Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/16/20230900

Comparative study of feature extraction algorithms for panorama stitching

Tianxingjian Ding

College of Information and Electronic Engineering, Zhejiang University, Hangzhou, 310000, China

3200104836@zju.edu.cn

Abstract. Panorama stitching is a fascinating and rapidly advancing research field. By integrating many photographs that were taken from various angles and viewpoints, with various exposure and color settings, a seamless image is primarily the aim of panorama stitching. This paper investigates the performance of three widely used feature extraction algorithms Speeded-Up Robust Features (SURF), Scale-Invariant Feature Transform (SIFT), and Oriented FAST and Rotated BRIEF (ORB) for panorama stitching. The study compares these algorithms in terms of accuracy, robustness, and speed. Results indicate that while SURF and SIFT produce more accurate and robust results than ORB, they require longer processing time. The study evaluates the approach on a real-world dataset and demonstrates its effectiveness in creating seamless and visually appealing panoramas. This study provides valuable insights into the trade-offs between different feature extraction algorithms and presents a practical solution for panorama stitching applications.

Keywords: panorama stitching, feature extraction algorithms, SURF, SIFT, ORB.

1. Introduction

Panorama stitching is a popular technique for creating wide-angle images by combining multiple images. In recent years, with the increasing popularity of panoramic photography, the development of this technology has become even more critical. The primary goal of panorama stitching is to create a seamless image by combining multiple images that are taken from different angles and perspectives, with different exposure and color settings [1]. The ultimate goal is to create a realistic and immersive representation of a scene that can capture a viewer's attention.

The potential applications of panorama stitching are vast, ranging from scientific imaging to virtual reality. One of the most significant benefits of panorama stitching is the ability to capture vast landscapes, cityscapes, and architecture in a single shot. This technology is particularly valuable in the tourism industry, where panoramic images can provide immersive experiences to potential visitors, allowing them to explore and evaluate different destinations remotely. Panorama stitching is also an essential tool in the field of computer vision and robotics, where it can be used for 3D mapping, object recognition, and autonomous navigation. The significance of research in panorama stitching lies in its potential to revolutionize the way we see and interact with our environment. The technology has already made significant strides in recent years, with advances in image processing algorithms, computer vision techniques, and hardware development. The current research in panorama stitching is focused on improving the quality and speed of image stitching, as well as developing new applications in fields such as virtual reality, augmented reality, and autonomous vehicles.

In summary, panorama stitching is a rapidly growing research field with significant potential for various applications in photography, computer vision, and robotics. The development of panorama stitching technology has already had a significant impact on the tourism industry and is expected to revolutionize the way we interact with our environment in the future. The current research is focused on improving the quality and speed of image stitching, as well as developing new applications in emerging fields such as virtual and augmented reality.

In our research, we generate panoramic images using a three-step process, with each step being performed by a distinct algorithm. This paper aims to optimize the final result by carefully testing and evaluating the quality and speed of each algorithm at every step. This allows us to identify the best-performing algorithm at each stage and use it to enhance the final output.

2. Related work

Panorama stitching has been the subject of extensive research in the past decade, with numerous advancements in both theory and practice. The primary focus of early research was on developing algorithms that could stitch images together accurately and efficiently. One of the most significant contributions in this area was the SIFT (Scale-Invariant Feature Transform) algorithm proposed by David Lowe in 2004, which uses local features to match and align images [2]. Since then, many variations and improvements of the SIFT algorithm have been proposed, including SURF (Speeded-Up Robust Features), ORB (Oriented FAST and Rotated BRIEF), and AKAZE (Accelerated-KAZE) [1][3]. Another critical area of research in panorama stitching has been the development of image blending techniques to create seamless and natural-looking panoramas. Traditional image blending methods, such as linear and feather blending, have been widely used, but more advanced blending techniques, such as multi-band blending and seam-aware blending, have been proposed in recent years.

In addition to these technical advances, researchers have also explored new applications of panorama stitching technology. One promising area is virtual reality (VR), where panoramic images can be used to create immersive and realistic virtual environments. Researchers have proposed methods for generating high-resolution panoramas for VR, such as the MegaDepth algorithm proposed by researchers at Stanford University, which can create panoramic images from unstructured image collections. Another emerging area of research is the use of panorama stitching in autonomous vehicles. Panoramic cameras can be used to provide a 360-degree view of the vehicle's surroundings, which is useful for navigation and obstacle detection. Researchers have proposed methods for real-time stitching and image processing, such as the Fast Panorama Stitching algorithm proposed by researchers at the University of Hong Kong.

