

Research on Adaptive RANSAC Panoramic Stitching Optimization for Virtual Reality Vision Generation

Mei Huang

Changsha Normal University, Hunan, Changsha, China

Abstract: Aiming at the high standard requirements for real-time generation of visual content in virtual reality (VR) scene construction, as well as the efficiency bottlenecks and robustness deficiencies of traditional image stitching methods in dealing with complex environments, this paper proposes an improved random sampling consistency (M-RANSAC) algorithm based on adaptive enhancement strategy. The algorithm aims to optimize the model parameter estimation in the panoramic stitching process. Specifically, this study first uses the HARRIS operator to extract image feature points, and introduces a priority ranking mechanism based on the distance of matching point pairs to realize the preprocessing of the feature point set; on this basis, the sampling scale of the minimum sample set is dynamically adjusted according to the real-time ratio of the internal and external points, and the invalid model is effectively filtered by combining the double threshold data inspection strategy. The experimental data show that the improved strategy significantly shortens the time consumption of model sampling and data verification through feature point sorting and pretest screening, and reduces the estimation deviation of homography matrix, which greatly improves the operation efficiency and stability of the algorithm under the premise of ensuring the splicing accuracy. The research results provide an efficient and reliable technical solution for the real-time construction of immersive virtual reality visual content, which has broad application value.

Keywords: Virtual Reality; RANSAC; Image Stitching; Feature Point Matching; Feature Ranking

1. Introduction

Image stitching technology has become one of

the core technologies of modern visual computing by seamlessly merging multiple images with overlapping regions to generate panoramic images with wide viewing angle and high resolution [1]. This technology breaks through the limitation of the physical field angle of a single optical sensor, and shows irreplaceable application value in the fields of medical image diagnosis, automatic driving environment perception, virtual reality (VR) immersive experience and industrial precision detection [2]. This technology has shown great application potential in many disciplines: in medical diagnosis, it breaks through the physical limitation of the limited field of view of endoscope or ultrasonic probe, and realizes the complete observation and functional evaluation of large-scale lesions [3]. In the field of autonomous driving and environmental perception, it enhances the vehicle's perception of complex traffic environments through multi-sensor collaboration [4]; in the field of virtual reality (VR) and panoramic media, high-quality panoramic scene construction directly determines the user's immersive experience [5]; in addition, in the fields of remote sensing mapping, surveillance video collaboration and medical image analysis, image stitching is also a key means to realize the reconstruction of scene information integrity [6,7].

Among the many image registration techniques, the stitching method based on feature points has become the most commonly used mainstream solution because of its strong scene adaptability [8]. The core of this kind of method is to realize image alignment through feature extraction, matching and estimation of transformation matrix. As the cornerstone of model parameter estimation, the Random Sample Consensus (RANSAC) algorithm has been widely used in the field of image stitching due to its robustness to outliers since it was proposed [9]. However, with the complexity of the application scenarios, the limitations of the traditional RANSAC

algorithm have become increasingly prominent : when the original data volume is large, the data error rate is high, or the complexity of the model estimation increases, the traditional algorithm often requires a large number of random sampling and full data verification, resulting in an exponential increase in the amount of calculation, which is difficult to meet the real-time requirements of virtual reality and dynamic vision generation scenarios [10,11].

Although deep learning technology has developed rapidly in feature extraction and panoramic generation in recent years, its balance between real-time performance and computing resource consumption has not been perfectly solved [1,12]. The traditional geometric registration framework based on feature points is still the mainstream foundation of current industry and academia because of its strong interpretability, low demand for computing resources and high robustness in texture-rich scenes [13]. In this framework, the Random Sample Consensus (RANSAC) algorithm is the core of eliminating mismatched points and estimating the global homography matrix. Its performance directly determines the geometric accuracy and visual coherence of the final panoramic image [3]. Therefore, in view of the low efficiency and poor stability of the traditional RANSAC algorithm in image stitching, it is still an important direction of current research to optimize and improve it.

