

# Research of different feature detection and matching algorithms on panoramic image

**Jindong Xiao**

School of Software Engineering, Shenzhen University, Guangzhou, China

2020151007@email.szu.edu.cn

**Abstract.** Image stitching is the process of combining numerous photos to make a panorama. The technique of image stitching has rapidly advanced and grown to be a significant area of digital image processing. Many image stitching methods have been proposed and studied in prior study. In this paper, the image stitching process is implemented using different algorithms. For keypoints detection, the algorithms of Harris corner detection, SIFT(Scale-Invariant Feature Transform), SURF(Speeded Up Robust Feature) and ORB(Oriented FAST and Rotated BRIEF) algorithms are applied, then use different methods (i.e., Brute Force, etc.) for feature matching. The RANSAC(Random Sample Consensus) method is used to calculate a homography matrix from matched feature vectors and use it to warp the images. Image blending and cropping methods are proposed to enhance the image quality. Given groups of the self-captured images, experiments have been down to shown the performance of different techniques.

**Keywords:** panoramic mosaic, SIFT, Harris, SURF, ORB, blending, warping, homography.

## 1. Introduction

Image stitching is the process of integrating multiple images with overlapping fields of vision to create a segmented panorama image. This method is a crucial component of digital picture processing and is widely used in fields such as remote sensing, aerospace, virtual reality, and medical imaging. The process involves feature detection and extraction, feature matching, using RANSAC(Random Sample Consensus) to estimate homography matrix, image warping, alignment, and image blending and cropping. In previous years, significant progress has been made in each of these steps. For example, algorithms like Harris corner, SIFT(Scale-Invariant Feature Transform), SURF(Speeded Up Robust Feature), and ORB(Oriented FAST and Rotated BRIEF) have been developed for feature detection and extraction for satisfying the increasing demands of time and precision, while image blending and cropping techniques are used to enhance image quality, for obtaining an image with excellent stitching which clear, has no black edge [1].

In this paper authors apply different algorithms of feature matching, and explore the methods of image blending and cropping, after the estimation of homography matrix using RANSAC and the image warping between two images. Meanwhile, we consider the order of the stitching by making the central image to be fixed and warping the images under the analysis of the positions for better image stitching results.

The remainder of the paper is divided into the following parts. Section 2 shows the methods of feature detection and descriptor extractions, feature matching algorithms, image stitching process, and image

blending and cropping methods. Section 3 illustrate our image matching methodology by using the example given the input groups of images, the difference of the results is explored and analyzed. We offer our findings and suggestions in Section 4.