Overall, the research in panorama stitching has made significant contributions to the field of computer vision, robotics, and photography, and the development of new applications and techniques is expected to continue in the coming years.

3. Methods

3.1. Overall framework

Panorama stitching can be implemented using the following Figure 1.

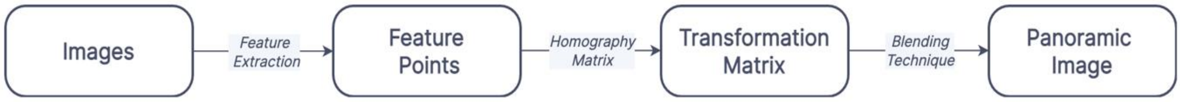


Figure 1. Implementation process

Firstly, the feature extraction algorithm is applied to multiple images to detect and extract distinctive features or points. To achieve robust and accurate results, it is important to test different feature

DOI: 10.54254/2755-2721/16/20230900

extraction algorithms such as SIFT, SURF, and ORB, and select the one that works best for the specific dataset.

Secondly, once the features are extracted, a feature matching algorithm is used to find the corresponding features across the images [2][4]. This step is critical to establishing the correspondence between different images and estimating the transformation parameters required for image alignment.

Thirdly, a homography estimation algorithm is used to estimate the transformation matrix that maps one image onto the other. This transformation matrix describes the relationship between the points in the two images and is used to warp the images to a common coordinate system [5-6].

Finally, the images are warped using the estimated homography matrix, and the overlapping regions are blended to create a seamless panoramic image. The blending technique used can significantly impact the final quality of the panorama, and different approaches such as linear blending, multi-band blending, and gradient domain blending can be tested.

3.2. Technical approach

3.2.1. Feature extraction.

- **SIFT**

SIFT features are highly robust and invariant to changes in image scale, orientation, and affine distortion. The SIFT algorithm is a comprehensive approach for feature extraction, covering all aspects from the detection of feature points to the generation of descriptors that accurately describe the image [4][7]. Gaussian filtering is used in the algorithm, as it is the only kernel function that is scale-invariant. Different levels of blurring can be achieved by varying the Gaussian kernel, mimicking the way objects appear at different distances on the retina. The key steps of the SIFT detector algorithm are outlined in Table 1.

Table 1. SIFT detector algorithm.

SIFT Detector Algorithm:
1. Generate a Difference of Gaussian Pyramid to construct the scale space.
2. Detect spatial extreme points to explore potential key points.
3. Precisely position stable key points.
4. Allocate stable key point orientation information.
5. Describe key points using their local image patches.
6. Match feature points by finding the closest descriptors in two images.

The Gaussian filter used in the SIFT algorithm is defined as:

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

where x and y are the pixel coordinates, and σ is the standard deviation of the Gaussian distribution. The Difference of Gaussian (DoG) function used to construct the scale space is defined as:

$$D(x,y,\sigma)=(G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y) \tag{2}$$

where k is the scale factor, and I (x, y) is the input image. The * denotes the convolution operation. The gradient orientation histogram H_i is defined as:

$$H_i=\sum_{p\in P}\omega_p\cdot d_p\cdot\delta(\theta_p-\theta_i) \tag{3}$$

where d_p is the gradient vector at pixel p, θ_p is the orientation of the gradient vector, $\delta(\theta_p-\theta_i)$ is a function that returns 1 if θ_p is within the I orientation bin, and ω_p is a weight that accounts for the distance of pixel p from the key point location.

- **ORB**

ORB features are scale-invariant and can detect features in different orientations, making them suitable for a wide range of applications [2][3]. The key steps of the ORB detector algorithm are outlined in Table 2.

Table 2. ORB detector algorithm: ORB detector algorithm.

