

Article

A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching

Surendra Kumar Sharma ^{1,2}, Kamal Jain ² and Anoop Kumar Shukla ^{3,*}¹ Indian Institute of Remote Sensing, Dehradun 248001, India² Indian Institute of Technology Roorkee, Roorkee 247667, India³ Manipal School of Architecture and Planning, Manipal Academy of Higher Education, Manipal 576104, India* Correspondence: anoop.shukla@manipal.edu

Abstract: Image stitching is a technique that is often employed in image processing and computer vision applications. The feature points in an image provide a significant amount of key information. Image stitching requires accurate extraction of these features since it may decrease misalignment flaws in the final stitched image. In recent years, a variety of feature detectors and descriptors that may be utilized for image stitching have been presented. However, the computational cost and correctness of feature matching restrict the utilization of these techniques. To date, no work compared feature detectors and descriptors for image stitching applications, i.e., no one has considered the effect of detectors and descriptors on the generated final stitched image. This paper presents a detailed comparative analysis of commonly used feature detectors and descriptors proposed previously. This study gives various contributions to the development of a general comparison of feature detectors and descriptors for image stitching applications. These detectors and descriptors are compared in terms of number of matched points, time taken and quality of stitched image. After analyzing the obtained results, it was observed that the combination of AKAZE with AKAZE can be preferable almost in all possible situations.

Keywords: image stitching; feature detector; feature descriptor; image mosaic; image alignment; image quality assessment



Citation: Sharma, S.K.; Jain, K.; Shukla, A.K. A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching. *Appl. Sci.* **2023**, *13*, 6015. <https://doi.org/10.3390/app13106015>

Academic Editor: Yu-Dong Zhang

Received: 12 February 2023

Revised: 8 May 2023

Accepted: 12 May 2023

Published: 13 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Panoramic images, also known as panoramas, are wide-angle photographs with a horizontal field of view (FOV) of up to 360 degrees. Panoramas can be created by merging numerous photos using an image stitching technique. It is a widely used technique of increasing the camera's FOV in a constructive manner. Real-time image stitching has been a challenging area for image processing professionals in recent decades. Image stitching techniques have been employed in various applications: aerial and satellite image mosaics [1,2], SAR image mosaicking [3,4], video frame mosaicking [5], augmented reality [6], real state and many more. Image stitching methods contain two main steps: image alignment and image blending. The advancement of image stitching technology is often dependent on the advancement of these two components. Image alignment is being used to determine motion relationships by identifying and matching feature points across images. It has a direct link to the speed and success rate of the image stitching procedure. Integral aspects of image alignment are feature detection, feature description and feature matching. In an image, a feature refers to a certain meaningful pattern. It can be a pixel, edge or contour, and it can also be an object. The procedure of finding these meaningful patterns in an image is called feature detection. The feature detector generates a number of distinct points in an image, known as key-points or feature points. The feature detection algorithm selects these locations depending upon their tolerance to noise and other deformations. Many feature detectors have their own feature descriptors for detected

feature points. A feature descriptor is a vector of values that represents the image patch surrounding a feature point. It might be as basic as the raw pixel values or as complex as a histogram of gradient orientations. Various feature detectors and descriptors have evolved over time. However, due to their poor speed and high number of false matches, the majority of them are not useful for image stitching applications. Feature detectors and descriptors are crucial in image alignment; if a good pairing of feature detector and descriptor is not chosen, then the images to be stitched will not be aligned properly, which will add distortions in the resulting stitched image. Thus, a comparative study is required to determine the best combination of feature detector and descriptor that can provide high-quality stitched images with lower computational cost.

In the field of computer vision and image processing, the research on feature detectors and descriptors has grown rapidly in last couple of decades. A brief literature review is presented in the following to show the continuous development in feature detection and description. The first corner detector was proposed by Morevec [7]. The authors of [8] proposed a fast operator for the detection and precise location of distinct points, corners and centers of circular image features. Harris and Stephens [9] proposed a new corner detector named the Harris corner detector, which overcomes the limitations of the Moravec detector. Tomasi and Kanade [10] developed Kanade–Lucas–Tomasi (KLT) tracker based on the Harris detector for the detection and tracking of features across images.

A comparison of the Harris corner detector with other detectors was carried out by [11,12]. Shi and Tomasi [13] presented an improved version of the Harris corner detector named Good Features to Track (GFTT). Lindeberg [14] presented a junction detector with auto-scale selection. It is a two-stage process that begins with detection at coarse scales and progresses to localization at finer scales.

Wang and Brady [15] presented a fast corner detection approach based on cornerness measurement of total curvature. The authors of [16] proposed a new approach for feature detection, named SUSAN; however, it is not rotation- or scale-invariant. Hall et al. [17] presented a definition of saliency for scale change and evaluated the Harris corner detector, the method presented in [18] and the Harris–Laplacian corner detector [19]. Harris–Laplacian is a combination of the Harris and Laplacian methods which can detect feature points in an image, each with their own characteristic and scale. The authors of [20] compared various popular detectors. Later, Schmid et al. [21] modified the comparison procedure and conducted several qualitative tests. These tests showed that the Harris detector outperforms all other algorithms. Lowe [22] presented the scale-invariant feature transform (SIFT) algorithm, which is a combination of both detector and descriptor. SIFT is invariant to scaling, orientation and illumination changes. Brown and Lowe [23,24] used SIFT for automatic construction of panoramas. They used the SIFT detector for feature detection and description. Multi-band blending was used to smoothly blend the images. Image stitching techniques based on SIFT are well suited for stitching high-resolution images with a number of transformations (rotation, scale, affine and so on); however, they suffer from image mismatch and are computationally costly. The authors of [25] combined the Harris corner detector with the SIFT descriptor with the aim to address scaling, rotation and variations in lighting condition issues across the images. The authors of [26] tried to accelerate the process and reduced the distortions by modifying the traditional algorithm. The center image was selected as a reference image, then the following image in the sequence was taken based on a set of statistical matching points between adjacent images. The authors of [27] compared different descriptors with SIFT, and the results proved that SIFT is one of the best detectors based on the robustness of its descriptor. Zuliani et al. [28] presented a mathematical comparison of various popular approaches with the Harris corner detector and KLT tracker. Several researchers examined and refined the Harris corner detector and SIFT [29–31]. A comparative study of SIFT and Its variants was carried out by [32]. Mikolajczyk and Schmid [33] proposed a novel approach for detecting interest points invariant to scale and affine transformations. It uses the Harris interest point detector to build a multi-scale representation and then identifies points where a local measure (the Laplacian)

is maximum across scales. Mikolajczyk [34] compared the performance of different affine covariant region detectors under varying imaging conditions.

The authors of [35] evaluated Harris, Hessian and difference of Gaussian (DoG) filters on images of 3D objects with different viewpoints, lighting variations and changes of scale. Matas et al. [36] proposed the Maximally Stable Extremal Regions (MSER) detector, which detects blobs in images. Nistér and Stewénius [37] provided its faster implementation. Bay et al. [38] proposed SURF, which is both a detector and descriptor. It is based on a Hessian matrix and is similar to SIFT. It reduces execution time at the cost of accuracy as compared to SIFT. The authors of [39] proposed a phase correlation and SURF-based automated image stitching method. First, phase correlation is utilized to determine the overlapping and the connection between two images. Then, in the overlapping areas, SURF is used to extract and match features. The authors of [40] used SURF features for image stitching. They attempted to speed up the image stitching process by removing non-key feature points and then utilized the RELIEF-F algorithm to reduce dimension and simplify the SURF descriptor. Yang et al. [41] proposed an image stitching approach based on SURF line segments, with the goal of achieving robustness to scaling, rotation, change in light and substantial affine distortion across images in a panorama series. For real-time corner detection, the authors of [42] proposed a new detector named the Features from Accelerated Segment Test (FAST) corner detector. The authors of [43] surveyed different local invariant feature detectors. The authors of [44] provided a good comparison of local feature detectors. Some application-specific evaluations were presented by [45–48]. Agrawal et al. [49] introduced CenSurE, a novel feature detector that outperforms current state-of-the-art feature detectors in image registration and visual odometry. Willis and Sui [50] proposed a fast corner detection algorithm for use on structural images, i.e., images with man-made structures. Calonder et al. [51] proposed the Binary Robust Independent Elementary Features (BRIEF) descriptor, which is very helpful for extracting descriptors from interest points for image matching. The authors of [52] presented a new local image descriptor, DAISY, for wide-baseline stereo applications.

Leutenegger et al. [53] proposed a combination of both detector and descriptor named Binary Robust Invariant Scalable Key-Points (BRISK), which is based on the FAST detector and in combination with the assembly of a bit-string descriptor. The use of the FAST detector reduced its computational time. Rublee et al. [54] presented the Oriented FAST and Rotated BRIEF (ORB) detector and descriptor as a very fast alternative to SIFT and SURF. The authors of [55] presented the Fast Retina Key-Point (FREAK) descriptor based on the retina of the human visual system (HVS). Alcantarilla [56] developed a new feature detector and descriptor named KAZE in which they formed nonlinear scale spaces to detect features in order to improve localization accuracy. The faster version of KAZE was presented by the authors in the year 2013 and named Accelerated-KAZE (AKAZE) [57]. The authors of [58] proposed the AKAZE-based image stitching algorithm. The authors of [59,60] presented a detailed evaluation of image stitching techniques. Lyu et al. [61] surveyed various image/video stitching algorithms. They first discussed the fundamental principles and weaknesses of image/video stitching algorithms and then proposed the potential solutions to overcome those weaknesses. Megha et al. [62] analyzed different image stitching techniques. They reviewed various image stitching algorithms based on the three components of the image stitching process: calibration, registration and blending. Further, they discussed challenges, open issues and future directions of image stitching algorithms.

Various prominent feature detectors and descriptors were analyzed by [63,64] and [65]. However, these studies included a few detectors and descriptors, and they have not considered all combinations of detectors and descriptors. The authors of [66] introduced a new simple descriptor and compared it to BRIEF and SURF. The authors of [67] compared several feature detectors and descriptors and analyzed their performance against various transformations in general and then presented a practical analysis of detectors and descriptors for feature tracking.

Several outstanding feature detectors and descriptors have been reported in the existing literature. Some studies compared detectors and descriptors in general, and some are application-specific, and a few have compared only detectors, while others have compared only descriptors. The number of detectors and descriptors used in all of these studies is limited, and they have not considered all pairs of detectors and descriptors. However, as the introduction of new detectors and descriptors is an evolving process, improved comparative studies are always needed. In this study, we aimed to update the comparative studies in connection with existing literature.

The following statements show a connection of existing studies with the proposed work and the attributes that differentiate our study from previous studies:

1. Several detectors and descriptors were included in this study, apart from those already compared by [63,64,67–69].
2. A variety of prominent measures were employed in our research in accordance with [63]. We also used image quality assessment measures to compare the quality of the stitched image to the input image.
3. We present an appropriate framework on image stitching to analyze the performance of the chosen detector and descriptor combination. Our aim is to present an assessment of all detectors and descriptors for image stitching applications. For future research in image stitching area this framework can assist as a guideline.
4. We present a comparison and performance analysis of detectors and descriptors for different types of datasets for image stitching applications.
5. Finally, we obtained a rank for the detector—descriptor pairs in terms of individual datasets.

This paper compares several feature detectors and descriptors for image stitching applications. It may assist researchers in selecting a suitable detector and descriptor combination for image stitching applications. First, all popular detectors and descriptors are presented. Next, each combination of detector and descriptor is evaluated for stitching the images. For the evaluation, a variety of datasets including various possible scenarios are considered. The outcomes of this work, as presented above, add various new contributions to this domain. These contributions are outlined after a brief overview of earlier studies. The paper is structured as follows: A brief overview of feature detectors and descriptors used in this study is presented in Section 2. Section 3 provides details about the datasets and methodology used in the study. Section 4 presents results and discusses the outcomes. Finally, the paper is concluded in Section 5.

2. Overview of Feature Detectors and Descriptors

We have made an effort to include an extensive variety of feature detectors and descriptors in terms of universality, approach and age.

2.1. Good Features to Track (GFTT)

The authors of [13] made a minor change to the Harris corner detection algorithm that yields better results than the Harris corner detector. The authors claimed that detected corners are more uniformly distributed across the image.

2.2. SIFT

Lowe [22] proposed one of the most commonly used feature detection and description algorithms, named SIFT. SIFT is invariant to scale, rotation, illumination and viewpoint change; therefore, it can also correctly locate key-points in noisy, cluttered and occluded environments. It identifies interest points in two steps. First, a scale space is generated by continuously smoothing the original image using Gaussian filters that ensure scale invariance. Then, the original image is resized to half size, and again smoothing is performed using a Gaussian filter. This process is repeated. In this manner, an image pyramid is formed, with the reference image at the bottom (level 1). Next, it computes the DoG by subtracting two consecutive scales. Second, interest points are identified by examining

the $3 \times 3 \times 3$ neighborhood of any pixel. The image point at which DoG values attain an extrema among all neighbors is considered as a key-point. A description algorithm was also developed for extracting the descriptor of detected key-points. The descriptor is a position-dependent histogram of local image gradient directions in the neighborhood of the key-point.

2.3. MSER

The authors of [36] presented the Maximally Stable Extremal Regions (MSER) algorithm, which is used to detect blobs in images. Ref. [37] presented its updated version, which is computationally faster than the original algorithm. During the procedure, regions are initially recognized using connectivity analysis, and then connected maximum and minimum intensity regions are calculated. Extremal regions are closed by affine or projective transformations as well as changes in illumination. As a result, they are also scale- and rotation-invariant.

2.4. FAST

The FAST corner detector was proposed by [42]. Corner detection is accomplished by first picking a collection of training images then running the FAST algorithm on each image to detect points and then using machine learning to find the optimum detection criterion. The detection criteria are judgements regarding whether or not a pixel is a corner. A decision tree is developed that can accurately categorize all of the corners. After that, the decision tree is transformed into C code, which is subsequently employed as a corner detector. The FAST corner detector is ideal for use in real-time applications, but it is not robust to high degrees of noise.

2.5. SURF

The authors of [38] developed the SURF algorithm, which is a quick and robust approach for feature detection and extraction in the images. The major attraction of the SURF method is its ability to calculate operators fast by applying box filters, enabling its use for real-time applications. For interest point identification, it utilizes Hessian matrix approximation because of its high performance in terms of computing time and accuracy. There are two phases of creating SURF description. The first stage is to establish a repeatable orientation using data from a circular region around the key-point. The SURF descriptor is then extracted from a square area that is aligned to the specified orientation.