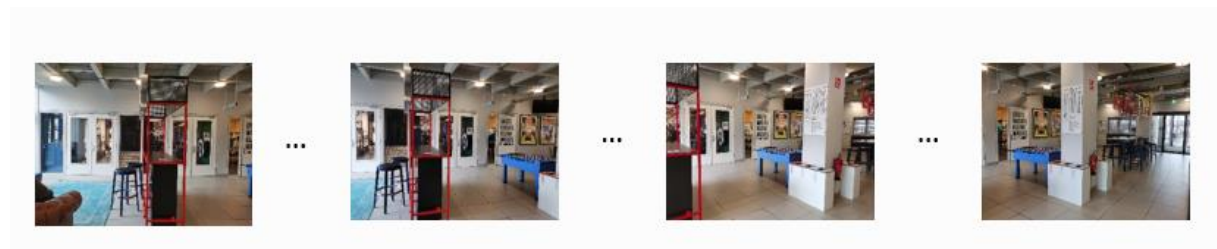
## 2. Method and technology

### 2.1. Problem description

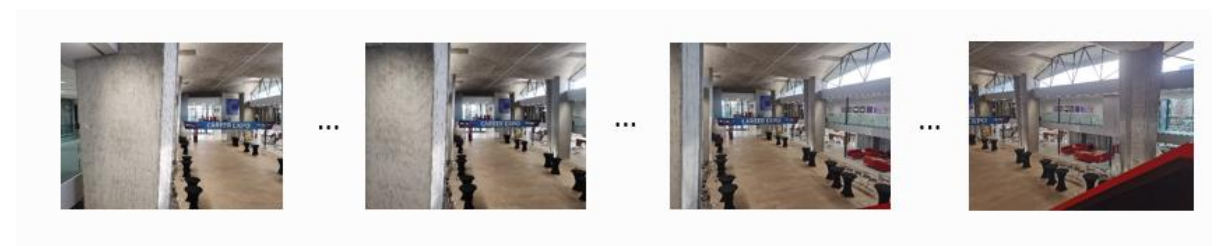
Panoramic image stitching, in which many photos are stitched together to generate a larger, wider image, is a useful approach in many industries. Yet, there are numerous obstacles in picture alignment and fusion that must be overcome in order to achieve high-quality panoramic image stitching. Accurate feature point recognition and matching, handling various perspective distortions and lighting variations, and handling lens distortion are important considerations. This paper focuses on the stitching of multiple pictures using feature point detection and matching algorithms and evaluates the performance of several approaches. This procedure is implemented using Python and OpenCV.

### 2.2. Input data

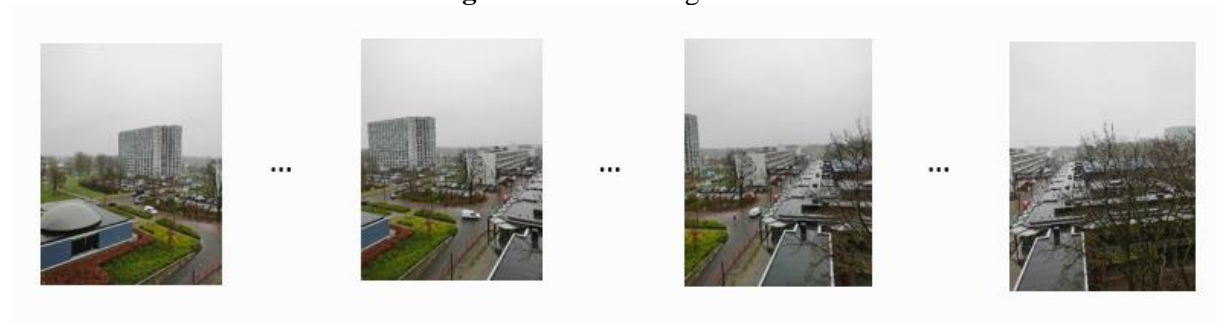
The input data is linear multiple images captured by same camera with overlapping areas. (Figure 1- Figure 3). Three data sets are selected and captured in different condition. Figure 1 and Figure 2 are indoor images set. Figure 3 is outdoor images set. Three sets of images have different lighting conditions, shooting angles, and sceneries, which also help to evaluate the adaptability and universality of the algorithms. The number of images in each set are all above 10 (about 10 to 12 images).



**Figure 1.** Indoor images set 1.



**Figure 2.** Indoor images set 2.



**Figure 3.** Outdoor images set.

### 2.3. Feature detection and extraction

**2.3.1. Harris corner detector.** The Harris uses the intensity fluctuation in a local neighborhood to identify points. A small area close to the feature should display a significant intensity change compared to windows that have been moved in either direction [2]. There are procedures of Harris corner detector in table 1.

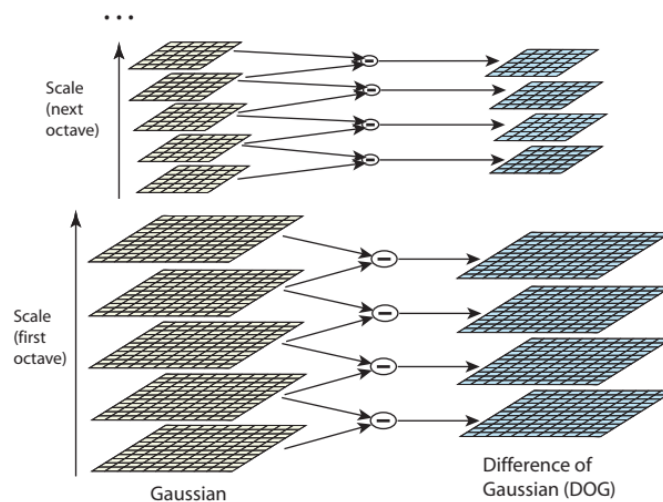
**Table 1.** The procedures of Harris.

