x$ is the F1 score computed using (3) for method $x$. Fig. 9 shows the F1 score differences ($dF_1$) between the FTR- and TMSR-based matching methods.

Although both methods are computed differently, they behave competitively in “Bird”, “Home”, and “Machines” groups. The TMSR is slightly better in images that include structures and mainly have rotation or scale variations or both, such as image pairs 1&2 in “Bird” and “Home” groups and image pairs 1&6 in “Home” and “Machines” groups. However, FTR representation dominates in the remaining groups: “Abstract”, “Bricks”, and “Busstop”. This result is due to the images’ repeated patterns and fine details. Also, the nature of the FTR computation, which considers all pixels in the triangular regions, contributes to that result. The average F1-score differences ($dF_1$ in (4)) per image pair are shown in Fig. 9, which illustrates the overall competency between the two representations.

D. COMPARISON WITH CONVENTIONAL APPROACHES

A comparative study with well-known algorithms, such as SIFT, SURF, BRISK, and KAZE, is conducted to validate the representations presented for image matching. Although different implementations of these methods exist, MATLAB built-in implementations are adopted here with their default values. The comparative algorithms are descriptor-based methods; however, they are de facto algorithms that judge similarities based on keypoints independent of the proposed descriptor-free algorithm. Our study targeted images with viewpoint changes, affine, and perspective transformations. Therefore, we chose the same six image groups shown in Fig. 5 to accomplish this task. The matching performance using such images for all the tested algorithms is evaluated using the F1-score measure in (3), where the reference image
FIGURE 10. Examples of correct correspondences (blue lines) and matching scores (in parentheses) comparisons for image pairs (1&5) for the SURF, SIFT, KAZE, BRISK, FTR-based and TMSR-based matching algorithms.

FIGURE 11. Examples of correct correspondences (blue lines) and matching scores (in parentheses) comparisons for image pairs (1&6) for the SURF, SIFT, KAZE, BRISK, and FTR-based and TMSR-based matching algorithms.

“1” is matched to the other five images “2 to 6” in each group.

Examples of the matching output for the four conventional methods compared to the presented FTR-based and
TMSR-based methods are shown in Fig. 10 for image pairs 1 and 5, and Fig. 11 for image pairs 1 and 6 among all tested groups. Blue lines indicate the correct correspondences between the reference and query images in each group, and the correct matching scores are indicated in parentheses below each matching pair. It is evident from these figures that the FTR and TMSR representations using the proposed matching method have intense and superior correct matches compared with the other methods tested. In addition, for most image pairs, the differences between the proposed methods are minor and interchangeable among the test image groups.

The evaluation of the proposed representations using the descriptor-less matching algorithm against the SURF, SIFT, KAZE, and BRISK methods is illustrated in Fig. 12. Except
for the BRISK method, the descriptor-based algorithms tested had a competitive performance, with our algorithm offered only in low-geometrical-transformation images. This performance can be seen in the F1 score values in Fig. 12 for the “Abstract” group with the SURF method and the image pairs 1 & 2, and 1 & 3 for the SIFT and KAZE methods, respectively. Further examples are image pairs 1&2 in “Bird” for SURF and SIFT and pair 1&2 in “Bricks” for SURF and KAZE methods. Furthermore, in the “Home” and “Machines” groups, these three methods perform better for image pairs 1&6 than other image pairs in the same groups. Indeed, the relative geometric transformation between pairs 1 & 6 in both groups showed the lowest variation among all the pairs. Although other algorithms can tolerate different degrees of rotational transformation, particularly the SURF descriptor, the chart verified the performance of the proposed algorithm. In addition, the average matching curve shows that our applied methodology and its extended modes achieved the highest percentage of image matching.

All the image groups showed different degrees of viewpoint change and were mainly affected by perspective deformation. The experimental results for the FTR and TMSR representations indicate that these methods are slightly affected by a moderate degree of transformation. The F1-scores and their averages, shown in Fig. 12 plots, for the FTR -and TMSR-based algorithms are higher than the others in all tested groups, although SURF, SIFT, and KAZE competed with these methods in a few cases, as described above. However, the matching rate of BRISK is the lowest among the groups, and this method is not robust when the perspective of the image changed. The worse performance of conventional descriptors is represented in the image groups “Bird”, “Bricks”, “Home”, and “Machines” for images that have a higher degree of perspective changes relative to their reference images.

VI. CONCLUSION AND FUTURE WORK

Two representations based on triplets of keypoints with significant invariance to viewpoint changes were presented. The proposed algorithms use anchor keypoints with two neighboring points to form triangular regions in the reference and query images and then transform them into rectangular matrices. In the first method, a full triangle represents the FTR, where the anchor point is connected to all the possible points between the complementary vertices of the triangle. In the second method, triangle medians and side representation TMSR, only the intensity of the region along the six standard lines, triangle medians, and sides converted into rectangular matrices. Consequently, the latter representation is lighter regarding the number of pixels involved in the triangular region and rows for the produced rectangular matrices.