Aiming at the challenges of high real-time requirements of current visual content generation and insufficient efficiency of traditional methods, this paper proposes an improved RANSAC algorithm based on adaptive enhancement strategy-M-RANSAC algorithm. This method focuses on improving the estimation algorithm of image transformation model. Firstly, the feature points are extracted by HARRIS algorithm. On the basis of sorting and prioritizing the matching feature points according to the distance, the minimum sample set sampling times are dynamically determined based on the proportion of internal points and external points, and the double threshold method is introduced for data pretest. Through this optimized sampling and testing mechanism, the M-RANSAC algorithm significantly reduces the estimation error probability of the homography matrix while ensuring the matching accuracy, and effectively reduces the sampling time and data testing time. The experimental results show

that the algorithm has excellent efficiency and stability in processing large-scale data, which provides an efficient and feasible technical basis for the real-time generation of immersive visual scenes such as virtual reality.

2. RANSAC Algorithm

In feature-based stitching techniques, calculating the transformation matrix is essential when seeking correspondences between feature points. Although the matched points after feature extraction, matching, and coarse filtering far exceed the number needed for model estimation, there are still a significant number of mismatches. The traditional RANSAC (Random Sample Consensus) algorithm [14] estimates the transformation matrix through random sampling consistency. However, when the original data is large, the error rate is high, and the model complexity increases, the computational load can become significant, leading to a drastic reduction in efficiency. To address this issue, this paper proposes an improved RANSAC algorithm, M-RANSAC, which effectively resolves the problem of decreased efficiency.

2.1 Content of the RANSAC Algorithm

The RANSAC algorithm is a robust parameter estimation method, with its core focused on iteratively verifying and searching for the optimal model parameters. The algorithm can be summarized in the following steps: establish a mathematical model based on the specific problem and define the objective function; randomly select M groups of minimal sample sets from the data set, with the size of each sample group determined by the objective function. For each sample group, calculate the initial parameters, and classify the data points into inliers (points that satisfy the initial parameter values for that group) and outliers (points that do not satisfy the initial parameter values for that group) based on a predefined error threshold. Estimate the model parameters for each sample group and count the inliers. By comparing the number and distribution quality of inliers corresponding to each parameter set, the optimal model parameters are selected as the final solution. Below are the steps for estimating the transformation matrix using the RANSAC algorithm:

1) Based on the projection transformation relationship between the two images, a transformation function is determined. In this

paper, a projective transformation [15-17] is adopted, which aligns with the transformation relationship of planar objects captured from different viewpoints and angles. The corresponding feature points of the two images satisfy the following transformation relationship:

$$\begin{pmatrix} x_l^{i'} \\ y_l^{i'} \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{pmatrix} \begin{pmatrix} x_r^i \\ y_r^i \\ 1 \end{pmatrix} \quad (1)$$

$(I_l(x_l^i, y_l^i), I_r(x_r^i, y_r^i))$ is the i -th pair of matched feature points, (x_r^i, y_r^i) is the feature point from the right image. After applying the transformation matrix, the corresponding transformed feature point $(x_l^{i'}, y_l^{i'})$ is obtained. If the parameter model is correct, then there should be $x_l^{i'} = x_l^i$ and $y_l^{i'} = y_l^i$;

2) Randomly and repeatedly extract a set of data from the feature point pair set S , with each sampled set containing the minimum number of data required to estimate the model. The model corresponding to this sampled set is calculated, and the model parameters used in this paper require 4 pairs of matched points;

3) Use all the data in the set S to validate the model, which is called full data validation. Count the number of inliers that satisfy the model, as well as the sum of the Euclidean distances between the transformed inliers and the matched inliers. The formula for the sum of distances is as follows:

$$sum = \sum_{i \leq num}^{i=1} \sqrt{(x_l^{i'} - x_l^i)^2 + (y_l^{i'} - y_l^i)^2} \quad (2)$$

num is the number of inliers that satisfy the

Table 1. Table of Changes in M at P = 0.95

Number of Matching Pairs in the Sampling Group m	The data error rate ε						
	5%	10%	20%	25%	30%	40%	50%
4	2	3	6	8	11	22	47
6	3	4	10	16	24	63	191
8	3	6	17	29	51	177	766
10	4	7	27	52	105	494	3067

From the above table 1, it can be observed that when the original data volume is large, the model is complex, and the data error rate is high, the efficiency of the RANSAC algorithm directly decreases. Assuming there are SN pairs of matching points in the feature point set S , the time required to randomly sample a group is t_1 ,

model;

4) Select model parameters based on the number of inliers and the sum of Euclidean distances;

5) Repeat steps 2 and 3 until a model that meets the selection criteria is identified. Find the inliers corresponding to this model, and use these inliers to calculate the final model parameters.