The procedures of Harris	
1)	Gradient calculation: Use a filter like Sobel, Scharr, or Prewitt to calculate the image gradients in the x and y directions.
2)	Structure tensor computation: Using the gradients, compute a 2x2 structure tensor for each pixel. The local image structure surrounding the pixel is encoded by the structure tensor, which is a matrix.
3)	Corner response calculation: Use the Harris corner detector to determine the corner response function for each pixel.
4)	Non-maximum suppression: Suppress non-maximum responses in the image to produce a set of local maxima that correspond to corner locations
5)	Thresholding: Choose the most prominent corners by applying a threshold to the corner response values.

**2.3.2. SIFT algorithm.** SIFT (Scale-Invariant Feature Transform) is a well-liked feature detection approach in computer vision and is used for a range of applications like picture matching, object recognition, and 3D reconstruction. It mainly has four stages to get the set of image features [3].

#### A. Scale-space peak selection

Using scale-space peak selection, the system can identify features at various scales. The image is convolved with a succession of Gaussian filters at progressively larger sizes to produce the scale-space representation. By removing adjacent levels from the Gaussian pyramid, the SIFT method creates a difference-of-Gaussian (DoG) pyramid from the scale-space representation as Figure 4. The DoG pyramid acts as a scale-invariant feature detector and draws attention to areas of the picture that have high contrast and curvature.



**Figure 4.** Difference-of-Gaussian (DoG) pyramid from the scale-space representation (from [2]).

#### B. Keypoint localization

SIFT recognizes probable keypoints as local DoG function scale and space extrema at each level of the pyramid. A pixel's 26 immediate neighbors in the level it is now in and 9 immediate neighbors in the levels above it is used to determine the local extrema. By calculating the difference between the pixel and its neighbors and determining whether the pixel is the maximum or minimum in its neighborhood, a comparison is made.

#### C. Orientation assignment

By calculating the orientation of the gradient histogram, the main orientation can be determined, which gives the SIFT algorithm rotational invariance [4].

#### D. Keypoint descriptor

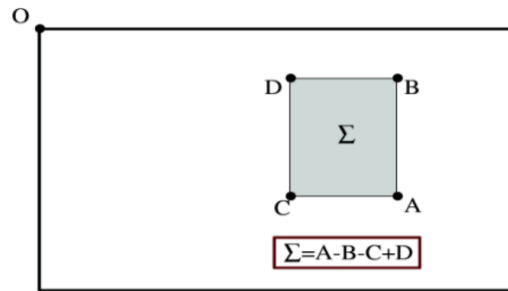
SIFT generates scale, orientation, and translation invariant key point descriptors by computing a gradient orientation histogram of the picture around the key point location.

**2.3.3. SURF algorithm.** The SURF (Speeded Up Robust Feature) algorithm approximate the DoG (Difference of Gaussians) using box filters instead of image Gaussian averaging, because integral image convolution using squares is quicker [5]. Additionally, the SURF algorithm accelerates calculations by using fast approximations of the Hessian matrix and descriptors through the use of integral images [6]. There are mainly 2 steps in SURF:

##### A. Interest point detection

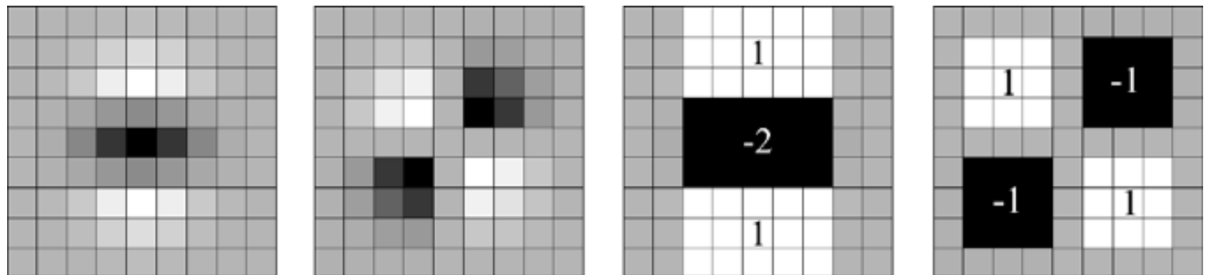
In the SURF algorithm, the original image is converted into a composite image, where  $I_{\Sigma}(x,y)$  represents the total amount of pixels in a rectangle whose top-left corner is (0,0) and bottom-right corner is (x,y) (Figure 5) [7]. This is achieved by using only four array references, thus allowing for efficient computation of the total pixel sum in any rectangular region through the use of integral images.

$$I_{\Sigma}(x,y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x,y) \quad (1)$$



**Figure 5.** Using integral images.

The SURF detector relies on the Hessian matrix's positive determinant, which is calculated by convolving the picture with an appropriate kernel to obtain the second-order partial derivatives of a function. Figure 6 [6] illustrates the convolution of the integral image with a box filter.



**Figure 6.** The integral image convoluted with box filter.

#### B. Interest point description

There are two steps in the construction of the SURF descriptor:

##### a. The orientation assignment.

For each keypoint, it is necessary to determine its main orientation. Haar wavelet responses can be computed in the image region surrounding the keypoint to provide rotation invariance, and the main orientation can be determined using these responses.

##### b. Extract the descriptor.

Once the main orientation of a keypoint is determined, it is necessary to compute its local feature descriptor. The SURF algorithm uses a technique called accelerated integral images to calculate the local feature descriptor.

**2.3.4. ORB algorithm.** ORB (Oriented FAST and Rotated BRIEF) is a swift binary descriptor that combines Binary Robust Independent Elementary Features (BRIEF) keypoints with FAST (Features from Accelerated Segment Test) detectors to create an efficient and robust method for feature detection and description [8]. There are two main steps of ORB algorithm:

#### A. Oriented FAST corner detection

ORB initially uses the FAST corner detector to find keypoints in the image. FAST is a method for detecting corners characterized by its fast computation speed and stability across images of different scales.

#### B. rBRIEF description

After obtaining the Oriented FAST keypoints, the ORB algorithm uses an enhanced version of the BRIEF algorithm. BRIEF is a binary vector descriptor composed of a series of 0s and 1s, offering a compact and efficient representation of the features [9]. ORB encodes the local region around keypoints using BRIEF descriptors. The BRIEF descriptor is combined with the orientation data of the keypoints in ORB's rotation-invariant BRIEF descriptor, which achieves rotation invariance.

### 2.4. Feature matching

The matching procedure compares descriptor data between matching points in two images. The locations of identical features are recognized as matched pairs if the features in the input images match. Matching algorithms such as Brute Force (BF), Fast Library for Approximate Nearest Neighbors (FLANN), and K-Nearest Neighbors matcher (KNN) [10] are used in this study.

#### A. BF matcher:

The BF matcher considers all possible matches and selects the best matches from the initial set. It compares one feature from the first image to all features in the second image by measuring the distance between them [10].

#### B. FLANN matcher:

The FLANN uses a custom algorithm library that can efficiently search for nearest neighbors in high-dimensional feature space. It employs random KD tree and k-means tree algorithms to conduct a prioritized search. The random KD algorithm swiftly locates the nearest points to a given input point by conducting parallel tree searches. The priority search K-means tree algorithm segments data into regions and reorganizes them until each leaf node contains more than M elements. It selects the initial center randomly, making it faster than the BF matcher for large datasets [10].

#### C. KNN matcher:

KNN matcher is a matching algorithm that finds the best matches between features in two images based on the k-nearest neighbors. It displays the k-best matches, where k is determined by the user. The algorithm generates lines from the features in the first image to the matching best match in the second image after stacking the two photos horizontally [10].