2.6. Star Detector (CENSURE)

Agrawal et al. [49] introduced the Center Surround Extremas (CENSURE) detector. It is characterized by two considerations: stability (features invariant to viewpoint changes) and accuracy. First, it uses Hessian–Laplacian method to calculate the maxima. Center surround filters known as bi-level filters are used to approximate the Laplacian. Then, a basic center-surround Haar wavelet is constructed by using this Laplacian; this is the CENSURE response. Then, necessary features are provided by a non-maximal suppression.

2.7. BRIEF

Calonder et al. [51] presented the Binary Robust Independent Elementary Features (BRIEF) descriptor, which is based on binary strings. First, binary strings are computed from image patches, then the individual bits are generated by comparing the brightness of pairs of points along the same lines. BRIEF is fast compared to other descriptors such as SIFT and SURF because SIFT and SURF have a high-dimensional descriptor vector, which requires more time and memory to compute. It can be used with the detectors which do not have their own descriptors.

2.8. DAISY

Tola et al. [52] proposed a local image descriptor named DAISY. DAISY gives very good results in wide-baseline situations in comparison to pixel- and correlation-based approaches. It is faster than other descriptors such as SIFT and the Gradient Location and Orientation Histogram (GLOH). SURF is also efficient, but it introduces artifacts that degrade the matching performance. DAISY computes descriptors at every pixel without inserting any artifacts, which improves its matching performance when used densely. Gaussian convolutions are used for descriptor computation. This method computes descriptors in real time. This speed increase is because DAISY uses sums of convolutions instead of weighted sums used by other descriptors. It uses a circularly symmetrical weighting kernel.

2.9. AGAST

The authors of [70] introduced the Adaptive and Generic Accelerated Segment Test (AGAST) detector, which uses specialized decision trees to enhance the performance. It can be used for any type of scene structure. The optimum trees are identified by using the whole binary configuration space. In FAST, the detector must be trained for a specific scene during the training step, but in AGAST, it adjusts to the environment dynamically while processing an image.

2.10. BRISK

The authors of [53] presented a method, Binary Robust Invariant Scalable Key-Points (BRISK), for key-point detection and description. It uses a novel scale space FAST-based detector for detecting key-points. In this algorithm, key-points are discovered in the image scale-space pyramid. Following that, a circular sampling pattern is created in the vicinity of each detected key-point, which calculates brightness comparisons to form a descriptor string.

2.11. ORB

The authors of [54] developed an Oriented FAST and Rotated BRIEF (ORB) detector and descriptor as an efficient alternative to SIFT and SURF. ORB is a combination of the FAST detector and the BRIEF descriptor, with several enhancements to improve the performance. ORB adds a fast and accurate orientation component to the FAST algorithm. It uses an image pyramid to generate multiscale features. For the descriptor, it steers BRIEF according to the orientation of key-points to make it rotation-invariant.

2.12. FREAK

Alahi et al. [55] presented the Fast Retina Key-Point (FREAK) descriptor, which is based on the retina of the HVS. First, a circular sample grid is used to construct a retinal sampling pattern. The density of points is greater around the center. Then, by comparing brightness across a retinal sampling pattern, a cascade of binary strings is produced employing DoG. A "saccadic search", which is a human vision-like search, is used to obtain relevant features. Finally, key-point rotation is achieved by summing local gradients over chosen pairs.

2.13. KAZE

The authors of [56] developed a new feature detection and description method in nonlinear scale spaces. Earlier methods detected and described features at various scales by constructing or approximating an image's Gaussian scale space. However, Gaussian blurring smooths features and noise to the same extent, reducing localization accuracy and distinctiveness. It uses nonlinear diffusion filtering to detect and describe features in a nonlinear scale space. The nonlinear scale space is constructed with the use of efficient Additive Operator Splitting (AOS) methods and variable conductance diffusion. As a result, blurring may be made locally adaptable to the image data, reducing noise and achieving improved localization accuracy and distinctiveness.

2.14. AKAZE

The authors of [57] enhanced the computation cost of KAZE and proposed AKAZE, a faster and multi-scale feature detection and description algorithm. Earlier developed methods for detecting and describing features in nonlinear scale space take more time because constructing the nonlinear scale space is computationally costly. AKAZE employs the Fast Explicit Diffusion (FED) scheme, which is incorporated in a pyramidal structure and speeds up feature detection in nonlinear scale space. By employing FED techniques, nonlinear scale space may be created quicker than any other discretization technique. FED schemes also outperform AOS schemes in terms of ease of implementation and accuracy. AKAZE implements the Modified-Local-Difference Binary (M-LDB) descriptor, which is efficient, scale- and rotation-invariant and uses less memory.

2.15. MSD Detector

Tombari [71] proposed a new detector, Maximal Self-Dissimilarities (MSD), which is able to detect features across multi-modal image pairs. It is based on the intuition that image patches which are very distinct across a reasonably large area of their surroundings are repeatable and distinctive. This idea of contextual self-dissimilarity inverts the essential paradigm of previous successful techniques based on the occurrence of similar rather than different patches. Furthermore, it extended the local self-dissimilarity concept embedded in existing detectors of corner-like interest points to contextual information, resulting in improved repeatability, distinctiveness and localization accuracy.

3. Materials and Methods

To test our framework, we used a variety of datasets (building dataset, tree dataset and classroom dataset) that covered various scene categories with variable degrees of distortion. A subset of input images is depicted in Figure 1.

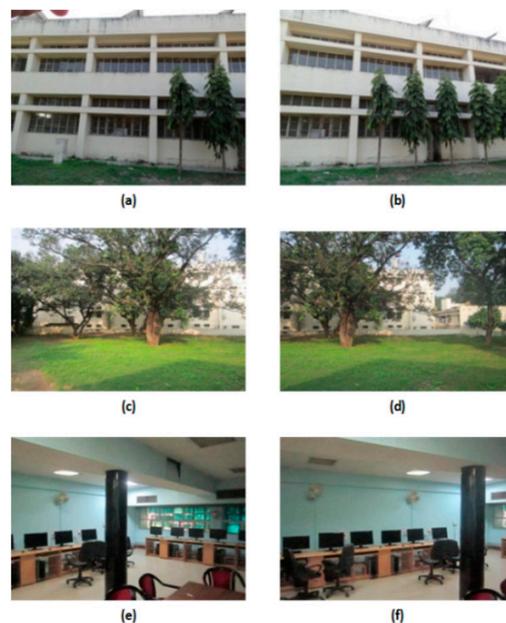


Figure 1. Input datasets used in this study: (a,b) building dataset, (c,d) tree dataset, (e,f) classroom dataset.

3.1. Method: Image Stitching Algorithm

Figure 2 depicts the image stitching algorithm used in this work. Image alignment and image blending are the two major components of this approach. In the image alignment phase, a feature detector is used to detect feature points, followed by a feature descriptor to extract descriptors for the detected feature points. The k-nearest neighbor (k-NN) approach is used to match each feature point. False matches may result from automatic

feature matching, which can lead to poor alignment; consequently, false matches must be eliminated. The M-estimator sample consensus (MSAC) algorithm is employed to eliminate outliers by reducing matching error. The goal is to collect the most feature points acceptable for transformation. The entire procedure is iterative in nature. Correct matches or inliers are those produced by reducing the matching error. The successful identification of a conjugate feature point on the second image for each corresponding feature point on the first image is defined as a correct match. These inlier points are then utilized to calculate the homography (transformation) between two images. The mathematical relationship described in Equation (1) is used to compute the homography matrix.

$$X' = HX, \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

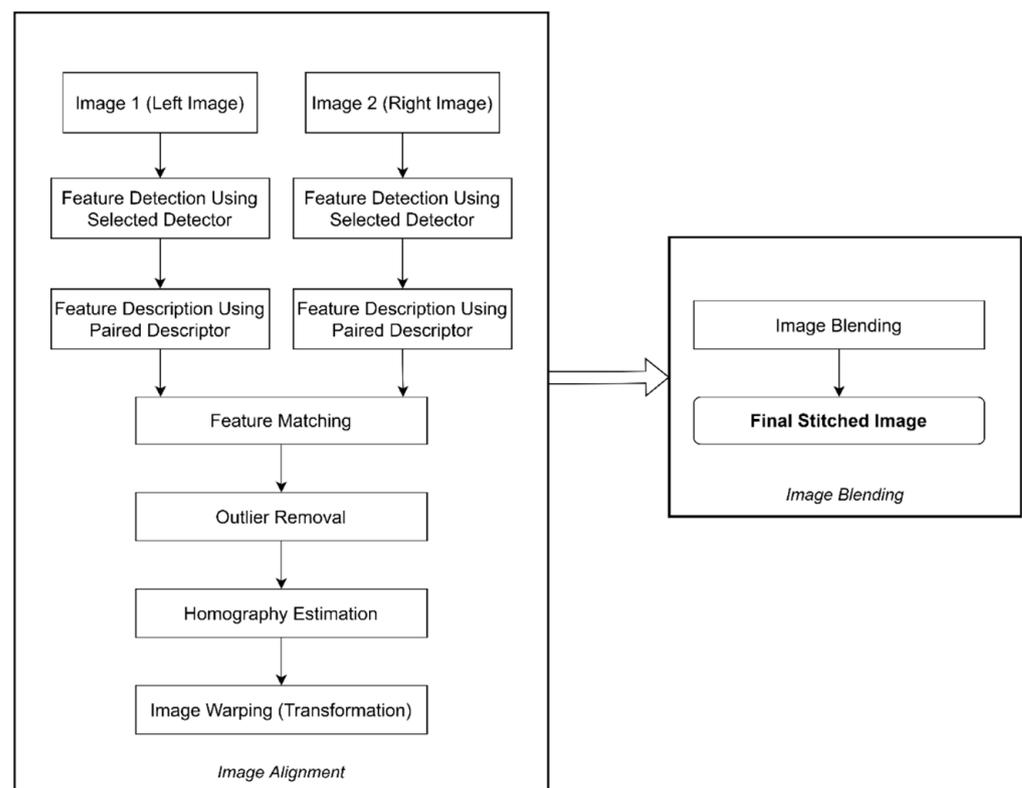


Figure 2. Workflow adopted for image stitching.

Once homography is obtained, the target image is warped (transformed) onto the source image. After the images have been aligned, there may be noticeable seams because of differences in illumination and unintentional camera motions. As a result, image blending is performed to remove the seam.

Table 1 shows feature detectors and descriptors, listed in the order of their development. As shown in the table, some of them are either detectors or descriptors, while others are both. Therefore, to perform a comparative assessment, the best approach is to pair each detector with each descriptor and evaluate its performance. We tested each detector-descriptor combination on a variety of datasets. For each dataset, we use the detector to detect features on the left and right images, and then the paired descriptor was used to extract features. Next, the features were matched using the k-NN matching algorithm. Finally, both the images were stitched using the algorithm described in Figure 2.

Table 1. Feature detectors and descriptors proposed over the years.

Name	Detector or Descriptor	Year Proposed
GFTT	Detector	1994
SIFT	Detector + Descriptor	1999
MSER	Detector	2004
FAST	Detector	2006
SURF	Detector + Descriptor	2006
Star (CENSURE)	Detector	2008
BRIEF	Descriptor	2010
DAISY	Descriptor	2010
AGAST	Detector	2010
BRISK	Detector + Descriptor	2011
ORB	Detector + Descriptor	2011
FREAK	Descriptor	2012
KAZE	Detector + Descriptor	2012
AKAZE	Detector + Descriptor	2013
MSD	Detector	2014

3.2. Method: Comparative Analysis

Different feature detector and descriptor combinations were compared in order to choose the best combination for image stitching applications. Inlier ratio, stitched image quality and execution time are the three parameters used for comparison.

3.2.1. Inlier Ratio

Inlier ratio is the ratio of correct matches to the initial matches. This fraction represents the detector–descriptor combination accuracy. A higher inlier ratio value indicates a better detector–descriptor pair. For some image pairs, the value of the inlier ratio may be zero, which indicates that all the initial matches were wrong. In image stitching, at least four correct matched points are required in order to stitch the images. If some combination has less than four correct matched points, we used an inlier ratio equal to zero. The inlier ratio is defined by Equation (2):

$$\text{InlierRatio} = \frac{\text{numberofcorrectmatches}}{\text{numberofinitialmatches}} \quad (2)$$

The number of accurate matches has an effect on the quality of the final stitched image since the homography matrix is created using correctly matched points, which are then used in the image alignment stage. As a result, the inlier ratio is one of the most significant parameters to consider when choosing the best detector and descriptor combination.

3.2.2. Stitched Image Quality

The subjective and objective techniques are the two most used techniques for evaluating image quality. In the subjective assessment approach, a number of observers are chosen, and then they are presented a set of photographs. Following that, they are asked to judge image quality based on their assessment. Human inspection is used to give a rating to each image depending on its quality [72]. The objective image quality assessment (IQA) approaches rely on automated algorithms to assess image quality without the need for human intervention [72]. Since subjective assessment techniques differ from person to person, small distortions in the final stitched image cannot be detected. As a result, we employed objective IQA techniques to compare the quality of stitched images in this research. The quality of stitched images generated using different pairs of feature detectors and descriptors was compared based on objective IQA metrics. The objective image quality metrics utilized in this study to assess the quality of stitched images are listed below.

Peak Signal to Noise Ratio

Peak Signal to Noise Ratio (PSNR) is a metric for measuring how different two images are. The PSNR of comparable pixel values is expressed as (Equation (3)):

$$\text{PSNR} = \frac{10 \log_{10}(\max(G(i,j), O(i,j)))^2}{\text{MSE}} \quad (3)$$

where $G(i, j)$ and $O(i, j)$ represent the $(i, j)^{\text{th}}$ pixel values in the input image and stitched image, respectively. MSE is the mean square error, which is defined as:

$$\text{MSE} = \frac{\sum_i \sum_j (G(i,j) - O(i,j))^2}{N}$$

where N is the total number of pixels in each image. A lower MSE value indicates that the difference between two images is low, thus resulting in higher PSNR values.