ORB Detector Algorithm:
1. Generate a scale pyramid to construct the scale space.
2. Detect key points using the Features from Accelerated Segment Test algorithm.
3. Orient the key points based on their intensity centroid.
4. Generate binary descriptors using the BRIEF (Binary Robust Independent Elementary Features) algorithm [2].
5. Match feature points by finding the closest descriptors in two images

The scale pyramid is generated by resizing the input image at different scales:

$$I\_i\ (x, y)\ =\ I(x, y)\sigma\_i \tag{4}$$

where $I_i(x, y)$is the image at scale i, $I(x, y)$is the input image, and σ_i is the scaling factor. Key points are detected using the FAST algorithm, which tests the brightness of pixels along a circle of 16 pixels:

$$V(p)\ =\ max_{n\in 1..16}\left|I_n\ -\ I_p\right| \tag{5}$$

where V(p) is the brightness variation at pixel p, and I_n and I_p are the intensities of neighboring pixels. The orientation of key points is determined by computing the intensity centroid and computing the dominant orientation:

$$\theta = atan2(\sum_{p\in P} w(p)x(p)\ , \sum_{p\in P} w(p)y(p)) \tag{6}$$

where θ is the dominant orientation, p is the set of pixels in the key point's neighborhood, and x(p) and y(p) are the coordinates of pixel p relative to the key point. Feature points are matched by finding the closest descriptors in two images using a distance metric such as Hamming distance. The best matches are then identified using a ratio test.

- **SURF**

SURF is a feature extraction algorithm that is both fast and robust to image scale, rotation, and affine distortion. The SURF algorithm is based on the computation of local image features that are highly distinctive and invariant to changes in image appearance [6]. These features are extracted using a series of filters that are applied at different scales and orientations.

The key steps of the SURF algorithm are as follows in Table 3.

Table 3. SURF detector algorithm: SURF detector algorithm.

ORB Detector Algorithm
1. Construct the scale space:
2. Compute the Difference of Gaussian (DoG) pyramid:
3. Detect feature points:
4. Compute the SURF descriptor for each key point:
5. Match feature points by finding the closest descriptors in two images

The scale space is constructed by convolving the input image with a series of Gaussian kernels at different scales (Formula 1). The Difference of Gaussian (DoG) pyramid is computed by subtracting adjacent scales of the Gaussian pyramid (Reference formula 2). Feature points are detected by finding extrema in the DoG pyramid and performing sub-pixel localization:

$$D(x) =\ D(x) + \frac{1}{2}\big(D(x+1) -\ D(x-1)\big)\delta(x) \tag{7}$$

The SURF descriptor is computed by computing the Haar wavelet responses in a 4x4 neighborhood around the key point and forming a 64-dimensional feature vector:

$$d = [d_1, d_2, \dots, d_{64}] \tag{8}$$

where each element $d_i$is the signed sum of the wavelet responses in a particular subregion. Feature points are matched by finding the closest descriptors in two images using a distance metric such as Euclidean distance or Hamming distance [8]. The best matches are then identified using a ratio test.

The SURF algorithm is known for its speed and efficiency, making it suitable for real-time applications such as object recognition and tracking. The algorithm achieves its speed by using an

DOI: 10.54254/2755-2721/16/20230900

approximation of the Laplacian of Gaussian filter, which can be efficiently computed using box filters. The resulting features are highly distinctive and invariant to a wide range of image transformations, making them suitable for use in a variety of computer vision applications.

3.2.2. Homography estimation. RANSAC (RANdom Sample Consensus) is used to implement homography estimation. The RANSAC algorithm is an efficient and widely-used method for estimating model parameters in the presence of outliers or noise. It works by randomly selecting a subset of data points from a set of n points and fitting a model to these points. The process is repeated for a fixed number of iterations, with the largest consensus set of inliers being used to determine the final model. Specifically, if we assume that the majority of the data points can be explained by a model, and that at least M points are required to estimate the model parameters, then the RANSAC algorithm proceeds as Table 4.

Table 4. RANSAC algorithm.

RANSAC Algorithm:
1. Randomly select M data points from the input set
2. Detect spatial extreme points to explore potential key points.
3. Estimate the model parameters based on the selected data points
4. Repeat steps 1-3 for a fixed number of iterations, selecting the largest consensus set of inliers as the final model.

3.2.3. Blending technique. Blending techniques can be used in combination to create the best possible panorama. The specific blending technique used will depend on the characteristics of the images being stitched and the desired final result.