### 2.5. Image stitching using homography

Once we have obtained the feature matching information for all images, we can utilize this for image matching. During the image matching process, RANSAC algorithm is employed to estimate a homography matrix.

RANSAC (Random Sample Consensus) [11] is an iterative model fitting algorithm used to estimate model parameters from data with a lot of outliers (such as noise or incorrect matching points). Table 2 is the mainly procedure of RANSAC.

**Table 2.** RANSAC algorithm procedure.

<b>RANSAC algorithm</b>	
1)	Select n data points randomly from them;
2)	Estimate parameter x to calculate the transformation matrix;
3)	Use this data point to fit a model;
4)	The remaining data points' distance from the model should be calculated. An outlier point is one when the distance is greater than the cutoff. An intra-office point is one where the value does not exceed the threshold. Consequently, identify the model's corresponding intra-office point value.

After the homography matrix, we apply a warping transformation to stitch the images. The steps are as follows:

- A. Get the height and width of two images.
- B. Extract the coordinates of four corners from each image.
- C. Apply homography transformation to the corner points of the source image, which is the image to be warped.
- D. Concatenate the corner points of two images, and then find the minimum x and y coordinates as well as the maximum y coordinate, which are used for image stitching.
- E. Since we consider the central image is fixed, we assume that if the top-left corner of the source image has a coordinate less than 0, then it is stitched to the left side of destination image; otherwise, it is stitched to the right side.

### 2.6. Blending and cropping

#### A. Blending

Non-binary alpha mask is used to blend two images. The method is to select a suitable size window to blur the seam of image stitching and use non-binary alpha mask to blend two images. the value of the alpha mask image is between 0 and 1, representing the proportion of the weight of each pixel. For each pixel, the original image and overlay image pixel values are linearly blended according to the weight in the alpha mask. Using following formula to compute the value of each pixel:

$$F(i, j) = \omega_1 A(i, j) + \omega_2 B(i, j) \quad (2)$$

An important thing of non-binary alpha mask is to choose an appropriate size of window, it makes resulting images smooth but no ghosting. One eighth of destination image is chosen in this project.

Obviously,  $\omega_1 + \omega_2 = 1$ . At the seam of image stitching, the value of  $\omega_1$  and  $\omega_2$  are both between 0 and 1. Otherwise,  $\omega_1 = 1$  and  $\omega_2 = 0$ , meaning that it's the A image part.  $\omega_1 = 0$  and  $\omega_2 = 1$ , meaning that it's the B image part.

#### B. Cropping

There are also some black edges because of warping transformation. So it needs to implemented a function to crop image. There are several steps for cropping in table 3.

**Table 3.** The procedure of cropping.

The procedure of cropping
<ol style="list-style-type: none"> <li>1) Determine the corners' minimum and maximum x and y coordinates (4 corners of the warped image and 4 corners of the destination image).</li> <li>2) Determine the translation vector <math>t</math> (<math>t = [-x_{min}, -y_{min}]</math>) that will be used to get the displacement of the image.</li> <li>3) The warped picture is stitched to the left side of the target image if the x-coordinate of the top-left corner of the warped image is smaller than 0. Otherwise, is stitched to the right side.</li> <li>4) Then using these corners, the edge of cropped image can be calculated without black edge.</li> </ol>

Thus, the overall process of image stitching is as follows:

**Table 4.** Image stitching procedure.

Image stitching
<ol style="list-style-type: none"> <li>1) Read the input image;</li> <li>2) Detect key feature points of images 1 and 2, using algorithms of Harris corner, SIFT, SURF, ORB respectively, and calculate feature descriptors; <ul style="list-style-type: none"> <li>▪ Build Harris corner, SIFT, SURF, ORB generator;</li> <li>▪ Detect Harris corner, SIFT, SURF, ORB feature points and calculate descriptors;</li> <li>▪ Return the set of feature points and remember the corresponding description feature.</li> </ul> </li> <li>3) Set up matcher;</li> <li>4) Use BF and KNN to detect SIFT feature matching pairs from images 1 and 2, <math>K=2</math>;</li> <li>5) When the matching point pairs after screening are larger than 4, the view transformation matrix is calculated, and <math>H</math> is the view transformation matrix of <math>3 \times 3</math>;</li> <li>6) Match all the feature points of the two images and return the matching result;</li> <li>7) If the return result is empty, it proves that there is no feature point matching the result and exits the program;</li> <li>8) Otherwise, the matching result is extracted;</li> <li>9) Transform the view Angle of image 1, and the result is the transformed image;</li> <li>10) Judge if the warped image 1 is on the left or the right side of the source image, stitch the images.</li> </ol>