2.2 Computational Complexity of the RANSAC Algorithm

In the RANSAC algorithm, to ensure that there is at least one set of all inliers in the random sampling with a certain confidence probability P , it is required that in one sample, all 4 pairs of matching points are inliers. Suppose M groups of samples are taken, and the probability that all M groups of matching points are outliers is $(1-P)$. Let the data error rate be ε (the proportion of outliers in the original data), then $(1-\varepsilon)$ is the probability of selecting any pair of matching points as inliers. The probability that m pairs of matching points are all inliers is $(1-\varepsilon)^m$. Therefore, the probability that after M samples, all sampled groups are outliers is $(1-(1-\varepsilon)^m)^M$. From this, formula (3) can be derived.

$$(1-(1-\varepsilon)^m)^M = 1-P \quad (3)$$

From formula (3) [18], it can be seen that the number of matched pairs m obtained from a single sample, the number of samples M , the confidence probability P , and the data error rate ε have an exponential relationship. Table 1 shows the variations of M with respect to ε and m when $P = 0.95$.

while calculating the model parameter values

from a sample set takes t_2 . The average time required to test one data point against a model is t_3 . Thus, the total time to complete the RANSAC algorithm consists of two parts: the

random sampling time $M \times (t_1 + t_2)$ and the data verification time $M \times SN \times t_3$. Therefore, to improve the algorithm's efficiency, efforts can be made to reduce both the sampling time and the data verification time.

3. M-RANSAC Algorithm

3.1 Sampling Count

In the RANSAC algorithm, M groups of samples are randomly drawn repeatedly from the original dataset until a model that meets the selection criteria is selected. If we sort the original dataset and start sampling from the highest quality data, we can find a model that satisfies the selection criteria more quickly, thereby significantly reducing the number of samples needed.

Sorting the data in the original dataset involves returning to the feature point matching process. During matching, some incorrect matches between feature point pairs are inevitable, leading to a certain error rate in the original dataset. This, in turn, results in inaccurate model estimation or an increased number of samples needed. In the feature point matching process, arranging the matched pairs in a specific order—prioritizing the most reliable matches—allows for a more efficient sampling strategy. By sorting the original dataset and starting the sampling process with high-quality data, we can reduce the number of required samples.

Assuming the set of feature point pairs is denoted as S , $(I_l(x_l^i, y_l^i), I_r(x_r^i, y_r^i))$ and $(I_l(x_l^j, y_l^j), I_r(x_r^j, y_r^j))$ represent the correct matching pairs for the i -th and j -th pair, respectively, the corresponding relationship between them can be expressed as follows:

$$|x_l^i - x_r^i| = |x_l^j - x_r^j| \quad (4)$$

$$|y_l^i - y_r^i| = |y_l^j - y_r^j| \quad (5)$$

The expression (4) represents the distance metric between the row indices of the i -th feature point pair, which is equal to the distance metric between the row indices of the j -th feature point pair. The expression (5) represents the distance metric between the column indices of the i -th feature point pair, which is equal to the distance metric between the column indices of the j -th feature point pair.

This means that for any two correctly matched feature point pairs, the corresponding coordinates should satisfy expressions (4) and (5). We can assume that after matching feature points and removing spurious feature points, the correct feature point pairs constitute the majority in the set. Therefore, we will sort the feature points based on the distance metrics of the feature point pairs. The steps for sorting and partitioning the feature point pair set are as follows:

1) Calculate the distance metrics for all feature point pairs. The distance metrics for the row indices are stored in the *distance_x* matrix, while the distance metrics for the column indices are stored in the *distance_y* matrix.

2) Count the different types of distance metrics for all feature point pairs and tally the number of occurrences of each type. Record the feature point pairs that belong to each type. For example,

if $(distance_x_i, distance_y_i)$ represents the distance metric for the i -th feature point pair, and this distance metric does not match any existing types of distance metrics that are equal to both $distance_x_i$ and $distance_y_i$,

then $(distance_x_i, distance_y_i)$ is considered a new type of distance metric. If there exists a type of distance metric among the previously encountered types that is equal to both $distance_x_i$ and $distance_y_i$, increment the count for that type by 1 and record that the i -th feature point pair belongs to that type of distance metric.