Structural Similarity Index

The SSIM (Structural Similarity) index is used to determine the similarity of two images. The image that is being assessed is known as the target image, and the image that is used to compare the quality of the target image is known as the reference image. The SSIM index meets the requirements of symmetries, boundedness and having a unique maximum. The luminance measurement is assumed to be qualitatively consistent with Weber's law, which specifies that the visible change in luminance in the HVS is roughly proportional to the background luminance. Similarly, the contrast measurement follows the HVS by focusing primarily on the relative difference rather than the absolute difference in the contrast. The final SSIM index is a combination of these two values, as well as a structural similarity component determined using brightness and contrast values [73]. SSIM between two images x and y is described as follows (Equation (4)):

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4)$$

where μ_x and μ_y represent local means for x and y images, respectively. σ_x and σ_y represent standard deviations, and σ_{xy} is the cross-covariance for x and y images. The smaller the difference between the structure of x and y , the higher the SSIM index value. If there is no structural difference, $\text{SSIM}(x, y) = 1$.

Feature Similarity Index

The Feature Similarity (FSIM) index is based on how the HVS interprets an image based on its low-level features [74]. Two characteristics, phase congruency (PC) and gradient magnitude (GM), have to be considered in order to calculate the FSIM index. PC and GM are complementary for describing local quality of the image. The FSIM measurement is divided into two components, $f_1(x)$ and $f_2(x)$, one for PC and one for GM. The measure of similarity in terms of $PC_1(x)$ and is described as (Equation (5)):

$$S_{PC}(x) = \frac{2PC_1(x) \cdot PC_2(x) + T_1}{PC_1^2(x) + PC_2^2(x) + T_1} \quad (5)$$

where T_1 is a positive constant.

The measure of similarity in terms of $G_1(x)$ and $G_2(x)$ is described as (Equation (6)):

$$S_G(x) = \frac{2G_1(x) \cdot G_2(x) + T_2}{G_1^2(x) + G_2^2(x) + T_2} \quad (6)$$

where T_2 is a positive constant.

The combination of and gives the similarity of $f_1(x)$ and $f_2(x)$, which is represented as:

$$S_L(x) = S_{PC}(x) \cdot S_G(x)$$

The FSIM index between f_1 and f_2 is described as (Equation (7)):

$$FSIM = \frac{\sum_{X \in \Omega} S_L(X) \cdot PC_m(X)}{\sum_{X \in \Omega} PC_m(X)} \tag{7}$$

where $PC_m(X) = \max(PC_1(x) \cdot PC_2(x))$.

The FSIM index was designed to be used with grayscale images or the luminance component of color images. As chrominance information influences the HVS’s interpretation of images, including chrominance information in FSIM for color IQA may result in improved performance. A simple addition to the FSIM framework may be used to accomplish this purpose.

The RGB color images are first transformed to a different color space, where luminance and chrominance may be segregated. The commonly used YIQ color space, in which Y contains luminance information and I and Q provide chrominance information, is utilized to achieve this. The following equation (Equation (8)) may be used to convert from RGB to YIQ space:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{8}$$

The similarity between chromatic features is represented as follows:

$$S_I(x) = \frac{2I_1(x) \cdot I_2(x) + T_3}{I_1^2(x) + I_2^2(x) + T_3}$$

$$S_Q(x) = \frac{2Q_1(x) \cdot Q_2(x) + T_4}{Q_1^2(x) + Q_2^2(x) + T_4}$$

where $I_1(I_2)$ and $Q_1(Q_2)$ are the I and Q chromatic channels of the image $f_1(f_2)$, respectively. T_3 and T_4 are positive constants. $T_3 = T_4$ is used for convenience since the I and Q components have almost the same dynamic range.

The chrominance similarity between $f_1(x)$ and $f_2(x)$ is the obtained by combining the $S_I(x)$ and $S_Q(x)$, which is represented by $S_C(x)$ as follows:

$$S_C(x) = S_I(x) \cdot S_Q(x)$$

Finally, by simply adding chromatic information to the FSIM index, it may be easily modified to $FSIM_C$ (Equation (9)).

$$FSIM_C = \frac{\sum_{x \in \Omega} S_L(x) \cdot [S_C(x)]^\lambda \cdot PC_m(x)}{\sum_{x \in \Omega} PC_m(x)} \tag{9}$$

where $\lambda > 0$ is the parameter that is used to alter the significance of the chromatic components.

Visual Saliency Induced Index

Over the past decade, visual saliency (VS) has been extensively investigated in psychology, neuroscience and computer science to determine which portions of an image would grab the major attention of the HVS. Zhang et al. [75] introduced the visual saliency induced (VSI) index, a simple but highly effective full-reference IQA approach based on VS. The VSI metric between image 1 and image 2 is expressed as follows (Equation (10)):

$$VSI = \frac{\sum_{X \in \Omega} S(X) \cdot VS_m(X)}{\sum_{X \in \Omega} VS_m(X)} \tag{10}$$

where Ω denotes the complete spatial domain $S(X) = S_{VS}(X) \cdot [S_G(X)]^\alpha \cdot [S_C(X)]^\beta$.
 $S_{VS}(X)$ is a component of visual saliency similarity between two images.
 $S_G(X)$ is the similarity component of gradient modulus between two images.
 $S_C(X)$ is the component of chrominance similarity between two images.

$$VS_m(X) = \max(VS_1(x) \cdot VS_2(x))$$

$VS_1(x)$ and $VS_2(x)$ are visual saliency maps of image1 and image2, respectively.

3.2.3. Execution Time

The overall time required to identify, match and stitch the input images is referred to as the execution time. The execution time is a significant consideration when choosing an image stitching method for real-time applications. For some detector–descriptor pairs, either correct matches are zero or less than four. Therefore, for those pairs, execution time has no meaning because we consider execution time as total time from detection to final stitched image generation. In order to quantitatively compare those pairs with other pairs, we used a large dummy value for the execution time of those pairs. In this study, set this dummy value equal to 100 s.

The abovementioned IQA measures are used to assess the stitched image’s quality. The quality of the stitched image produced from every feature detector–descriptor pair is assessed by comparing the quality of the overlapping area of the input and the final stitched image. As shown in Figure 3, the overlapping regions of each left image (L) and right image (R) are compared to the overlapping areas of the final stitched image. In Figure 3, the overlapping area in the left image, right image and stitched image is represented by Lo , Ro and So , respectively. Quality metrics were calculated for each image pair and each detector–descriptor pair. In order to acquire quality metrics, the overlapping region So was compared to Lo and Ro individually, and then the mean value was computed. Figure 3 illustrates the entire workflow for evaluating the performance of the feature detector–descriptor combination for image stitching.

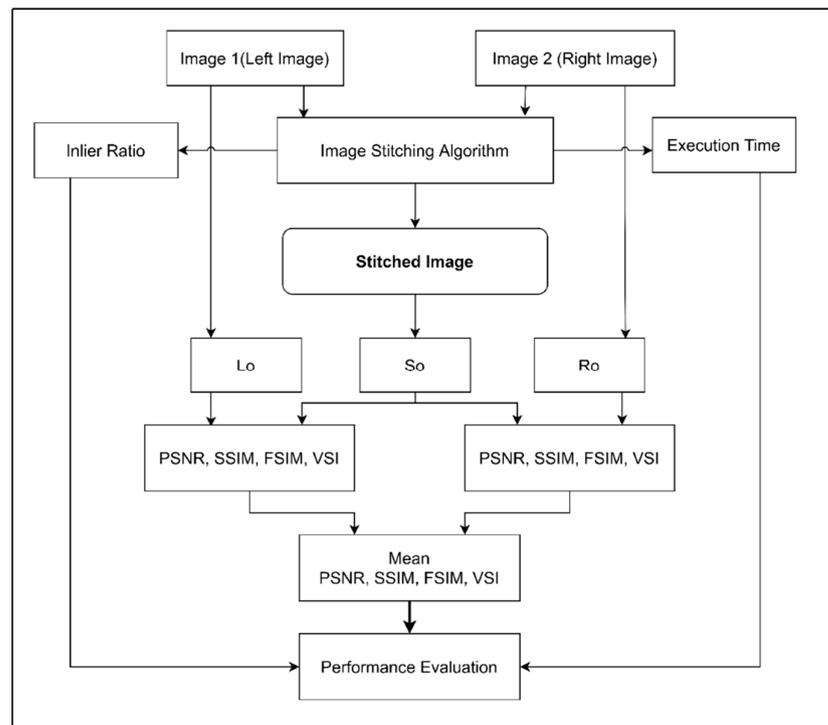


Figure 3. Workflow adopted for comparison of feature detector–descriptor pair for image stitching algorithm.

3.3. Method: Rank Computation

A rank is assigned to each combination of detector and descriptor based on the inlier ratio, stitching image quality and execution time. For comparison, six metrics are available: inlier ratio, execution time, PSNR, SSIM, FSIM and VSI. The rank computation algorithm has been adopted from [67,76]. The methodology for rank computation is outlined below:

1. For method m (detector–descriptor pair) with dataset d and metric t , find the positional rank $P_{m,d,t}$ by sorting the metric values (a low value denotes better positional rank for execution time, while a high value denotes better positional rank for all other metrics).
2. Compute the average rank of m for dataset d , over all metrics: $R_{m,d} = \frac{1}{N_t} \sum_t P_{m,d,t}$. Here, N_t denotes the number of metrics.
3. Compute the positional rank of each method m for each dataset d based on the value of $R_{m,d}$, by sorting $R_{m,d}$ in ascending order (low $R_{m,d}$ indicates better positional rank). If two successive rankings have extremely close values within a given threshold T , then the same positional rank is assigned. As a result, the positional ranks start with 1 and increment when two successive ranks differ by a value greater than the threshold T . The positional rank is denoted by $P_{m,d}$. For the rank computation, we have selected $T = 0.05 \times SD$, where SD is standard deviation.
4. Find the average rank across all datasets: $R_m = \frac{1}{N_d} \sum_d P_{m,d}$. Here, N_d represents the number of datasets.
5. Finally, determine P_m , the positional rank across datasets by sorting R_m in ascending order.

Using the abovementioned algorithm, we calculated the rank of each method m (detector and descriptor combination). We computed the rank of each method m for datasets 1, 2 and 3 separately.

4. Results and Discussion

In this study, we analyzed a total 85 feature detector–descriptor combinations pairs. It is not possible to show outputs of all combinations here; therefore, we show stitched image outputs of the top six combinations in each dataset category. Figure 4 shows the output stitched images of the top six combinations which produce better quality of stitched images in comparison to all other combinations for dataset 1. Upon close examination of the stitched images, it can be observed that AKAZE + AKAZE combination gives the best-quality image, which is further verified by the objective IQA metrics.

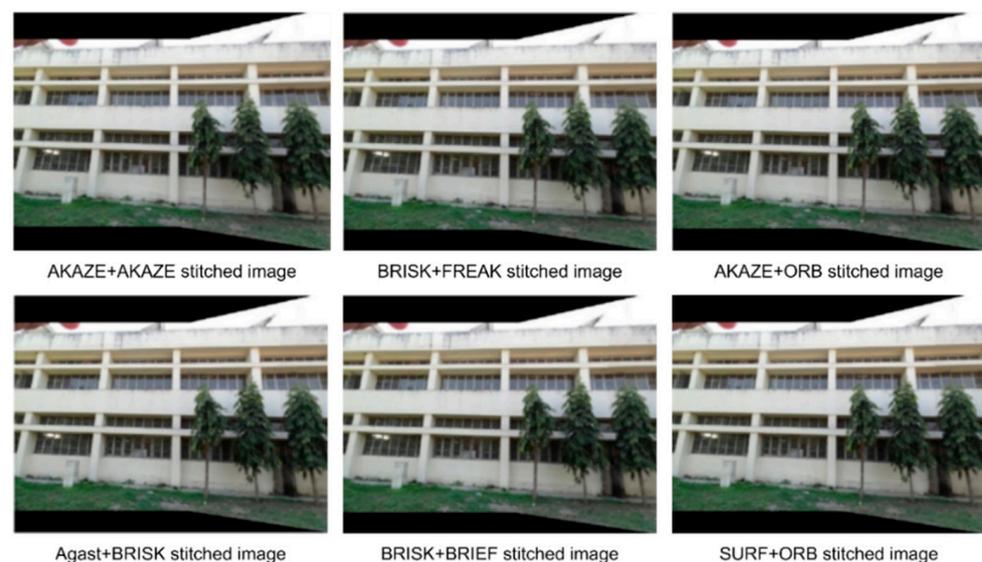


Figure 4. Image stitching outputs obtained for different feature detector–descriptor combinations for dataset 1.

Table 2 shows the inlier ratio obtained using different combinations of feature detector and descriptors for dataset 1. Tables 3–6 presents the values of PSNR, SSIM, FSIM and VSI obtained using various combinations of detector–descriptor pairs for dataset 1. Table 7 shows execution time of different detector–descriptor-based algorithms for dataset 1. In all the tables, some entries are marked as NC. NC stands for “not compatible”, i.e., that particular descriptor is not compatible with combined detector.