Furthermore, a descriptor-less matching algorithm is implemented based on these representations for viewpoint change detection. Such an algorithm shows that the two representations can reliably handle matching images that undergo geometrical transformation problems, such as affine and weak perspective transformations. The performance was further compared with other conventional descriptor-based algorithms: SURF, SIFT, KAZE, and BRISK. The results showed that the proposed representation and matching algorithm competitively outperformed conventional methods.

Such representations can be used for geometrical invariant image matching applications with no descriptor usage is required. Although the presented representations, FTR and TMSR, exhibit high performance in matching images with moderate viewpoint change transformations, we believe there is room for improvement. The points of improvement include the following. First, the filtration process of the neighboring keypoints introduced herein can be optimized to accelerate the algorithm with a high matching rate. In addition, the image content details may deteriorate in triangular regions because of large-scale changes between images, which affect the representations and rectangular matrices. Therefore, an extension to the triangular regions for this type of transformation can solve this problem.

REFERENCES

[1] G. Lv, “A novel correspondence selection technique for affine rigid image registration,” IEEE Access, vol. 6, pp. 32023–32034, 2018.
[2] X. Jiang, J. Ma, G. Xiao, Z. Shao, and X. Guo, “A review of multimodal image matching: Methods and applications,” Inf. Fusion, vol. 73, pp. 22–71, Sep. 2021.
[3] X. Guo and X. Cao, “Good match exploration using triangle constraint,” Pattern Recognit. Lett., vol. 33, no. 7, pp. 872–881, 2012.
[4] Y. Li and R. Stevenson, “Multimodal image registration by iteratively searching keypoint correspondences,” in Visual Information Processing and Communication IV, vol. 8666. Bellingham, WA, USA: SPIE, 2013, Art. no. 86660E.
[5] Y. Li and R. Stevenson, “Incorporating global information in feature-based multimodal image registration,” J. Electron. Imag., vol. 23, no. 2, Apr. 2014, Art. no. 020313.
[6] H. Wang, D. Omea, J. Verbeek, and C. Schmid, “A robust and efficient video representation for action recognition,” Int. J. Comput. Vis., vol. 119, no. 3, pp. 219–238, 2016.
[7] O. Chum and J. Matas, “Matching with PROSAC—Progressive sample consensus,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1, Jun. 2005, pp. 220–226.
[8] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.
[9] J. Chen and J. Tian, “Real-time multi-modal rigid registration based on a novel symmetric-SIFT descriptor,” Proc. Natl. Sci., vol. 19, no. 5, pp. 643–651, May 2009.
[10] Y. Kanarazawa and K. Uemura, “Wide baseline matching using triplet vector descriptor,” in Proc. Brit. Mach. Vis. Conf., 2006, pp. 267–276.
[11] K. Li, J. Yao, M. Xia, and L. Li, “Joint point and line segment matching on wide-baseline stereo images,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2016, pp. 1–9.
[12] X. Zhao, Z. He, and S. Zhang, “Improved keypoint descriptors based on Delaunay triangulation for image matching,” Optik, vol. 125, no. 13, pp. 3121–3123, Jul. 2014.
[13] Y. Huang, Z. Zhang, H. Cui, and L. Zhang, “A low-dimensional binary-based descriptor for unknown satellite relative pose estimation,” Acta Astronautica, vol. 181, pp. 427–438, Apr. 2021.
[14] M. Saleiro, K. Terzić, J. M. F. Rodrigues, and J. M. H. D. Buf, “BINK: Biological binary keypoint descriptor,” Biosystems, vol. 162, pp. 147–156, Dec. 2017.
[15] H. Chatbri, K. Kameyama, P. Kwan, S. Little, and N. E. O’Connor, “A novel shape descriptor based on salient keypoints detection for binary image matching and retrieval,” Multimedia Tools Appl., vol. 77, no. 21, pp. 28925–28948, Nov. 2018.
[16] Y. Li, J. Zou, J. Jing, H. Jin, and H. Yu, “Establish keypoint matches on multispectral images utilizing descriptor and global information over entire image,” Infr. Phys. Technol., vol. 76, pp. 1–10, May 2016.

[17] G. Xu, Q. Wu, Y. Cheng, F. Yan, Z. Li, and Q. Yu, “A robust deformed image matching method for multi-source image matching,” Infr. Phys. Technol., vol. 115, Jun. 2021, Art. no. 103691.