3) Sort the distance metric types based on their counts. Assuming we have identified n types of distance metrics, the feature point pair set S can be divided into n subsets $S_1, S_2, S_3, \dots, S_n$. The

first subset S_1 corresponds to the feature point pairs of the most frequent distance metric type,

the second subset S_2 corresponds to the second most frequent type, and so on. After sorting, we

will re-divide these n subsets $S_1, S_2, S_3, \dots, S_n$ into m new subsets

$\{S_1\}, \{S_1, S_2\}, \{S_1, S_2, S_3\}, \dots, \{S_1, S_2, S_3, \dots, S_n\}$,

assigning m increasing data error rates to these

new subsets, where the data error rate of $\{S_1\}$ is the lowest, and so forth. According to formula

(3), the maximum number of sampling attempts can be calculated for each different error rate. For instance, if the number of sampling attempts for $\{S_1\}$ is $M1$ and the conditions for selecting model parameters are not met within those $M1$ attempts, the sampling range will be expanded to include the subset $\{S_1, S_2\}$, with the number of sampling attempts now set to $M2$. This process continues sequentially until satisfactory model parameters are found.

3.2 Data Verification Time

In the RANSAC algorithm, the transformation matrix model computed in each iteration undergoes a full data verification, meaning that all points in the dataset are involved in model verification. Reducing the number of models that participate in full data verification can help decrease verification time. Based on the partitioning of feature points into subsets S from the previous section, we can check the quality of the data in these subsets when generating new model parameters. If the inlier ratio of this model exceeds a predefined inlier ratio $B1$ (the ratio of inlier points to the total number of data points in the sampled subsets) and the Euclidean distance threshold $T1$, then this model can proceed to full data verification; otherwise, the model is discarded and the next sampling occurs. This approach can significantly reduce model verification time when dealing with large volumes of original data.

3.3 Implementation of M-RANSAC Algorithm

In this study, the HARRIS algorithm [19] is used to extract feature points at different scales [20-22] and a bidirectional normalization method [23,24] is employed to match these feature points. Additionally, a circular threshold and a distance comparison method for feature points [22] are applied to eliminate false matches, resulting in a set of feature point pairs S . Based on these foundations, we implement the M-RANSAC algorithm.

The core steps of M_RANSAC algorithm:

- 1) Sort and partition the set SS based on the number of types of distance pairs of feature points, resulting in n subsets $S_1, S_2, S_3, \dots, S_n$.
- 2) Partition the n subsets $S_1, S_2, S_3, \dots, S_n$ into m new subsets

$\{S_1\}, \{S_1, S_2\}, \{S_1, S_2, S_3\}, \dots, \{S_1, S_2, S_3, \dots, S_n\}$. Set m incremental data error rates and calculate the maximum number of samples $M_{\max_i}$ for each subset at different error rates using formula (3).
 3) Randomly sample 4 pairs of matched points from the i -th subset (with i starting from 1, beginning with $\{S_1\}$). Using this sampled data, calculate the model parameters. Let the number of samples from the i -th subset be denoted as M_i . Each time the subset is iterated, increment M_i by 1. If M_i exceeds the maximum number of samples $M_{\max_i}$ for the i -th subset, increment i by 1 and begin sampling from the next subset. If M_i is less than or equal to $M_{\max_i}$, proceed to step (4).

4) Calculate the inlier ratio and the sum of the Euclidean distances of inliers for the model parameters in the i -th subset. Set the threshold for the inlier ratio as $B1$ and the threshold for the sum of the Euclidean distances as $T1$. If the inlier ratio is greater than $B1$ and the sum of the Euclidean distances is less than $T1$, proceed to step (5). Otherwise, return to step (3) and restart the sampling process.

5) Perform full data verification of the model parameters, setting the threshold for the inlier ratio as $B2$ and the threshold for the sum of Euclidean distances as $T2$. If the model's inlier ratio is greater than $B2$ and the sum of Euclidean distances is less than $T2$, proceed to step (6). If these conditions are not met, return to step (3) and restart the sampling process.

6) Using the selected model parameters, calculate all inliers of the model to determine the final model.

The M-RANSAC algorithm, based on the sorting and partitioning of feature points, calculates the minimum sampling times at different error rates, thereby reducing sampling time. Additionally, by employing a pre-detection method, it reduces the number of models participating in full data verification, resulting in significant savings in data verification time.

4. Experimental Results

To validate the effectiveness of the improved algorithm, we conducted four sets of experiments using eight images of different sizes. The image resolutions are shown in Table 2, and

the images are presented in Figure 1.