Table 2. Inlier ratio for different feature detector–descriptor combinations for dataset 1.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.60251	0.80000	0.60709	0.71831	NC	0.41398	0.51759	0.00410
AKAZE	0.86667	0.75000	0.59322	0.88148	0.61538	0.83333	0.62295	0.91489	0.58879
BRISK	NC	0.56075	0.52500	0.64253	0.67742	NC	0.52941	0.53763	0.69101
FAST	NC	0.58482	0.54386	0.74601	0.73239	NC	0.42045	0.79915	0.62058
GFTT	NC	0.50000	0.64103	0.75758	0.60526	NC	0.51613	0.60465	0.05063
KAZE	NC	0.54286	0.77778	0.75758	0.71429	0.77439	0.55714	0.70886	0.58000
MSD	NC	0.50746	0.63333	0.54206	0.62500	NC	0.55882	NC	0.66197
MSER	NC	1.00000	0.66667	0.80000	0.66667	NC	0.50000	0.83333	0.75676
ORB	NC	0.83871	0.94737	0.51471	0.84615	NC	0.51852	0.96154	0.81379
SIFT	NC	0.52941	0.71795	0.77586	0.50000	NC	NC	0.77622	0.49495
Star	NC	0.86364	0.83333	0.66667	0.81818	NC	0.73333	0.68182	0.72000
SURF	NC	0.57664	0.55556	0.71886	0.66667	NC	0.66667	0.59669	0.51444

Table 3. PSNR for different feature detector–descriptor combinations for dataset 1.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	21.54000	21.71477	20.95036	20.96922	NC	21.45695	21.15432	0.00000
AKAZE	21.76170	19.38284	21.72805	21.21242	21.34421	21.19080	21.60196	21.27625	20.47913
BRISK	NC	21.62408	19.97168	21.41030	21.52767	NC	18.33011	21.46034	21.40856
FAST	NC	21.55781	21.65323	21.34158	21.13314	NC	21.21108	20.95768	21.34512
GFTT	NC	21.01897	21.67666	21.31260	21.28477	NC	20.14706	21.04440	0.00000
KAZE	NC	21.25731	20.92894	21.40749	21.04062	21.46285	21.25410	19.47301	20.85841
MSD	NC	17.87572	19.83321	21.14125	21.14215	NC	19.30856	NC	21.47661
MSER	NC	0.00000	21.32188	0.00000	0.00000	NC	0.00000	0.00000	21.32188
ORB	NC	19.79712	17.97762	19.98834	0.00000	NC	20.48180	17.45504	21.40705
SIFT	NC	21.00432	19.66354	21.38760	18.60319	NC	NC	21.53374	21.28695
Star	NC	21.27717	0.00000	20.65158	19.39711	NC	19.25112	19.58099	21.14633
SURF	NC	21.52246	21.31893	21.16714	20.85799	NC	21.41367	20.97204	21.80610

Table 4. SSIM for different feature detector–descriptor combinations for dataset 1.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.88586	0.89520	0.87894	0.87983	NC	0.89424	0.88179	0.00000
AKAZE	0.89995	0.86496	0.89842	0.88312	0.88614	0.88650	0.89473	0.88479	0.87558
BRISK	NC	0.89300	0.87014	0.88856	0.89140	NC	0.85202	0.88985	0.88842
FAST	NC	0.89108	0.89541	0.88712	0.88115	NC	0.88701	0.87797	0.88643
GFTT	NC	0.88187	0.89984	0.88904	0.88494	NC	0.87255	0.88004	0.00000
KAZE	NC	0.88757	0.87775	0.89257	0.87984	0.88815	0.88547	0.87403	0.88592
MSD	NC	0.85835	0.86950	0.88176	0.88232	NC	0.86985	NC	0.88996
MSER	NC	0.00000	0.88502	0.00000	0.00000	NC	0.00000	0.00000	0.88502
ORB	NC	0.86732	0.86047	0.87287	0.00000	NC	0.84398	0.84205	0.88709
SIFT	NC	0.87935	0.86726	0.88790	0.85785	NC	NC	0.87156	0.88571
Star	NC	0.88511	0.00000	0.87525	0.87058	NC	0.86473	0.87082	0.88453
SURF	NC	0.89100	0.89185	0.88182	0.88308	NC	0.89292	0.88143	0.88022

Table 5. FSIM for different feature detector–descriptor combinations for dataset 1.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.94355	0.95077	0.93748	0.93947	NC	0.95000	0.94221	0.00000
AKAZE	0.95330	0.92549	0.95267	0.94204	0.94344	0.94460	0.95137	0.94286	0.93521
BRISK	NC	0.94916	0.92968	0.94548	0.94889	NC	0.91886	0.94745	0.94628
FAST	NC	0.94720	0.95091	0.94546	0.94126	NC	0.94487	0.93825	0.94504
GFTT	NC	0.94087	0.95315	0.94655	0.94392	NC	0.93165	0.93825	0.00000
KAZE	NC	0.94545	0.93638	0.94845	0.93915	0.94415	0.94436	0.93596	0.94456
MSD	NC	0.92137	0.92986	0.94078	0.94171	NC	0.93221	NC	0.94786
MSER	NC	0.00000	0.94379	0.00000	0.00000	NC	0.00000	0.00000	0.94379
ORB	NC	0.92842	0.92445	0.93269	0.00000	NC	0.90059	0.91069	0.94448
SIFT	NC	0.93883	0.92854	0.94578	0.92059	NC	NC	0.94881	0.94438
Star	NC	0.94192	0.00000	0.93588	0.92741	NC	0.92715	0.92577	0.94362
SURF	NC	0.94779	0.94702	0.94071	0.94120	NC	0.95054	0.94099	0.94022

Table 6. VSI for different feature detector–descriptor combinations for dataset 1.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.91869	0.92354	0.90394	0.90831	NC	0.92501	0.91243	0.00000
AKAZE	0.92945	0.87339	0.92881	0.91095	0.91303	0.91170	0.92436	0.91196	0.89611
BRISK	NC	0.92252	0.88374	0.91531	0.92080	NC	0.85495	0.91904	0.91795
FAST	NC	0.91688	0.92586	0.91589	0.90936	NC	0.91269	0.90447	0.91535
GFTT	NC	0.90719	0.92617	0.92024	0.91344	NC	0.89364	0.90513	0.00000
KAZE	NC	0.91274	0.89847	0.92245	0.90594	0.91285	0.91383	0.89385	0.91251
MSD	NC	0.85767	0.88436	0.90943	0.90857	NC	0.88385	NC	0.91923
MSER	NC	0.00000	0.91407	0.00000	0.00000	NC	0.00000	0.00000	0.91407
ORB	NC	0.88139	0.86445	0.88829	0.00000	NC	0.90260	0.83293	0.91399
SIFT	NC	0.90636	0.88154	0.91642	0.86461	NC	NC	0.91924	0.91398
Star	NC	0.90999	0.00000	0.89682	0.88858	NC	0.87969	0.88514	0.91265
SURF	NC	0.91988	0.91946	0.90890	0.91036	NC	0.92380	0.91110	0.90780

Table 7. Execution time for different feature detector–descriptor combinations for dataset 1.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	1.749	2.02	2.193	1.793	NC	1.704	1.94	1.961
AKAZE	1.5779	1.694	1.862	1.785	1.699	1.805	1.698	1.757	1.7
BRISK	NC	1.57	1.81	1.73	1.58	NC	1.755	2.1	1.85
FAST	NC	1.61	1.725	2.04	1.74	NC	1.498	2.188	1.62
GFTT	NC	1.52	1.96	1.68	1.61	NC	1.685	1.587	1.54
KAZE	NC	1.926	2.087	2.065	1.983	1.959	1.787	2.039	1.95
MSD	NC	2.02	2.188	2.136	2.034	NC	2.03	NC	2.021
MSER	NC	1.56	1.775	1.666	1.56	NC	1.55	2.98	1.64
ORB	NC	1.555	1.63	1.645	1.734	NC	1.51	2.14	1.578
SIFT	NC	1.718	1.96	1.899	1.774	NC	NC	1.921	1.751
Star	NC	1.593	1.687	1.731	1.607	NC	1.61	1.704	1.586
SURF	NC	1.691	1.873	1.859	1.684	NC	1.697	3.47	1.794

For dataset 1, ORB + SIFT has a very good inlier ratio, but the quality of the stitched image is poor in comparison to other pairs. This pair has poor quality of stitched images despite its very good inlier ratio because in order to obtain a better-quality image, the matched points should be spread into the image, but for this pair, matched points are clustered. It also takes more execution time. ORB + BRISK also has a very good inlier ratio, but the quality of the stitched image is poor because of the low number of matched points. It has better execution time than ORB + SIFT, which may be because the BRISK descriptor takes less time to extract descriptors than SIFT. AKAZE + SIFT and AKAZE + DAISY both have very good inlier ratio, a little lower than ORB + BRISK. The quality of stitched image

obtained using both the combinations is also good because they have sufficient matches. AKAZE + AKAZE also has a good inlier ratio, and the quality of stitched image is also superior to all other combinations. This is because the number of matches is good and are well spread over the image. It also takes less execution time. GFTT + BRISK, AKAZE + ORB, AKAZE + BRISK and FAST + BRISK all have average inlier ratio, but the quality of stitched image is very good. This may be because these combinations produce more false matches, which reduces the inlier ratio, and the correct matches are well spread over the image, which leads to a better quality stitched image. BRISK + ORB and SIFT + FREAK have low inlier ratio and degraded quality of the stitched image. Overall, AKAZE + AKAZE has very good inlier ratio, generating very good stitched images with relatively short execution time (Table 7), which implies that AKAZE + AKAZE is best suited for stitching dataset 1.

Figure 5 shows results obtained using AKAZE + AKAZE, BRISK + ORB and SIFT + FREAK combinations for dataset 1. The first column in each row displays the initial matches for that particular detector–descriptor combination, and the second column shows the correct matches for the detector–descriptor pair. The last column shows the stitched image obtained using that particular combination.

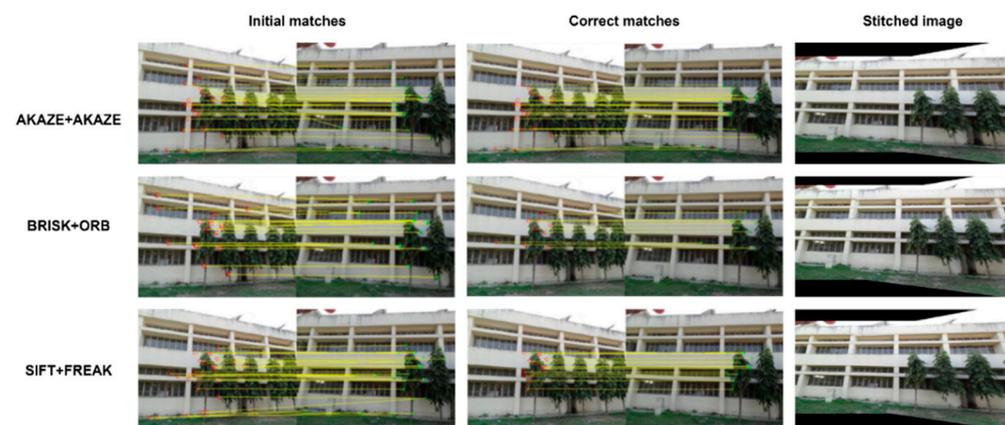


Figure 5. Initial matches, correct matches and stitched images for different feature detector–descriptor combinations for dataset 1.

Figure 6 shows the output stitched images of the top six combinations, which produces a better quality of stitched image in comparison to all other combinations for dataset 2. Again, it is observed that the AKAZE + AKAZE combination gives the best-quality image.

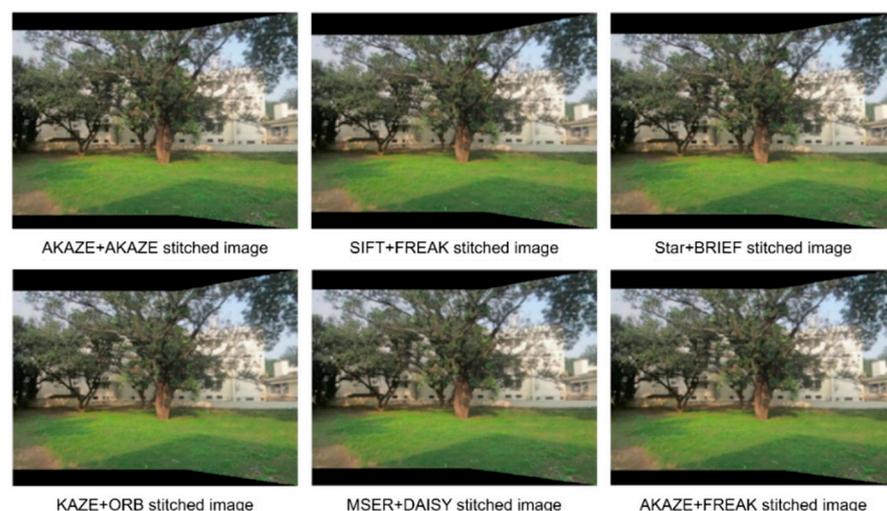


Figure 6. Image stitching outputs obtained for different feature detector–descriptor combinations for dataset 2.

Table 8 shows the inlier ratio obtained using different combinations of feature detector and descriptors for dataset 2. Tables 9–12 presents the values of PSNR, SSIM, FSIM and VSI obtained using various combinations of detector–descriptor pairs for dataset 2. It is observed that the AKAZE + AKAZE combination has highest values of quality metrics for dataset 2.

Table 8. Inlier ratio for different feature detector–descriptor combinations for dataset 2.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.64966	0.63636	0.71163	0.65574	NC	0.66492	0.63415	0.00169
AKAZE	0.90000	0.73134	1.00000	0.81776	0.76471	0.91228	0.86792	0.59441	0.79104
BRISK	NC	0.81452	0.92308	0.54906	0.88889	NC	1.00000	0.70703	0.73200
FAST	NC	0.54984	0.83333	0.78214	0.59091	NC	0.62911	0.84132	0.58296
GFTT	NC	0.78378	0.00000	0.80180	0.50000	NC	0.77273	0.85714	0.08333
KAZE	NC	0.57627	0.88235	0.86957	0.78947	0.53266	0.87234	0.85714	0.72549
MSD	NC	0.62500	0.66667	0.61069	1.00000	NC	0.73684	NC	0.74074
MSER	NC	0.00000	0.00000	0.89474	0.71429	NC	1.00000	0.88235	0.95000
ORB	NC	0.54167	1.00000	0.55102	0.00000	NC	0.55556	0.90000	0.79452
SIFT	NC	0.60345	0.85714	0.75806	0.86667	NC	NC	0.88732	0.41096
Star	NC	1.00000	0.00000	0.55263	0.50000	NC	1.00000	0.93478	0.87500
SURF	NC	0.59740	0.95238	0.59006	0.63636	NC	0.76471	0.82514	0.77273

Table 9. PSNR for different feature detector–descriptor combinations for dataset 2.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	23.00484	16.84685	23.08943	22.85816	NC	22.96099	23.30412	0.00000
AKAZE	24.34155	23.16761	22.64696	23.29457	23.74266	23.48015	22.90629	22.22603	23.13500
BRISK	NC	23.60624	22.85367	23.91664	23.55245	NC	23.96318	22.07020	23.05836
FAST	NC	21.63101	23.44776	23.23622	22.62471	NC	22.86712	23.41391	23.32138
GFTT	NC	22.52211	0.00000	23.22265	22.40035	NC	20.70295	23.13159	0.00000
KAZE	NC	22.18398	23.23336	23.11630	24.05871	22.54845	24.22516	22.68412	23.78771
MSD	NC	20.99705	0.00000	23.14319	22.60502	NC	21.63379	NC	22.86339
MSER	NC	0.00000	0.00000	23.93411	0.00000	NC	0.00000	19.13762	23.62422
ORB	NC	17.32199	17.22133	19.27183	0.00000	NC	16.34763	22.92395	22.93993
SIFT	NC	22.68530	22.42940	23.14216	24.28646	NC	NC	23.03123	23.06673
Star	NC	23.84176	0.00000	21.00664	0.00000	NC	22.67085	22.06882	22.41675
SURF	NC	23.16130	21.17148	23.72378	21.09816	NC	22.56551	22.94019	22.78473