[18] Q. Yu, D. Ni, Y. Jiang, Y. Yan, J. An, and T. Sun, “Universal SAR and optical image registration via a novel SIFT framework based on nonlinear diffusion and a polar spatial-frequency descriptor,” ISPRS J. Photogramm. Remote Sens., vol. 171, pp. 1–17, Jan. 2021.

[19] A. J. Malekabadi, M. Khoshtaphepour, and B. Emadi, “A comparative evaluation of combined feature detectors and descriptors in different color spaces for stereo image matching of tree,” Scientia Horticulturae, vol. 228, pp. 187–195, Jan. 2018.

[20] D. Unay and S. G. Stanciu, “An evaluation on the robustness of five popular keypoint descriptors to image modifications specific to laser scanning microscopy,” IEEE Access, vol. 6, pp. 40154–40164, 2018.

[21] S. A. K. Tareen and Z. Saleem, “A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK,” in Proc. Int. Conf. Comput., Math. Eng. Technol. (iCoMET), Oct. 2016, pp. 187–195, Jan. 2018.

[22] C. Barajas-Garcia, S. Solorza-Calderón, and E. Gutiérrez-López, “Scale, translation and rotation invariant wavelet local feature descriptor,” Appl. Math. Comput., vol. 363, Dec. 2019, Art. no. 124594.

[23] T. Song, F. Meng, Q. Wu, B. Luo, T. Zhang, and Y. Xu, “L2SSP: Robust keypoint description using local second-order statistics with soft-pooling,” Neurocomputing, vol. 230, pp. 230–242, Mar. 2017.

[24] U. Park, J. Park, and A. K. Jain, “Robust keypoint detection using higher-order scale space derivatives: Application to image retrieval,” IEEE Signal Process. Lett., vol. 21, no. 8, pp. 962–965, Aug. 2014.

[25] B. Li, H. Li, and U. Soderstrom, “Scale-invariant corner keypoints,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2014, pp. 5741–5745.

[26] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, “KAZE features,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Florence, Italy, Oct. 2012.

[27] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, “A comparison of affine region detectors,” Int. J. Comput. Vis., vol. 65, no. 1, pp. 43–72, 2005.

[28] S. Isik, “A comparative evaluation of well-known feature detectors and descriptors,” Int. J. Appl. Math., Electron. Comput., vol. 3, no. 1, pp. 1–6, 2014.

[29] S. A. K. Tareen and Z. Saleem, “A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK,” in Proc. Int. Conf. Comput., Math. Eng. Technol. (iCoMET), Mar. 2018, pp. 1–10.

[30] L. Peng, “Efficient matching of robust features for embedded SLAM,” M.S. thesis, Inst. Parallel Distrib. Syst., Univ. Stuttgart, Stuttgart, Germany, 2012.

[31] S. R. Saydam, I. A. E. Rube’, and A. A. Shoukry, “Contourlet based interest points detector,” in Proc. 20th IEEE Int. Conf. Tools Artif. Intell., Dayton, OH, USA, Nov. 2008, pp. 509–513.

[32] I. S. Duff and J. Koster, “On algorithms for permuting large entries to the diagonal of a sparse matrix,” SIAM J. Matrix Anal. Appl., vol. 22, no. 4, pp. 973–996, 2000.

[33] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, “HPatches: A benchmark and evaluation of handcrafted and learned local descriptors,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5173–5182.

IBRAHIM EL RUBE’ (Senior Member, IEEE) received the B.Eng. degree from the Department of Computer and Electronics Engineering, in 1992, the M.Sc. degree in electronics and communications engineering from the Arab Academy for Science and Technology, Egypt, in 1999, and the Ph.D. degree in systems design engineering from the University of Waterloo, Canada, in 2006. He has held lecturing positions with the Department of Electronics and Communications Engineering, Arab Academy for Science and Technology, from 1993 to 2011. In August 2006, he was an Assistant Professor and became an Associate Professor, in 2010. In 2011, he joined the Computer Engineering Department, Taif University, where he is currently an Associate Professor. He was the Department Chair of Taif University, from 2012 to 2016. His current research interests include signal and image processing, computer vision, pattern recognition, and machine learning.

SAMEER ALSHARIF received the bachelor's degree in computer engineering from Umm Alqura University, Saudi Arabia, in 2006, and the master's and Ph.D. degrees in systems and control engineering from Case Western Reserve University, USA, in 2013 and 2017, respectively. In 2007, he worked as a Teaching Assistant at Taif University, Saudi Arabia, where he is currently working as an Associate Professor and the Chair of the Computer Engineering Department. In 2008, he was awarded a Full Scholarship from the Saudi Government to study abroad. He is involved in many research projects, including the IoT, renewable energy, control systems, image processing, and machine learning.

* * *