### 3. Experiment

#### 3.1. Platform introduction

The platform configuration parameters used in this report are the compiler and version pycharm, using OpenCV in the python language as a framework, creating a virtual environment using Anaconda, and installing Python 3.10.0 and 3.6 in the virtual environment for testing different algorithms.

#### 3.2. Results illustration

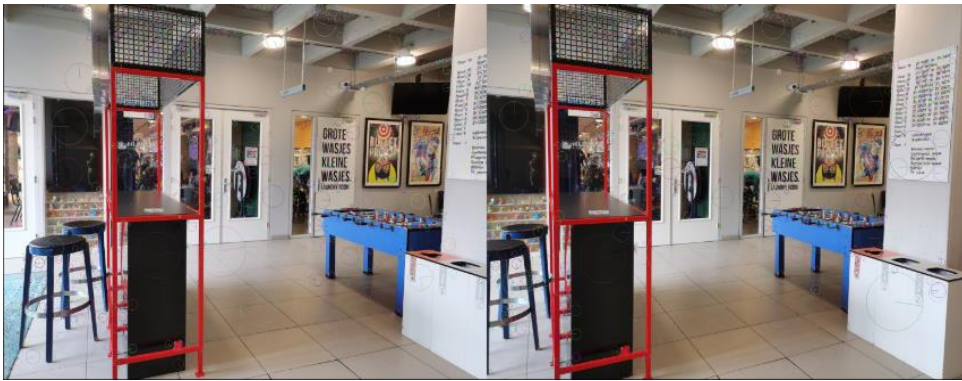
##### 3.2.1. KeyPoints detection

Figure 7 shows the results of the keypoints detection of the 1<sup>st</sup> and 2<sup>nd</sup> images of indoor image set 1, applying different feature detection methods.

Harris corner detection:



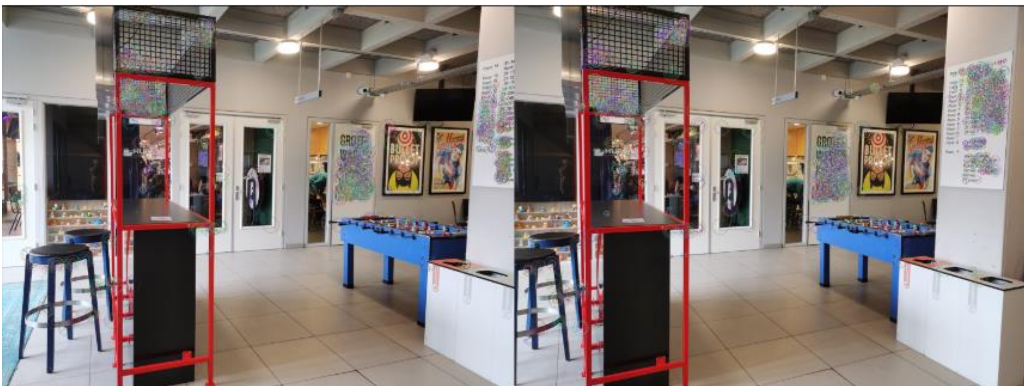
- SIFT:



- SURF:



- ORB:



**Figure 7.** Keypoints detections results (indoor images set 1).

### 3.2.2. Feature matching

Figure 8 shows the results of the feature matching of the 1<sup>st</sup> and 2<sup>nd</sup> images of indoor image set 1, given the detected keypoints and corresponding descriptors obtained in 3.2.1.