Table 10. SSIM for different feature detector–descriptor combinations for dataset 2.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.90377	0.85146	0.90017	0.90272	NC	0.90198	0.90157	0.00000
AKAZE	0.90828	0.90134	0.89826	0.90195	0.90504	0.90420	0.89978	0.89541	0.90148
BRISK	NC	0.90339	0.89830	0.90446	0.90380	NC	0.90609	0.89466	0.90039
FAST	NC	0.89339	0.90131	0.90109	0.89753	NC	0.89865	0.90272	0.90247
GFTT	NC	0.89805	0.00000	0.90073	0.89472	NC	0.88489	0.89852	0.00000
KAZE	NC	0.89447	0.90355	0.90123	0.90555	0.89700	0.90768	0.89868	0.90562
MSD	NC	0.88723	0.00000	0.90344	0.89946	NC	0.89348	NC	0.90024
MSER	NC	0.00000	0.00000	0.90696	0.00000	NC	0.00000	0.87682	0.90396
ORB	NC	0.85524	0.85148	0.86754	0.00000	NC	0.83955	0.89755	0.90041
SIFT	NC	0.89933	0.89299	0.90052	0.90759	NC	NC	0.89993	0.90245
Star	NC	0.90369	0.00000	0.88529	0.00000	NC	0.89795	0.89454	0.89701
SURF	NC	0.90221	0.88801	0.90492	0.88590	NC	0.89784	0.90061	0.89932

Table 11. FSIM for different feature detector–descriptor combinations for dataset 2.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.96231	0.92468	0.96185	0.96158	NC	0.96206	0.96236	0.00000
AKAZE	0.96815	0.96361	0.95989	0.96350	0.96638	0.96505	0.96240	0.95716	0.96387
BRISK	NC	0.96483	0.96040	0.96593	0.96617	NC	0.96667	0.95559	0.96320
FAST	NC	0.95133	0.96271	0.96375	0.96016	NC	0.96053	0.96549	0.96425
GFTT	NC	0.95884	0.00000	0.96299	0.95429	NC	0.94643	0.96266	0.00000
KAZE	NC	0.95427	0.96394	0.96309	0.96688	0.95725	0.96788	0.96099	0.96684
MSD	NC	0.94700	0.00000	0.96353	0.96082	NC	0.95147	NC	0.96173
MSER	NC	0.00000	0.00000	0.96803	0.00000	NC	0.00000	0.94032	0.96544
ORB	NC	0.92883	0.92786	0.93556	0.00000	NC	0.92064	0.95945	0.96191
SIFT	NC	0.95878	0.95433	0.96179	0.96767	NC	NC	0.96255	0.96283
Star	NC	0.96508	0.00000	0.94677	0.00000	NC	0.96008	0.95532	0.95894
SURF	NC	0.96285	0.94828	0.96626	0.94667	NC	0.96015	0.96206	0.96117

Table 12. VSI for different feature detector–descriptor combinations for dataset 2.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.94422	0.84767	0.94742	0.94294	NC	0.94438	0.94524	0.00000
AKAZE	0.95572	0.94936	0.94312	0.94749	0.95098	0.94875	0.94704	0.93782	0.94805
BRISK	NC	0.94962	0.94632	0.95064	0.95176	NC	0.95355	0.93561	0.94870
FAST	NC	0.92552	0.94729	0.94979	0.94521	NC	0.94667	0.95097	0.94813
GFTT	NC	0.94101	0.00000	0.94909	0.93461	NC	0.92189	0.94804	0.00000
KAZE	NC	0.93358	0.94621	0.94696	0.95459	0.93875	0.95381	0.94469	0.95096
MSD	NC	0.90969	0.00000	0.94615	0.94367	NC	0.92711	NC	0.94581
MSER	NC	0.00000	0.00000	0.95358	0.00000	NC	0.00000	0.89024	0.95079
ORB	NC	0.86680	0.86223	0.89071	0.00000	NC	0.84705	0.94044	0.94634
SIFT	NC	0.93854	0.93378	0.94606	0.95510	NC	NC	0.94793	0.94534
Star	NC	0.95047	0.00000	0.91623	0.00000	NC	0.94523	0.93481	0.94156
SURF	NC	0.94505	0.91856	0.95114	0.91616	NC	0.94321	0.94494	0.94530

Table 13 shows execution time of different detector–descriptor-based algorithms for dataset 2.

Table 13. Execution time for different feature detector–descriptor combinations for dataset 2.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	1.842	2.579	3.12	2.02	NC	1.76	2.54	2.41
AKAZE	1.4513	1.441	1.58	1.595	1.457	1.8218	1.444	1.579	1.467
BRISK	NC	1.853	2.12	2.231	1.87	NC	1.865	3.155	2.035
FAST	NC	1.956	2.41	3.364	2.155	NC	1.91	3.82	2.42
GFTT	NC	1.372	100	1.552	1.388	NC	1.4	1.468	1.426
KAZE	NC	1.656	1.92	1.833	1.738	2.0176	1.659	1.78	1.69
MSD	NC	1.782	1.867	1.918	1.777	NC	1.757	NC	1.797
MSER	NC	100	100	1.72	1.574	NC	1.556	2.57	1.63
ORB	NC	1.54	1.7	1.658	100	NC	1.46	2.022	1.584
SIFT	NC	1.51	1.71	1.786	1.573	NC	NC	1.841	1.576
Star	NC	1.358	100	1.497	1.399	NC	1.375	1.465	1.316
SURF	NC	1.426	1.62	1.636	1.472	NC	1.438	3.24	1.535

For dataset 2, AKAZE + BRISK and ORB + BRISK both have very good inlier ratio, but the quality of the stitched image is poor; this is because they have a smaller number of correct matched points, which affected the homography metric that in turn resulted into degraded quality of the stitched image. SURF + BRISK and ORB + SIFT also have a very good inlier ratio, but the quality of stitched image is poor; this is because the matched points are not well spread in the image. AKAZE + AKAZE has a very good inlier ratio, and stitched

image quality is also very good. This is because AKAZE + AKAZE has a good number of matched points that are well spread across the image. KAZE + ORB, SIFT + FREAK, KAZE + FREAK and AKAZE + FREAK have average inlier ratio in comparison to other combinations, but the quality of stitched image is good. The inlier ratio is low because of the higher number of initial matched points and smaller number of correct matched points in comparison to initial matched points, but correct matched points are sufficient to obtain an accurate homography metric, which in turn gives a good quality of stitched image. AGAST + BRISK has low inlier ratio and poor stitched image quality because of a smaller number of matched points. SIFT + BRIEF and AKAZE + SIFT also generate a poor quality of stitched image because of the low inlier ratio. Once again, AKAZE + AKAZE generated a very good quality of stitched image with relatively short execution time (Table 13), which makes AKAZE + AKAZE the best choice for stitching dataset 2, which contains outdoor scenes with irregular features such as trees.

Figure 7 shows results obtained using AKAZE + AKAZE, AGAST + BRISK and ORB + BRISK combinations for dataset 2. Each row in the figure presents a detector–descriptor combination, where the first column illustrates the initial matches for that particular pair, the second column shows the correct matches, and the last column exhibits the stitched image produced using that combination.

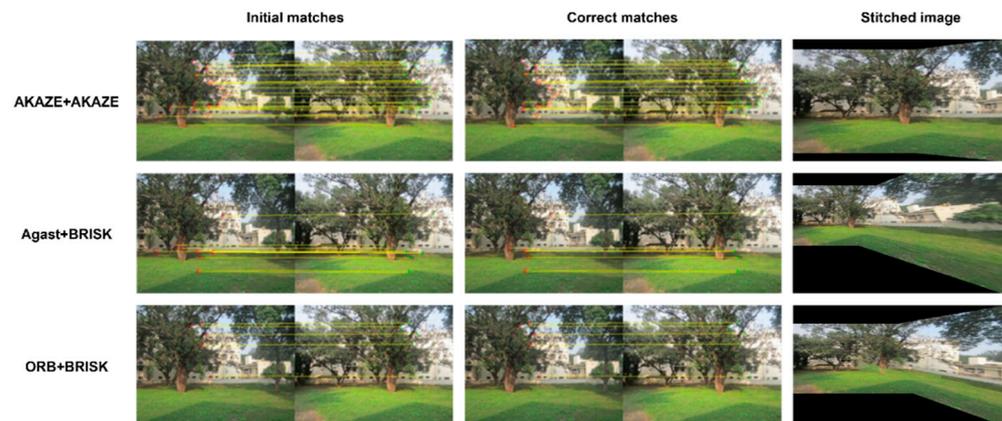


Figure 7. Initial matches, correct matches and stitched images for different feature detector–descriptor combinations for dataset 2.

Figure 8 shows the output stitched images of the top six combinations which produce a better quality of stitched image in comparison to all other combinations for dataset 3. It is observed that the AKAZE + AKAZE combination gives the best-quality image. We found one more combination, AGAST + BRIEF which also gives a better quality of stitched image.

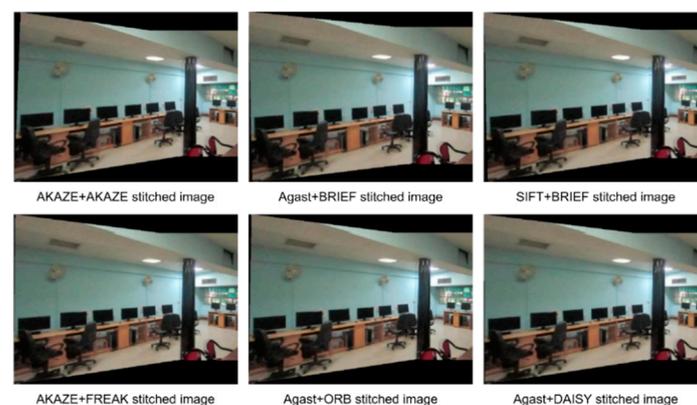


Figure 8. Image stitching outputs obtained for different feature detector–descriptor combinations for dataset 3.

Table 14 shows the inlier ratio, obtained using different combinations of feature detector and descriptors for dataset 3. Tables 15–18 presents the values of PSNR, SSIM, FSIM and VSI obtained using various combinations of detector–descriptor pairs for dataset 3. It is observed that the AKAZE + AKAZE combination has the highest values of quality metrics for dataset 3. It is observed that the AGAST + BRIEF combination also gives a very good quality of stitched image.

Table 14. Inlier ratio for different feature detector–descriptor combinations for dataset 3.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.62500	0.56250	0.72189	0.53846	NC	0.72093	0.41489	0.01511
AKAZE	0.79167	0.71795	0.30769	0.43820	0.63158	0.60377	0.52174	0.86792	0.42857
BRISK	NC	0.61290	0.88889	0.51818	0.75000	NC	1.00000	0.56522	0.57143
FAST	NC	0.58696	0.37500	0.33516	0.35294	NC	0.33333	0.55714	0.42241
GFTT	NC	0.63043	0.88889	0.46497	0.52632	NC	0.58974	0.40000	0.05479
KAZE	NC	0.46809	0.88235	0.61261	0.40541	0.51471	0.40909	0.37255	0.21429
MSD	NC	0.53659	1.00000	0.58108	0.61111	NC	0.50000	NC	0.50549
MSER	NC	0.00000	0.90000	0.00000	1.00000	NC	0.00000	0.00000	0.00000
ORB	NC	0.63415	1.00000	0.50000	0.00000	NC	0.75000	0.55556	0.51163
SIFT	NC	0.70000	0.66667	0.64634	0.66667	NC	NC	0.77778	0.42105
Star	NC	0.50000	0.00000	0.62500	0.57143	NC	0.00000	0.90909	0.50000
SURF	NC	0.51786	1.00000	0.58955	0.62500	NC	0.27778	0.68966	0.52000

Table 15. PSNR for different feature detector–descriptor combinations for dataset 3.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	22.02144	18.41438	20.76560	20.03954	NC	20.20800	19.54466	0.00000
AKAZE	22.84485	18.06152	0.00000	18.39431	20.76954	18.36875	19.53043	18.12729	19.31158
BRISK	NC	19.36902	18.46584	17.06048	18.95291	NC	0.00000	17.31023	19.32706
FAST	NC	16.78586	19.61524	17.78124	19.49148	NC	17.07517	18.61330	18.12281
GFTT	NC	20.08011	18.56651	18.09206	16.64466	NC	17.14658	20.52409	0.00000
KAZE	NC	20.19243	18.99885	18.52442	19.78398	19.07275	18.69729	19.85329	19.95500
MSD	NC	17.14233	20.09848	20.97889	20.89052	NC	19.42852	NC	18.33625
MSER	NC	0.00000	0.00000	0.00000	0.00000	NC	0.00000	0.00000	0.00000
ORB	NC	14.90741	16.15874	17.05856	0.00000	NC	16.89904	16.68980	18.53520
SIFT	NC	20.79336	16.22122	20.06282	17.22325	NC	NC	20.65888	20.48601
Star	NC	0.00000	0.00000	18.48441	0.00000	NC	0.00000	15.06165	0.00000
SURF	NC	19.93502	19.30531	20.44127	19.35691	NC	0.00000	15.39689	20.30642

Table 16. SSIM for different feature detector–descriptor combinations for dataset 3.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.92421	0.88787	0.91452	0.90307	NC	0.91055	0.90559	0.00000
AKAZE	0.92795	0.87511	0.00000	0.88470	0.91402	0.88450	0.90245	0.88707	0.89504
BRISK	NC	0.89793	0.88498	0.86930	0.90047	NC	0.00000	0.86685	0.89944
FAST	NC	0.86412	0.90337	0.87334	0.89907	NC	0.87504	0.89175	0.87711
GFTT	NC	0.90535	0.88760	0.88288	0.87678	NC	0.87336	0.91187	0.00000
KAZE	NC	0.90907	0.89474	0.88894	0.90696	0.89645	0.89486	0.90026	0.90812
MSD	NC	0.86467	0.90921	0.91625	0.91557	NC	0.90149	NC	0.88236
MSER	NC	0.00000	0.00000	0.00000	0.00000	NC	0.00000	0.00000	0.00000
ORB	NC	0.81804	0.83011	0.86148	0.00000	NC	0.85702	0.83477	0.88902
SIFT	NC	0.91456	0.84686	0.90694	0.86575	NC	NC	0.91301	0.91229
Star	NC	0.00000	0.00000	0.89265	0.00000	NC	0.00000	0.80608	0.00000
SURF	NC	0.90611	0.89552	0.91018	0.89768	NC	0.00000	0.82859	0.90664