Experimental Conditions:

- Processor: Intel(R) Core(TM) i7-13700H, 2.40 GHz
- Memory: 32 GB
- Operating System: Windows 11
- Software: MATLAB 2023

Data Source: The experimental data is sourced

from an open-source image dataset.

Table 2. Image Resolutions

Group	Image	Resolution
First Group	a1,a2	400×300
Second Group	b1,b2	600×400
Third Group	c1,c2	1000×665
Fourth Group	d1,d2	1500×1125



a1. Landscape Left Image



a2. Landscape Right Image



b1. Forest Left Image



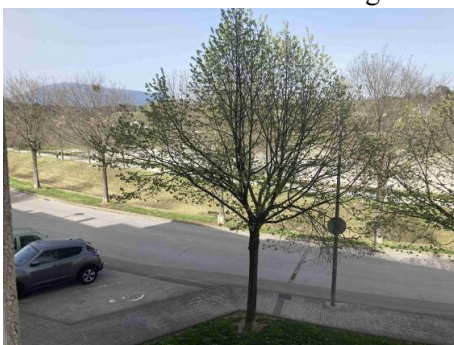
b2. Forest Right Image



c1. Street Scene Left Image



c2. Street Scene Right Image



d1. Trees Left Image



d2. Trees Right Image

Figure 1. Four Groups of Images to be Stitched

This paper compares the traditional RANSAC algorithm, the inlier loop RANSAC algorithm

(which calculates the sampling times based on the inlier ratio), and the improved M-RANSAC

algorithm.

In the traditional RANSAC algorithm, a minimum inlier ratio $min_inliers$ is set, and sampling times are calculated at a confidence level P of 0.95. After each model estimation, the entire dataset is tested to compute the errors of all points and identify the inliers. If the number of inliers in the current model exceeds the maximum inlier count and meets the minimum inlier requirement, the best model and inliers are updated until the model with the most inliers is

selected.

The inlier loop RANSAC algorithm primarily calculates the inlier ratio based on the content quantity of the current model. It dynamically adjusts the sampling times according to the inlier ratio using Equation (3), thereby optimizing algorithm efficiency.

Each of the four groups of images underwent nine experiments, and the average running times of the three algorithms for each group of images are shown in Table 3.

Table 3. Analysis of Algorithm Results

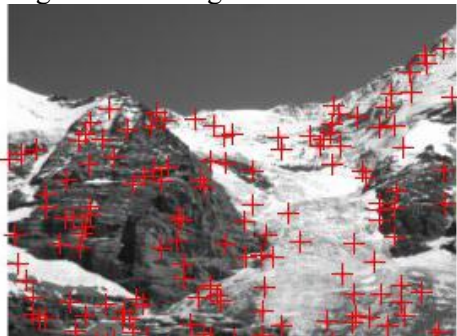
Group	Number of feature points (points)	Average Runtime (Seconds)		
		The traditional RANSAC algorithm	The inlier loop RANSAC algorithm	M-RANSAC algorithm
First Group	143	0.2686	0.0759	0.0228
Second Group	334	0.3656	0.1602	0.0474
Third Group	538	25.8780	2.1134	0.2257
Fourth Group	2204	49.2869	6.7193	1.1284

The first set of models obtained from the experiment is:

$$model_1 = \begin{bmatrix} 0.979549 & -0.037889 & -14.061338 \\ 0.053003 & 0.962958 & 143.995001 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix} \quad (6)$$

The images in the figure 2 have a size of

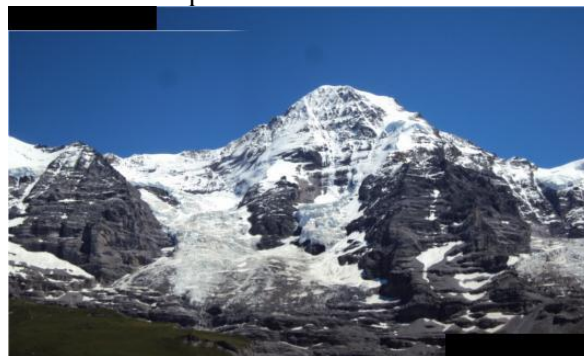
400×300 and an average of 143 feature points extracted. The model exists with translation and rotation relationships. The efficiency of the three algorithms shows little difference under conditions of few feature points and a simple model.