Harris corner detection + SIFT descriptors:



- SIFT:



- SURF:



- ORB:



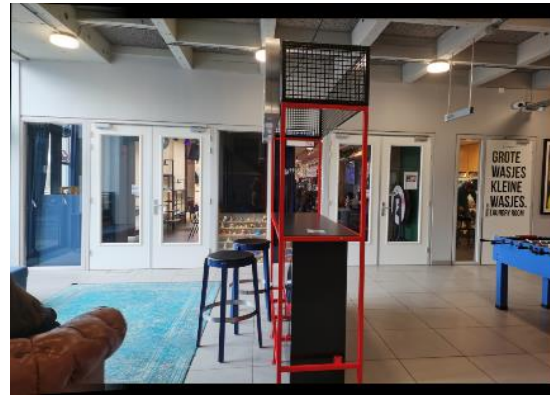
**Figure 8.** Feature matching results (indoor images set 1).

### 3.2.3. Image stitching

To obtain a well-stitched image which clear, smooth edge and high resolution, we apply image blending and cropping techniques to enhance image quality. Figure 9 shows the results of comparison with whether image blending and cropping are conducted.



(a) Before blending



(b) After blending



(a) Before cropping



(b) After cropping

**Figure 9.** Image blending and cropping results (indoor images set 1).

The results shows that image turns to smooth, because the total weight of all pixels in final images equals one after blending, instead of adding two images directly. It is obvious that black edge is eliminated after cropping, indicating that the four corners of the final image have been correctly identified.

### 3.2.4. Panorama results

Following the image stitching process described in Table 4, the panorama results using different detection and extraction algorithms and matching methods are shown in Figure 10.

- SIFT



- SURF:



- ORB



**Figure 10.** Panorama results (indoor images set 1).

From the results, we can see that the SIFT performs better than SURF and ORB, and SURF results better than ORB. The different of the 3 methods is reflected by the stitching result of the VIP area among the 3 panoramas.

### 3.2.5. Comparison and analysis of results

In this experiment, the feature matching accuracy percentages of four feature matching methods were analyzed. The accuracy percentage is calculated as  $(\text{number of correct matches} / \text{total number of matches}) * 100$ . RANSAC was used to identify correct matches and incorrect matches. Meanwhile, the processing time is measured to compare the efficiency of different algorithms. The data used consisted of the first two images from Indoor images set 1, FLANN matcher is used to match points. The table records the number of feature points and matched points searched in images 1 and 2, the number of correct matches after processing with the RANSAC algorithm, as well as the accuracy and processing time.

**Table 5.** Compares the feature matching accuracy percentages of four algorithms (the number of searched feature points was controlled).

	feature points (first image)	feature points (second image)	Match points	Correct matches (After ransac)	Matching rate	processing time
Harris (SIFT descriptors)	5000	5000	2543	1439	56.58%	2.6307s
SIFT	5001	5000	1426	641	44.95%	3.3216s
SURF	4713	5145	1425	735	51.58%	4.2654s
ORB	5000	5000	657	159	24.20%	2.0938s

**Table 6.** Compares the feature matching accuracy percentages of four algorithms (the number of searched feature points was not controlled).

	feature points (first image)	feature points (second image)	Match points	Correct matches (After ransac)	Matching rate	processing time
Harris (SIFT descriptors)	7036	5841	3312	1574	47.52%	3.6250s
SIFT	23668	20637	6088	2923	48.01%	23.5718s
SURF	35626	34833	10154	5406	53.24%	36.2369s
ORB	500	500	54	42	77.78%	1.5249s

In Table 5, For each algorithm, the number of searched feature points was controlled to be around 5000 to compare the search time. Harris with SIFT descriptors has best performance on matching rate (56.58%), followed by SURF (51.58%) and SIFT (44.95%). The matching rate of ORB only is 24.20%, which is lowest among four algorithms. While the ORB algorithm was the fastest in terms of processing time and SURF is slowest.