Table 17. FSIM for different feature detector–descriptor combinations for dataset 3.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.95084	0.92338	0.94169	0.93500	NC	0.93938	0.93235	0.00000
AKAZE	0.95365	0.91898	0.00000	0.92411	0.94159	0.92430	0.93203	0.92156	0.92990
BRISK	NC	0.93118	0.92508	0.91395	0.93028	NC	0.00000	0.91996	0.93030
FAST	NC	0.91604	0.93123	0.91919	0.92941	NC	0.92273	0.92661	0.92434
GFTT	NC	0.93543	0.92320	0.92314	0.92490	NC	0.91504	0.94085	0.00000
KAZE	NC	0.93654	0.92848	0.92566	0.93346	0.92890	0.92506	0.93035	0.93508
MSD	NC	0.91946	0.93784	0.94359	0.94208	NC	0.93042	NC	0.92493
MSER	NC	0.00000	0.00000	0.00000	0.00000	NC	0.00000	0.00000	0.00000
ORB	NC	0.90867	0.90987	0.91765	0.00000	NC	0.91657	0.90508	0.92584
SIFT	NC	0.94167	0.91692	0.93650	0.92028	NC	NC	0.94106	0.94027
Star	NC	0.00000	0.00000	0.92239	0.00000	NC	0.00000	0.91164	0.00000
SURF	NC	0.93525	0.92751	0.93854	0.92864	NC	0.00000	0.91272	0.93544

Table 18. VSI for different feature detector–descriptor combinations for dataset 3.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	0.93672	0.87900	0.91742	0.90365	NC	0.91201	0.90072	0.00000
AKAZE	0.93895	0.86206	0.00000	0.88116	0.91780	0.88055	0.89891	0.86782	0.89474
BRISK	NC	0.89504	0.88327	0.84984	0.88831	NC	0.00000	0.86461	0.89528
FAST	NC	0.85610	0.90165	0.85874	0.89234	NC	0.85849	0.88605	0.87461
GFTT	NC	0.90417	0.87889	0.87673	0.84947	NC	0.84800	0.91618	0.00000
KAZE	NC	0.90833	0.89154	0.88127	0.90430	0.89230	0.88185	0.89608	0.90372
MSD	NC	0.86812	0.90999	0.92171	0.91473	NC	0.89695	NC	0.88206
MSER	NC	0.00000	0.00000	0.00000	0.00000	NC	0.00000	0.00000	0.00000
ORB	NC	0.82422	0.80792	0.85765	0.00000	NC	0.83008	0.81993	0.88374
SIFT	NC	0.91806	0.84770	0.90704	0.86804	NC	NC	0.91775	0.91431
Star	NC	0.00000	0.00000	0.87590	0.00000	NC	0.00000	0.83340	0.00000
SURF	NC	0.90624	0.88967	0.91287	0.89395	NC	0.00000	0.83408	0.90552

For dataset 3, MSD + BRISK, ORB + BRISK, SURF + BRISK and GFTT + BRISK have very good inlier ratio, but the quality of stitched image is poor because of a smaller number of matched points. This also implies that the BRISK descriptor is not suitable for stitching indoor scene images. AKAZE + SIFT has a good inlier ratio, but the quality of stitched image is poor because matched points are not well spread across the image. AKAZE + AKAZE has a good inlier ratio, and the stitched image quality is also very good because it has sufficient number of matched points that are well spread across the images. SIFT + FREAK has an average inlier ratio in comparison to other combinations, and its quality is poor because it has a smaller number of matched points. AGAST + DAISY has average inlier ratio, but the quality of stitched image is very good because of the sufficient number of correct matches. SIFT + BRIEF, AKAZE + FREAK, AGAST + BRIEF and MSD + FREAK generate a good quality of stitched images because the correct matched points are well spread across the image. AGAST + BRISK, ORB + SIFT and KAZE + ORB have poor inlier ratio and poor quality of stitched image. This is because they have very a smaller number of matched points. The inlier ratio of AKAZE + ORB, KAZE + FREAK and FAST + BRISK is low, and the quality of stitched image is not good because of a smaller number of matched points. Overall, AKZE + AKAZE and AGAST + BRIEF generate a very good quality of stitched image, and both are computationally faster (Table 19) for stitching dataset 3 images. Therefore, it can be concluded that AKAZE + AKZE and AGAST + BRIEF are the best choices for stitching indoor scene images.

Table 19. Execution time for different feature detector–descriptor combinations for dataset 3.

Det/Desc	AKAZE	BRIEF	BRISK	DAISY	FREAK	KAZE	ORB	SIFT	SURF
AGAST	NC	1.461	1.658	1.651	1.566	NC	1.478	1.548	1.477
AKAZE	1.442	1.576	1.722	1.792	1.53	1.81245	1.543	1.634	1.586
BRISK	NC	1.61	1.738	1.793	1.694	NC	1.768	1.782	1.684
FAST	NC	1.593	1.529	1.67	1.426	NC	1.562	1.722	1.477
GFTT	NC	1.431	1.836	1.613	1.374	NC	1.53	1.44	1.416
KAZE	NC	1.671	2.023	2.073	1.687	2.0259	1.666	1.686	1.656
MSD	NC	1.946	2.023	2.102	1.855	NC	1.865	NC	1.925
MSER	NC	100	1.702	100	1.494	NC	100	100	100
ORB	NC	2.229	2.171	1.757	100	NC	1.62	2.304	1.662
SIFT	NC	1.518	1.752	1.658	1.603	NC	NC	1.646	1.563
Star	NC	1.368	100	1.629	1.406	NC	100	2.309	1.395
SURF	NC	1.522	1.614	1.639	1.532	NC	1.472	2.785	1.56

Figure 9 shows results obtained using AKAZE + AKAZE, AGAST + BRIEF, ORB + BRISK and GFTT + BRISK combinations for dataset 3. In the figure, each row represents a unique detector–descriptor combination. The first column demonstrates the initial matches for that particular pair, while the second column showcases the correct matches. The last column displays the resulting stitched image obtained using that specific combination.



Figure 9. Initial matches, correct matches and stitched images for different feature detector–descriptor combinations for dataset 3.

We also considered six publicly available datasets for image stitching. The input images of the dataset used and the generated stitched output images are given in Appendix A. We examined a pair of deep learning algorithms for image stitching that were recently introduced in the literature. Appendix B contains the results obtained using these deep learning algorithms.

Table 20 (dataset 1 column) shows the top ten methods (detector and descriptor combination) for dataset 1. AKAZE + AKAZE is top ranked for dataset 1, and after that BRISK + FREAK and AKAZE + ORB are at second position. Table 20 (dataset 2 column) presents the top ten methods for dataset 2. AKAZE + AKAZE performs the best in dataset 2 and hence ranks first. SIFT + FREAK and Star + BRIEF are at the second position. Table 20 (dataset 3 column) shows the top ten methods for dataset 3. AKAZE + AKAZE is at top position for dataset 3, and AGAST + BRIEF secures the second position. SIFT + BRIEF is in the third position.

Finally, the ranking may be adjusted by weighting particular metrics while averaging the results. In future, different weights can be assigned to each metric depending upon the requirements; for example, in real time applications, we can compromise with quality for shorter execution time; therefore, we can assign higher weight to execution time in

comparison to quality metrics. In virtual tour applications, quality is important, so we can assign quality metrics higher weight than execution time.

Table 20. Top ten detector–descriptor combinations for each dataset.

Dataset 1		Dataset 2		Dataset 3	
Method	Rank	Method	Rank	Method	Rank
AKAZE + AKAZE	1	AKAZE + AKAZE	1	AKAZE + AKAZE	1
BRISK + FREAK	2	SIFT + FREAK	2	AGAST + BRIEF	2
AKAZE + ORB	2	Star + BRIEF	2	SIFT + BRIEF	3
AGAST + BRISK	3	KAZE + ORB	3	AKAZE + FREAK	4
BRISK + BRIEF	3	MSER + DAISY	3	AGAST + ORB	5
SURF + ORB	3	AKAZE + FREAK	4	AGAST + DAISY	5
GFTT + BRISK	4	BRISK + ORB	4	SIFT + SIFT	5
AKAZE + BRISK	4	MSER + SURF	5	GFTT + SIFT	6
GFTT + DAISY	5	KAZE + FREAK	5	GFTT + BRIEF	6
FAST + BRISK	5	KAZE + SURF	6	MSD + FREAK	7

5. Conclusions

A comprehensive comparison of fifteen notable feature detection and description techniques for image stitching was carried out in the study. The following are some of the contributions made by this research: (1) The evaluations give a better picture of how feature detectors and descriptors work across a variety of image datasets; (2) cumulative comparisons for a complete set of 85 detector–descriptor pairs were performed over a number of datasets; and (3) a rank was assigned to each detector and descriptor pair for each dataset, which shows the relative performance of the detector and descriptor for that particular type of dataset. Overall, it was identified that the AKAZE detector paired with the AKAZE descriptor outperforms all other combinations for all types of selected datasets. The AGAST detector combined with the BRIEF descriptor also performs very well for dataset 3 containing indoor scenes.

In future, we can include more datasets, and a study of detectors and descriptors can be performed by varying parameters of each detector and descriptor. The ranking can also be made more application-specific by assigning suitable weights to each metric.

Author Contributions: Conceptualization, S.K.S. and K.J.; methodology, S.K.S.; software, S.K.S.; validation, S.K.S. and K.J.; formal analysis, S.K.S.; investigation, S.K.S.; writing—original draft preparation, S.K.S.; writing—review and editing, S.K.S. and A.K.S.; visualization, S.K.S. and A.K.S.; supervision, K.J. and A.K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

We also considered six publicly available datasets: (1) apartment dataset [77], (2) MATLAB dataset (MATLAB vision toolbox dataset), (3) synthetic dataset [78], (4) river dataset (VLFeat dataset), (5) garden dataset [77] and (6) UAV dataset [79]. Input images used from these datasets are shown in Figure A1. The stitched images obtained using different combinations of feature detectors and descriptors from these datasets are shown in Figures A2–A7.

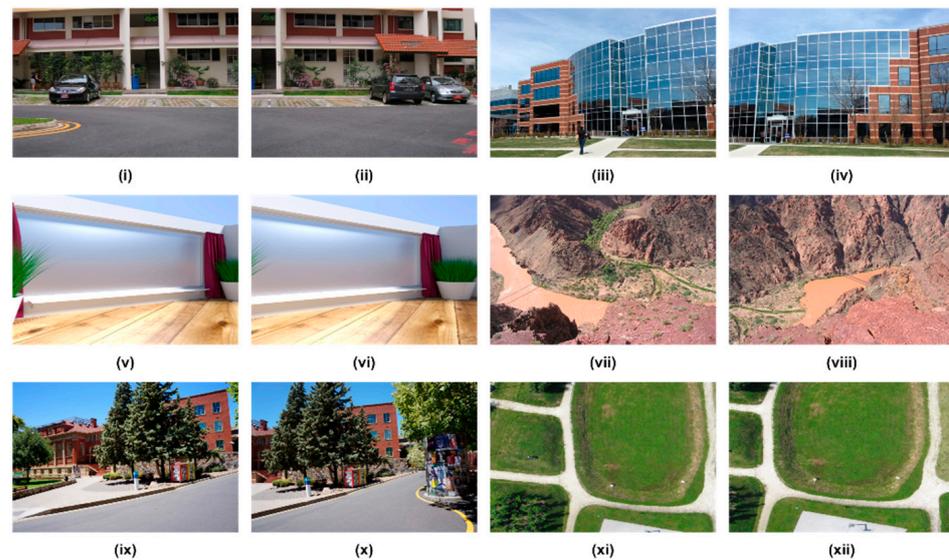


Figure A1. Input datasets: (i) and (ii) apartment dataset image pair; (iii) and (iv) MATLAB dataset image pair; (v) and (vi) synthetic dataset image pair; (vii) and (viii) river dataset image pair; (ix) and (x) garden dataset image pair; (xi) and (xii) UAV dataset.

Figure A2 shows a subset of output stitched images obtained using various detector–descriptor combinations for the apartment dataset. The first and second rows show a subset of stitched images obtained using different detector–descriptor combinations, which produces a good quality of stitched image in comparison to all other combinations. The third row shows a subset of stitched outputs using different detector–descriptor combinations, which produces poor quality of stitched images. It was observed that AKAZE + AKAZE produced a better quality of stitched image in comparison to all other combinations.

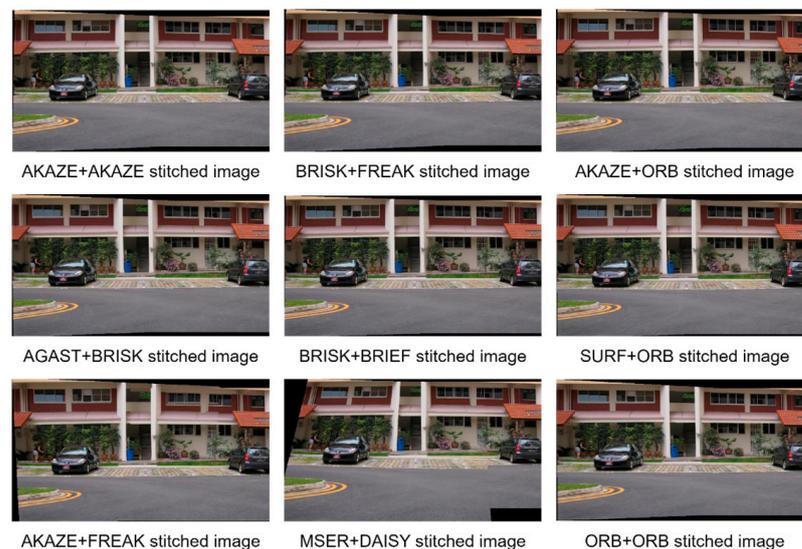


Figure A2. Image stitching outputs obtained for different feature detector–descriptor combinations for apartment dataset.

In Figure A3, a collection of stitched images generated from a range of detector–descriptor combinations on the MATLAB dataset is displayed. The first and second rows exhibit examples of high-quality stitched images, while the third row showcases a sample of poorly stitched outputs resulting from different detector–descriptor combinations. It was noted that the AKAZE + AKAZE combination produced the highest quality stitched image compared to all other combinations.