Smoothing is a commonly used technique in image processing and manipulation that allows for the creation of a smooth transition between two or more images. This is achieved by gradually blending the edges of each image into the next one, resulting in a seamless and visually pleasing transition.

Mathematically speaking, smoothing involves applying a smoothing mask to the edges of each image. The smoothing mask is a mathematical function that decreases in intensity from the edge of the image towards the center [9-10]. This can be represented by a Gaussian function, which has the property of smoothly decreasing in intensity from its peak value towards zero.

Let I1 and I2 be two adjacent images that we wish to blend using feathering. We first compute the smoothing mask F for each image, which is a 2D array of values between 0 and 1 that represents the intensity of the smoothing effect at each pixel. The mask F can be defined as follows:

$$F(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} \tag{9}$$

where (x,y) are the coordinates of a pixel in the image, (x_0, y_0) are the coordinates of the center of the smoothing effect, and σ is a parameter that controls the size of the smoothing effect.

Once we have computed the smoothing masks F1 and F2 for the two images, we can blend them together by computing a weighted sum of the two images using the following formula:

$$I_{blend} = (1 - F_1)I_1 + F_2 I_2 \tag{10}$$

where I_blend is the blended image, I1 and I2 are the original images, and F1 and F2 are the smoothing masks for each image.

4. Experiment and result

4.1. Platform introduction

The platform configuration parameters used in this report are the complier and version Pycharm 2021.3.1, using OpenCV in the python language as a framework, creating a virtual environment using Anaconda, and installing Python 3.8.0 in the virtual environment.

DOI: 10.54254/2755-2721/16/20230900

4.2. Result

The source pictures are shown in the Figure 2-3.

Figure 2. (a) Building image 001 (left) (b) 002(middle) (c) 003 (right).

Figure 3. (a) Chair 001(left). (b) Chair 002(middle) (c) Chair 003(right).

The above two sets of data sets show the two original images to be stitched. One set of data is for a large range of scenarios, and the other is for a small range of scenarios.

The preliminary splicing effect is shown in Figure 4-5

(a) SIFT (b) ORB (c) SURF

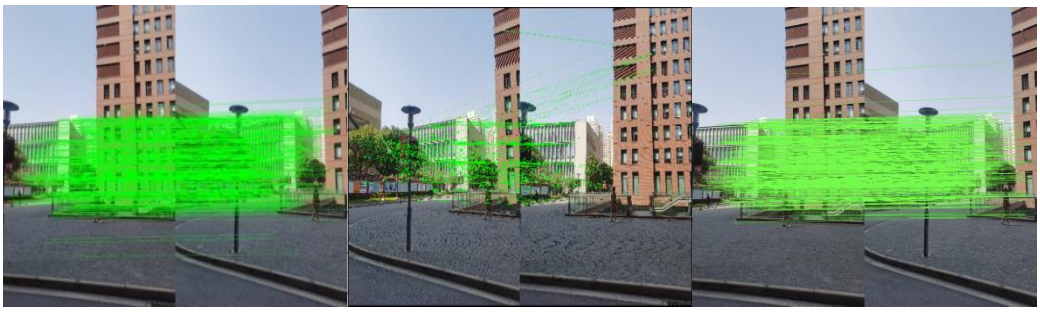
Figure 4. Chair result of three algorithms.

DOI: 10.54254/2755-2721/16/20230900

(a) SIFT (b) ORB (c) SURF

Figure 5. Building result of three algorithms.

Based on the analysis of the results obtained from applying feature extraction algorithms such as SURF, SIFT, and ORB to three images and then stitching them together, it can be observed that SURF and SIFT algorithms produce quite similar images with satisfactory stitching results. However, there seem to be some noticeable discrepancies in the stitched image obtained from the ORB algorithm. For instance, in the first image, there appears to be a slight bending at the leg of the stool, while in the second image, there seems to be a visible break at the step. The results of some matching points are shown in the figure 6.



(a)SIFT (b) ORB (c) SURF

Figure 6. Part of matching points of three algorithms.

Upon analyzing the matching point images, it can be observed that the SURF and SIFT algorithms yield more precise and resilient matching points in comparison to the ORB algorithm. This is due to the fact that SURF and SIFT implement more sophisticated and advanced techniques for feature detection and matching.