In Table 6, the number of searched feature points was not controlled in order to allow each algorithm to fully demonstrate its performance. It is obvious that ORB has fastest speed (1.5249s) and matching rate (77.78%). Compared with ORB performance in Table 6, it is more accurate when the number of features is small. Besides, Harris performs better when feature points was controlled, which means given an appropriate threshold, the Harris corner detection algorithm can identify more accurate corners and eliminate some false detections.

Considering that the experimental results may be affected by factors such as image content, transformation, and noise, outdoor dataset was also tested (Table 7).

**Table 7.** Compares the feature matching accuracy percentages of four algorithms (Outdoor image set).

	feature points (first image)	feature points (second image)	Match points	Correct matches (After ransac)	Matching rate	processing time
Harris (SIFT descriptors)	21719	18852	4683	3144	67.14%	18.3387s
SIFT	74797	58616	7022	6230	88.72%	185.1894s
SURF	54856	56016	6332	5038	79.56%	78.6469s
ORB	500	500	57	44	77.19%	1.6513s

In the experimental results, SIFT took the longest time but had the highest accuracy, SURF was more than twice as fast as SIFT while maintaining a certain level of accuracy, the corner detection accuracy was not high but the processing time was greatly reduced, and ORB remained the fastest detection algorithm with the least number of detected feature points.

Overall, using SIFT and SURF can find more feature points and consume more time. However, if a sufficient number of feature points is needed to ensure the quality of image stitching, they can achieve higher accuracy than the ORB algorithm.

#### 4. Conclusion

In this paper, SIFT, SURF, ORB, and Harris corner detection is employed as the techniques for feature extraction and matching. For feature matching, BF matching and FLANN matching is employed, then using the RANSAC algorithm to estimate the homography matrix, design transformation methods to align the images, and finally using image blending and cropping techniques to obtain a smooth and high-resolution panorama. In the above process, it is concluded that SIFT algorithm performs better than SURF and ORB in terms of image results. SIFT might be more suited for uses where high precision is essential and when computing expense is not a limiting factor. Meanwhile, SURF might be more suited for real-time applications where both accuracy and speed are crucial. ORB may be preferred over SIFT and SURF depending on the particular application requirements and limits, due to its higher processing speed and reduced memory requirements. In practical applications, the results may be affected by factors such as image content, transformation, and noise. Therefore, when choosing a suitable feature matching algorithm in practical applications, multiple factors need to be considered, such as algorithm performance, processing time, application scenarios, etc.

#### References

- [1] Zhaobin Wang, and Zekun Yang. Review on image-stitching techniques. 2020, *Multimedia Systems* **26**: 413-430.
- [2] Sánchez J, Monzón N, Salgado De La Nuez A. An analysis and implementation of the harris corner detector. 2018, *Image Processing on Line*.
- [3] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. 2014, *International Journal of Computer Vision* **60**, 91–110.
- [4] Yan Ke and R. Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, 2004. *IEEE Conference on Computer Vision and Pattern Recognition*. 137-149.
- [5] Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. 2006 *European Conference on Computer Vision*. **3951**, 404–417.
- [6] Herbert B., Andreas E., Tinne T. and Luc Van G.: Speeded up Robust Feature (SURF), 2008 *Computer Vision and Image Understanding*, **110 (3)**: 346- 359.
- [7] Utsav S., Darshana M. and Asim B.: Image Registration of Multi-View Satellite Images Using Best Feature Points Detection and Matching Methods from SURF, SIFT and PCA-SIFT 1(1): 2014 *European Conference on Computer Vision* 8-18.

- [8] E. Rublee, et al. ORB: An efficient alternative to SIFT or SURF. 2011 *International conference on computer vision*, 1-11.
- [9] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. 2010, *European Conference on Computer Vision*, 1-10.
- [10] S. A. Bakar, X. Jiang, X. Gui, and G. Li, Image Stitching for Chest Digital Radiography Using the SIFT and SURF Feature Extraction by RANSAC Algorithm Image Stitching for Chest Digital Radiography Using the SIFT and SURF Feature Extraction by RANSAC Algorithm, 2020, *European Conference on Computer Vision* 1-12.
- [11] Fischler, Martin A., and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. 2020 *Communications of the ACM* **24.6**, 381-395.