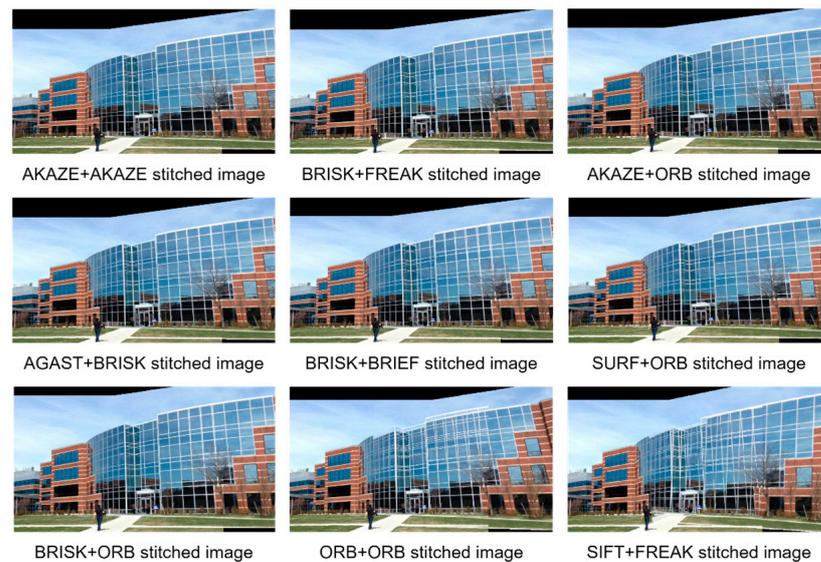


Figure A3. Image stitching outputs obtained for different feature detector–descriptor combinations for MATLAB dataset.

Figure A4 displays a variety of stitched images created from a selection of detector–descriptor combinations on the synthetic dataset. The first and second rows illustrate instances of well-stitched images, while the third row features a subset of poorly stitched outputs that resulted from various detector–descriptor combinations. The AKAZE + AKAZE and AGAST + BRIEF combinations were observed to produce the highest quality stitched images compared to all other combinations.

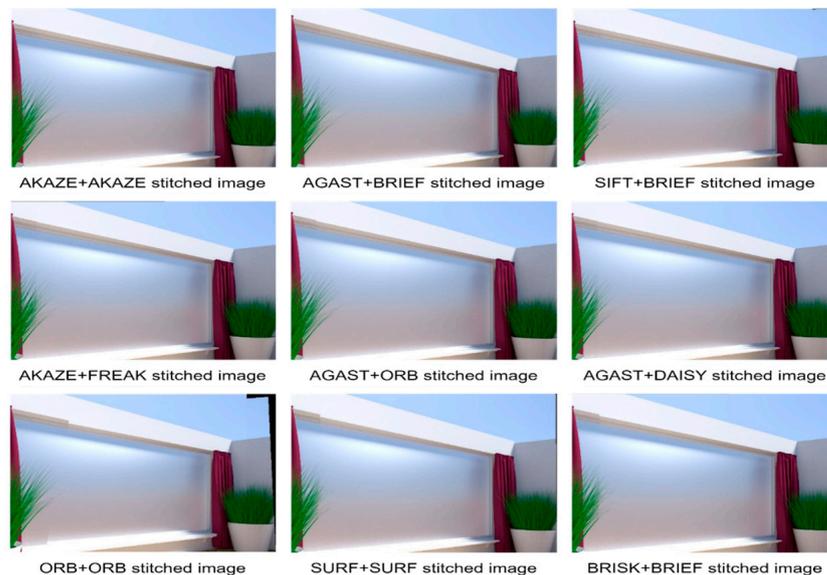


Figure A4. Image stitching outputs obtained for different feature detector–descriptor combinations for synthetic dataset.

Figure A5 showcases a range of stitched output images that were generated using various detector–descriptor combinations on the river dataset. The first and second rows display a subset of well-stitched outputs that outperformed all other combinations in terms of quality. Conversely, the third row exhibits a selection of poorly stitched outputs resulting from different detector–descriptor combinations. It was noted that the AKAZE + AKAZE and SURF + SURF combinations produced the highest quality stitched images among all the combinations tested.

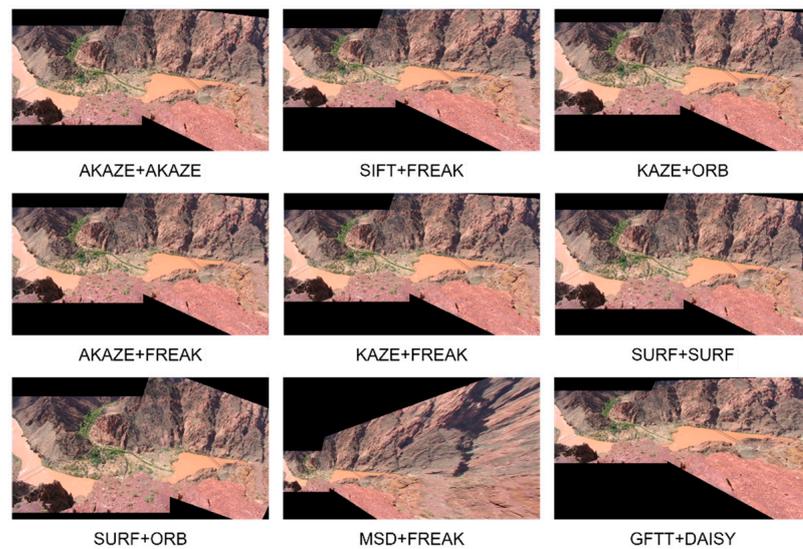


Figure A5. Image stitching outputs obtained for different feature detector–descriptor combinations for river dataset.

In Figure A6, a variety of stitched output images obtained from different detector–descriptor combinations on the garden dataset are showcased. The first and second rows exhibit a subset of well-stitched outputs that displayed superior quality compared to all other combinations. Conversely, the third row features a subset of poorly stitched outputs resulting from various detector–descriptor combinations. It was observed that the AKAZE + AKAZE combination produced the highest quality stitched image among all the combinations tested.

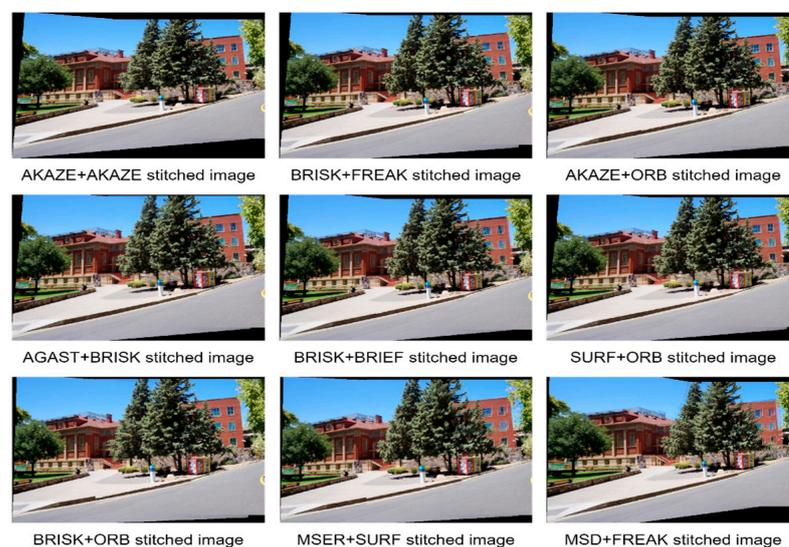


Figure A6. Image stitching outputs obtained for different feature detector–descriptor combinations for garden dataset.

Figure A7 displays a selection of stitched output images that were produced using different detector–descriptor combinations on the UAV dataset. The first and second rows exhibit a subset of well-stitched outputs that demonstrated superior quality compared to all other combinations. However, the third row features a subset of poorly stitched outputs resulting from various detector–descriptor combinations, as evidenced by the outputs in the last row. Among all the combinations tested, it was observed that the AKAZE + AKAZE produced the highest quality stitched image.

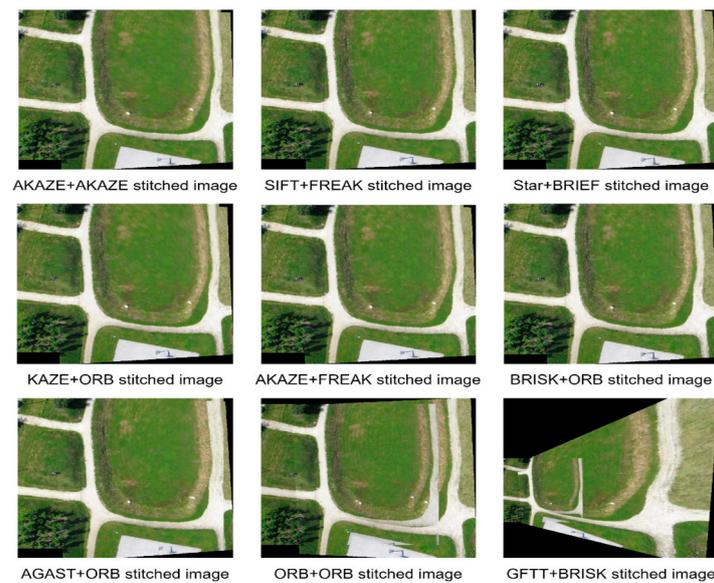


Figure A7. Image stitching outputs obtained for different feature detector–descriptor combinations for UAV dataset.

Appendix B

We evaluated two recent advanced image stitching algorithms that are based on deep learning techniques. These algorithms are: (1) A view-free image stitching network [80] and (2) unsupervised deep image stitching [81]. We utilized both of these algorithms to stitch images from all datasets under consideration. Figure A8 showcases the stitched images obtained using the view-free image stitching network algorithm, while Figure A9 presents those obtained through unsupervised deep image stitching. It was observed that while the deep learning-based algorithms produced acceptable results for synthetic datasets, the stitched images obtained using these algorithms for real-world datasets suffered from different types of distortions such as misalignment and projective distortions.

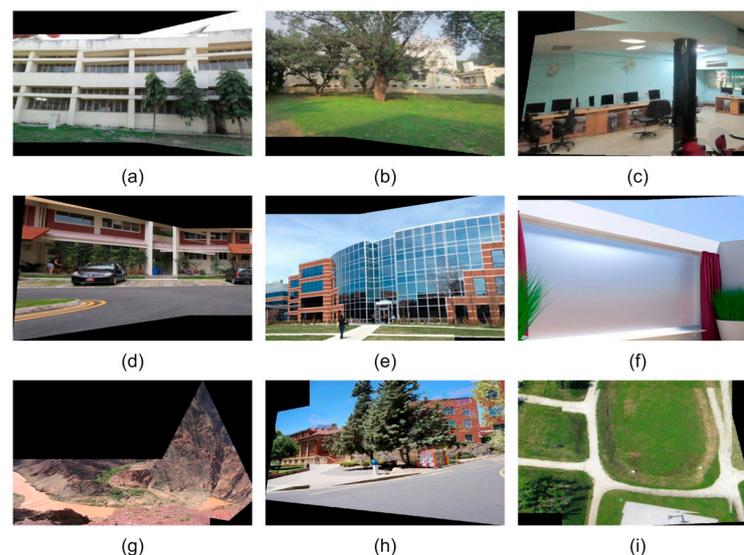


Figure A8. Image stitching outputs obtained using view-free image stitching network for different datasets: (a) stitched image dataset 1; (b) stitched image dataset 2; (c) stitched image dataset 3; (d) stitched image apartment dataset; (e) stitched image MATLAB dataset; (f) stitched image synthetic dataset; (g) stitched image river dataset; (h) stitched image garden dataset; (i) stitched image UAV dataset.

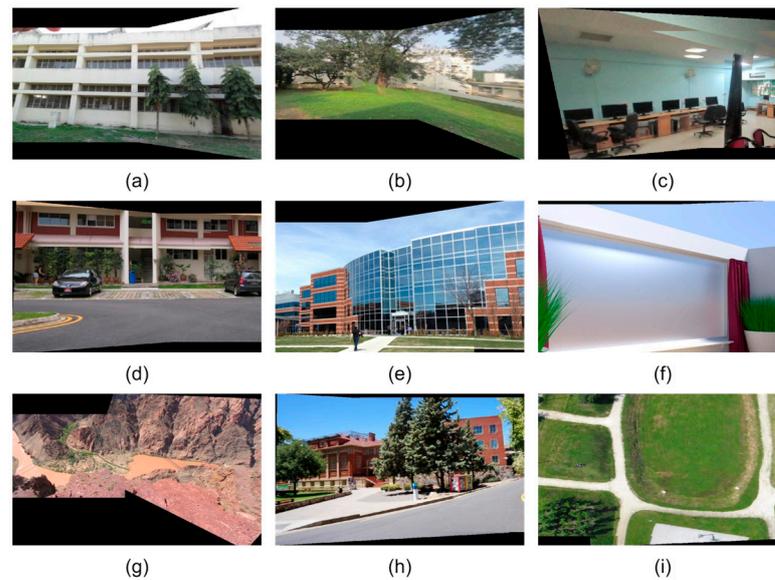


Figure A9. Image stitching outputs obtained using unsupervised deep image stitching algorithm for different datasets: (a) stitched image dataset 1; (b) stitched image dataset 2; (c) stitched image dataset 3; (d) stitched image apartment dataset; (e) stitched image MATLAB dataset; (f) stitched image synthetic dataset; (g) stitched image river dataset; (h) stitched image garden dataset; (i) stitched image UAV dataset.