The precision of the matching points is a crucial factor in the success of image stitching since it directly affects the quality of the final image. In the research, it was discovered that the matching points produced by the SURF and SIFT algorithms align well with each other, resulting in a seamless and natural-looking stitched image. Conversely, the ORB algorithm's matching points had more errors and crossing, leading to visible distortions and inconsistencies in the final stitched image. It seems that the homography matrices acquired through the ORB algorithm are not as precise as those acquired through SURF and SIFT algorithms. Therefore, the matching points created by the ORB algorithm are not as reliable, which can cause issues in the combined images.

Furthermore, table 5 in this study examines the processing time of the key points. The outcomes indicate that the ORB algorithm's binary descriptor, which has a faster computation speed and requires less memory, could contribute to the reduced precision.

Table 5. Compares the runtime per keypoint.

	SURF	SIFT	ORB
Chair Image	0.00046 s	0.00079 s	0.00031 s
Building Image	0.00043 s	0.00076 s	0.00027 s

The contents above lead to the following conclusion:

- SIFT is slower compared to SURF, but the matching results of both operators are similar and both are good
- ORB is the fastest operator, but its matching is less robust and will show significant ghosting
- SURF should be used in the final matching algorithm.

5. Conclusion

This paper aimed to compare the performance of three different feature detection algorithms, namely ORB, SIFT, and SURF, for panorama stitching. To estimate a single stress matrix and determine the overlap area between images, we employed the RANSAC algorithm. We observed that SIFT and SURF produced more accurate feature matches and stitching results compared to ORB, especially for images with noise or occlusion. Nonetheless, they tend to be more computationally intensive, making them less optimal for processing large datasets or real-time applications. The results of the experiments highlight the importance of careful selection of feature detection algorithms to achieve high-quality panorama stitching. In particular, the use of SIFT or SURF can lead to improved accuracy in stitching, but at the cost of increased computational complexity.

In conclusion, this work demonstrates the potential of feature detection algorithms such as SIFT, SURF, and ORB in panorama stitching. Further research could focus on optimizing the performance of these algorithms or exploring new techniques to improve the efficiency and accuracy of panorama stitching.

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in 2006 European Conference on Computer Vision, Graz, Austria, 2006, pp. 404-417. DOI: 10.1007/11744023_32.

[2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in 2010 European Conference on Computer Vision, Heraklion, Greece, 2010, pp. 778-792. DOI: 10.1007/978-3-642-15567-3_56.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564-2571. DOI: 10.1109/ICCV.2011.6126544.

[4] K. Mikolajczyk and C. Schmid, "Scale and Affine Invariant Interest Point Detectors," International Journal of Computer Vision, vol. 60, no. 1, pp. 63-86, Nov. 2004. DOI: 10.1023/B:VISI.0000027790.02288.f2.

[5] T. Tuytelaars and L. Van Gool, "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions," in 2000 British Machine Vision Conference, Bristol, UK, 2000, pp. 412-422.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359, Jun. 2008. DOI: 10.1016/j.cviu.2007.09.014.

[7] L. Liu and W. Ouyang, "SIFT Meets CNN: A Decade Survey of Instance Retrieval," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 8, pp. 1545-1565, Aug. 2017. DOI: 10.1109/TPAMI.2016.2631055.

[8] P. J. Burt and R. J. Kolczynski, "Enhanced Image Capture Through Fusion," in 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 2003, vol. 2, pp. II-173-II-180. DOI: 10.1109/CVPR.2003.1211459.

[9] Z. Lin, W. Gao, J. Jia, et al., "CapsNet Meets SIFT: A Robust Framework for Distorted Target Categorization," Neurocomputing, vol. 464, pp. 290-316, Oct. 2021. DOI: 10.1016/j.neucom.2021.06.101.

[10] H.-J. Chien, C.-C. Chuang, C.-Y. Chen, et al., "When to Use What Feature? SIFT, SURF, ORB, or A-KAZE Features for Monocular Visual Odometry," in 2016 International Conference on Image and Vision Computing New Zealand, Palmerston North, New Zealand, 2016, pp. 1-9. DOI: 10.1109/IVCNZ.2016.7804432.