References

1. Wang, J.; Fang, J.; Liu, X.; Zhao, D.; Xiao, Q. A fast mosaic method for airborne images: The new Template-Convolution Speed-Up Robust Features (TSURF) algorithm. *Int. J. Remote Sens.* **2014**, *35*, 5959–5970. [[CrossRef](#)]
2. Nguyen, T.L.; Byun, Y.; Han, D.; Huh, J. Efficient seamline determination for UAV image mosaicking using edge detection. *Remote Sens. Lett.* **2018**, *9*, 763–769. [[CrossRef](#)]
3. Kwokjet, R.; Curlander, J.C.; Pang, S.S. An automated system for mosaicking spaceborne SAR imagery. *Int. J. Remote Sens.* **1990**, *11*, 209–223. [[CrossRef](#)]
4. Rivard, B.; Toutin, T. A Mosaic of Airborne SAR Imagery for Geological Mapping in Rolling Topography. *Can. J. Remote Sens.* **1995**, *21*, 75–78. [[CrossRef](#)]
5. Majumdar, J.; Vanathy, B.; Varshney, N.; Rao, D.R.; Jalan, U.; Girish, R. Image Mosaicing from Video Sequences. *IETE J. Res.* **2002**, *48*, 303–310. [[CrossRef](#)]
6. Gheisari, M.; Sabzevar, M.F.; Chen, P.; Irizzary, J. Integrating BIM and Panorama to Create a Semi-Augmented-Reality Experience of a Construction Site. *Int. J. Constr. Educ. Res.* **2016**, *12*, 303–316. [[CrossRef](#)]
7. Morevec, H.P. Towards automatic visual obstacle avoidance. In Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, MA, USA, 22–25 August 1977; p. 584.
8. Förstner, W.; Gülch, E. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In Proceedings of the ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, 2–4 June 1987; pp. 281–305.
9. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the 4th Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; p. 151.
10. Tomasi, C.; Kanade, T. Detection and Tracking of Point Features. *Int. J. Comput. Vis.* **1991**, *9*, 137–154. [[CrossRef](#)]
11. Heitger, F.; Rosenthaler, L.; Von Der Heydt, R.; Peterhans, E.; Kübler, O. Simulation of neural contour mechanisms: From simple to end-stopped cells. *Vis. Res.* **1992**, *32*, 963–981. [[CrossRef](#)]
12. Förstner, W. A framework for low level feature extraction. In Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden, 2–6 May 1994; pp. 383–394.
13. Shi, J.; Tomasi, C. Good features to track. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600. [[CrossRef](#)]
14. Lindeberg, T. Junction detection with automatic selection of detection scales and localization scales. In Proceedings of the 1st International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994; Volume 1, pp. 924–928. [[CrossRef](#)]
15. Wang, H.; Brady, M. Real-time corner detection algorithm for motion estimation. *Image Vis. Comput.* **1995**, *13*, 695–703. [[CrossRef](#)]
16. Smith, S.M.; Brady, J.M. SUSAN—A New Approach to Low Level Image Processing. *Int. J. Comput. Vis.* **1997**, *23*, 45–78. [[CrossRef](#)]
17. Hall, D.; Leibe, B.; Schiele, B. Saliency of Interest Points under Scale Changes. In Proceedings of the British Machine Vision Conference (BMVC'02), Cardiff, UK, 2–5 September 2002; pp. 646–655.

18. Lindeberg, T. Feature Detection with Automatic Scale Selection. *Int. J. Comput. Vis.* **1998**, *30*, 79–116. [[CrossRef](#)]
19. Mikolajczyk, K.; Schmid, C. Indexing based on scale invariant interest points. *Proc. IEEE Int. Conf. Comput. Vis.* **2001**, *1*, 525–531. [[CrossRef](#)]
20. Schmid, C.; Mohr, R.; Bauckhage, C. Comparing and evaluating interest points. In Proceedings of the 6th International Conference on Computer Vision (IEEE Cat. No.98CH36271), Bombay, India, 7 January 1998; pp. 230–235. [[CrossRef](#)]
21. Schmid, C.; Mohr, R.; Bauckhage, C. Evaluation of Interest Point Detectors. *Int. J. Comput. Vis.* **2000**, *37*, 151–172. [[CrossRef](#)]
22. Lowe, D.G. Object recognition from local scale-invariant features. *Proc. IEEE Int. Conf. Comput. Vis.* **1999**, *2*, 1150–1157. [[CrossRef](#)]
23. Brown, M.; Lowe, D.G. Recognising panoramas. In Proceedings of the 9th International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 2, pp. 1218–1225. [[CrossRef](#)]
24. Brown, M.; Lowe, D.G. Automatic panoramic image stitching using invariant features. *Proc. Int. J. Comput. Vis.* **2007**, *74*, 59–73. [[CrossRef](#)]
25. Zhu, J.; Ren, M. Image mosaic method based on SIFT features of line segment. *Comput. Math. Methods Med.* **2014**, *2014*, 926312. [[CrossRef](#)]
26. Qu, Z.; Lin, S.P.; Ju, F.R.; Liu, L. The Improved Algorithm of Fast Panorama Stitching for Image Sequence and Reducing the Distortion Errors. *Math. Probl. Eng.* **2015**, *2015*, 428076. [[CrossRef](#)]
27. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630. [[CrossRef](#)]
28. Zuliani, M.; Kenney, C.; Manjunath, B.S. A mathematical comparison of point detectors. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, Washington, DC, USA, 27 June–2 July 2004. [[CrossRef](#)]
29. Gao, J.; Huang, X.; Liu, B. A quick scale-invariant interest point detecting approach. *Mach. Vis. Appl.* **2010**, *21*, 351–364. [[CrossRef](#)]
30. Feng, Y.; Ren, J.; Jiang, J.; Halvey, M.; Jose, J.M. Effective venue image retrieval using robust feature extraction and model constrained matching for mobile robot localization. *Mach. Vis. Appl.* **2011**, *23*, 1011–1027. [[CrossRef](#)]
31. Liao, K.; Liu, G.; Hui, Y. An improvement to the SIFT descriptor for image representation and matching. *Pattern Recognit. Lett.* **2013**, *34*, 1211–1220. [[CrossRef](#)]
32. Wu, J.; Cui, Z.; Sheng, V.S.; Zhao, P.; Su, D.; Gong, S. A comparative study of SIFT and its variants. *Meas. Sci. Rev.* **2013**, *13*, 122–131. [[CrossRef](#)]
33. Mikolajczyk, K.; Schmid, C. Scale & Affine Invariant Interest Point Detectors. *Int. J. Comput. Vis.* **2004**, *60*, 63–86. [[CrossRef](#)]
34. Mikolajczyk, K.; Tuytelaars, T.; Schmid, C.; Zisserman, A.; Matas, J.; Schaffalitzky, F.; Kadir, T.; Van Gool, L. A Comparison of Affine Region Detectors. *Int. J. Comput. Vis.* **2005**, *65*, 43–72. [[CrossRef](#)]
35. Moreels, P.; Perona, P. Evaluation of features detectors and descriptors based on 3D objects. In Proceedings of the 10th International Conference on Computer Vision (ICCV'05), Beijing, China, 17–21 October 2005; Volume 1, pp. 800–807. [[CrossRef](#)]
36. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767. [[CrossRef](#)]
37. Nistér, D.; Stewénius, H. Linear Time Maximally Stable Extremal Regions. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5303, pp. 183–196. ISBN 3540886850.
38. Bay, H.; Tuytelaars, T.; Gool, L. Van SURF: Speeded Up Robust Features. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3951, pp. 404–417. [[CrossRef](#)]
39. Niu, J.; Yang, F.; Shi, L. Improved method of automatic image stitching based on SURF. In Proceedings of the 1st International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), Jinhua, China, 1–3 July 2013. [[CrossRef](#)]
40. Zhu, L.; Wang, Y.; Zhao, B.; Zhang, X. A fast image stitching algorithm based on improved SURF. In Proceedings of the 10th International Conference on Computational Intelligence and Security, Kunming, China, 15–16 November 2014; pp. 171–175. [[CrossRef](#)]
41. Yang, Z.; Shen, D.; Yap, P.-T. Image mosaicking using SURF features of line segments. *PLoS ONE* **2017**, *12*, e0173627. [[CrossRef](#)]
42. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3951, pp. 430–443. [[CrossRef](#)]
43. Tuytelaars, T.; Mikolajczyk, K. Local invariant feature detectors: A survey. *Found. Trends Comput. Graph. Vis.* **2007**, *3*, 177–280. [[CrossRef](#)]
44. Li, J.; Allinson, N.M. A comprehensive review of current local features for computer vision. *Neurocomputing* **2008**, *71*, 1771–1787. [[CrossRef](#)]
45. Govender, N. Evaluation of feature detection algorithms for structure from motion. In Proceedings of the 3rd Robotics and Mechatronics Symposium (ROBMECH 2009), Pretoria, South Africa, 8–10 November 2009; p. 4.
46. Gil, A.; Mozos, O.M.; Ballesta, M.; Reinoso, O. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Mach. Vis. Appl.* **2009**, *21*, 905–920. [[CrossRef](#)]
47. Hartmann, J.; Klussendorff, J.H.; Maehle, E. A comparison of feature descriptors for visual SLAM. In Proceedings of the European Conference on Mobile Robots, Barcelona, Spain, 25–27 September 2013; pp. 56–61. [[CrossRef](#)]

48. Urban, S.; Weinmann, M. Finding a Good Feature Detector-Descriptor Combination for the 2D Keypoint-Based Registration of TLS Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *2*, 121–128. [[CrossRef](#)]
49. Agrawal, M.; Konolige, K.; Blas, M.R. CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5305, pp. 102–115. [[CrossRef](#)]
50. Willis, A.; Sui, Y. An algebraic model for fast corner detection. In Proceedings of the 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2296–2302. [[CrossRef](#)]
51. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary Robust Independent Elementary Features. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6314, pp. 778–792. [[CrossRef](#)]
52. Tola, E.; Lepetit, V.; Fua, P. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 815–830. [[CrossRef](#)] [[PubMed](#)]
53. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust invariant scalable keypoints. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555. [[CrossRef](#)]
54. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571. [[CrossRef](#)]
55. Alahi, A.; Ortiz, R.; Vanderghyest, P. FREAK: Fast retina keypoint. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 510–517. [[CrossRef](#)]
56. Alcantarilla, P.F.; Bartoli, A.; Davison, A.J. KAZE Features. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7577, pp. 214–227. [[CrossRef](#)]
57. Alcantarilla, P.F.; Nuevo, J.; Bartoli, A. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In Proceedings of the Tilo Burghardt, Dima Damen, Walterio W. Mayol-Cuevas, Majid Mirmehdi: British Machine Vision Conference, BMVC 2013, Bristol, UK, 9–13 September 2013. [[CrossRef](#)]
58. Sharma, S.K.; Jain, K. Image Stitching using AKAZE Features. *J. Indian Soc. Remote Sens.* **2020**, *48*, 1389–1401. [[CrossRef](#)]
59. Ghosh, D.; Kaabouch, N. A survey on image mosaicing techniques. *J. Vis. Commun. Image Represent.* **2016**, *34*, 1–11. [[CrossRef](#)]
60. Wang, Z.; Yang, Z. Review on image-stitching techniques. *Multimed. Syst.* **2020**, *26*, 413–430. [[CrossRef](#)]
61. LYU, W.; ZHOU, Z.; CHEN, L.; ZHOU, Y. A survey on image and video stitching. *Virtual Real. Intell. Hardw.* **2019**, *1*, 55–83. [[CrossRef](#)]
62. Megha, V.; Rajkumar, K.K. A Comparative Study on Different Image Stitching Techniques. *Int. J. Eng. Trends Technol.* **2022**, *70*, 44–58. [[CrossRef](#)]
63. Heinly, J.; Dunn, E.; Frahm, J.M. Comparative Evaluation of Binary Features. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7573, pp. 759–773. [[CrossRef](#)]
64. Noble, F.K. Comparison of OpenCV's feature detectors and feature matchers. In Proceedings of the 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Nanjing, China, 28–30 November 2016. [[CrossRef](#)]
65. Işık, Ş.; Özkan, K. A Comparative Evaluation of Well-known Feature Detectors and Descriptors. *Int. J. Appl. Math. Electron. Comput.* **2015**, *3*, 1–6. [[CrossRef](#)]
66. Ziegler, A.; Christiansen, E.; Kriegman, D.; Belongie, S. Locally Uniform Comparison Image Descriptor. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25.
67. Mukherjee, D.; Jonathan Wu, Q.M.; Wang, G. A comparative experimental study of image feature detectors and descriptors. *Mach. Vis. Appl.* **2015**, *26*, 443–466. [[CrossRef](#)]
68. Aanæs, H.; Dahl, A.L.; Pedersen, K.S. Interesting Interest Points. *Int. J. Comput. Vis.* **2011**, *97*, 18–35. [[CrossRef](#)]
69. Dahl, A.L.; Aanæs, H.; Pedersen, K.S. Finding the best feature detector-descriptor combination. In Proceedings of the 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Hangzhou, China, 16–19 May 2011; pp. 318–325. [[CrossRef](#)]
70. Mair, E.; Hager, G.D.; Burschka, D.; Suppa, M.; Hirzinger, G. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6312, pp. 183–196. [[CrossRef](#)]
71. Tombari, F. Interest Points via Maximal Self-Dissimilarities. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 9004, pp. 586–600. [[CrossRef](#)]
72. Opozda, S.; Sochan, A. The survey of subjective and objective methods for quality assessment of 2D and 3D images. *Theor. Appl. Inform.* **2014**, *26*, 39–67.
73. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
74. Zhang, L.; Zhang, L.; Mou, X.; Zhang, D. FSIM: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **2011**, *20*, 2378–2386. [[CrossRef](#)] [[PubMed](#)]

75. Zhang, L.; Shen, Y.; Li, H. VSI: A visual saliency-induced index for perceptual image quality assessment. *IEEE Trans. Image Process.* **2014**, *23*, 4270–4281. [[CrossRef](#)] [[PubMed](#)]
76. Sharma, S.K.; Jain, K.; Suresh, M. Quantitative evaluation of panorama softwares. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 500, pp. 543–561. [[CrossRef](#)]
77. Zaragoza, J.; Chin, T.J.; Brown, M.S.; Suter, D. As-projective-as-possible image stitching with moving DLT. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2339–2346. [[CrossRef](#)]
78. Meneghetti, G.; Danelljan, M.; Felsberg, M.; Nordberg, K. Image Alignment for Panorama Stitching in Sparsely Structured Environments. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9127, pp. 428–439. [[CrossRef](#)]
79. Yahyanejad, S.; Rinner, B. A fast and mobile system for registration of low-altitude visual and thermal aerial images using multiple small-scale UAVs. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 189–202. [[CrossRef](#)]
80. Nie, L.; Lin, C.; Liao, K.; Liu, M.; Zhao, Y. A view-free image stitching network based on global homography. *J. Vis. Commun. Image Represent.* **2020**, *73*, 102950. [[CrossRef](#)]
81. Nie, L.; Lin, C.; Liao, K.; Liu, S.; Zhao, Y. Unsupervised Deep Image Stitching: Reconstructing Stitched Features to Images. *IEEE Trans. Image Process.* **2021**, *30*, 6184–6197. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.