a1. Feature Point Distribution Map



a2. Model Transformation Effect Map



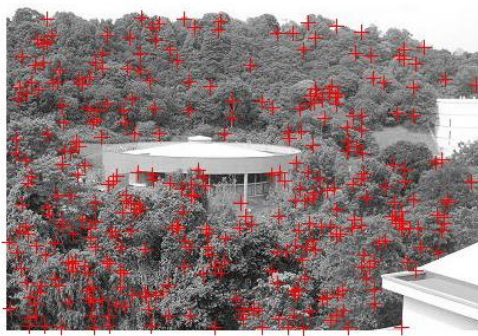
a3. Resulting Image of Landscape Painting Stitching

Figure 2. The First Group of Experiments

The second set of models obtained from the experiment is:

$$model_2 = \begin{bmatrix} 1.005601 & 0.028703 & -47.066278 \\ 0.004287 & 0.976469 & 202.563884 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix} \quad (7)$$

The images in the Figure 3 have a size of 600×400, with an average of 334 feature points extracted. The model includes translation and rotation relationships. As the number of feature points increases, the efficiency of the three algorithms shows significant differences.



b1. Feature Point Distribution Map



b2. Model Transformation Effect Map



b3. Resulting Image of Forest Stitching

Figure 3. The Second Group of Experiments

The third set of models obtained from the experiment is:

$$\text{model}_3 = \begin{bmatrix} 0.997091 & 0.003173 & -34.082644 \\ -0.040201 & 1.019027 & 316.835083 \\ -0.000009 & 0.000010 & 1.000000 \end{bmatrix} \quad (8)$$

The images in the Figure 4 have a size of 1000×665 , with an average of 538 feature points extracted. The model includes translation, rotation, and scaling relationships. As the image resolution and the number of feature points increase, the running times are as follows: the traditional algorithm takes 25.878 seconds, the Inlier Loop Algorithm takes 2.1134 seconds, and the M-RANSAC algorithm takes 0.2257 seconds. The running time of the traditional Ransac increases significantly, while M-RANSAC shows a notable improvement in efficiency.

The fourth set of models obtained from the experiment is:

$$\text{model}_4 = \begin{bmatrix} 1.019613 & 0.035169 & -2.131417 \\ -0.047937 & 1.023839 & 418.283406 \\ -0.000003 & 0.000002 & 1.000000 \end{bmatrix} \quad (9)$$

The images in the Figure 5 have a size of 1500×1125 , with an average of 2204 feature points extracted. The model includes translation, rotation, and scaling relationships. In high-resolution image stitching, the running times are as follows: the traditional algorithm takes 49.2869 seconds, the Inlier Loop

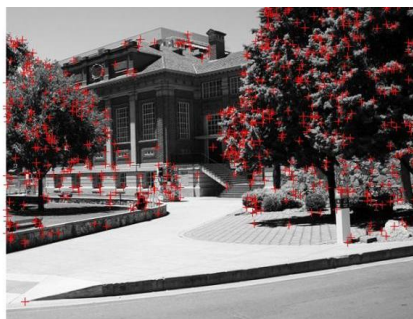
Algorithm takes 6.7193 seconds, and the M-RANSAC algorithm takes 1.1284 seconds. The running time of M-RANSAC is only about 1/44 of the traditional method and about 1/6 of the Inlier Loop Algorithm, demonstrating its strong adaptability to large-scale data.

From the Figure 6 stitching of the four sets with different resolutions and model complexities, the improved M-RANSAC algorithm consistently demonstrates optimal running efficiency. As the image resolution and the number of feature points increase, the time complexity of the traditional Ransac algorithm rises sharply. The Inlier Loop Ransac algorithm improves efficiency significantly by dynamically adjusting the number of samples compared to traditional methods, but further optimization is still needed to handle extremely large-scale data. In the fourth group of data, the M-RANSAC algorithm only requires 2.3% of the time of the traditional algorithm and 16.7% of the time of the Inlier Loop algorithm, maintaining high efficiency.

From the perspective of sampling, assume there are SN data points in the original dataset. The RANSAC algorithm randomly selects four pairs of points from these SN unordered data points, while the M-RANSAC algorithm selects four pairs of points from the top SM points among the SN ordered data, which are of higher quality. The higher quality set has a lower error rate.

According to Equation (3), as the error rate of the data decreases, the number of samples required from the set also decreases, leading to a reduced number of samples, meaning $M' < M$. From a probabilistic standpoint,

with $SM \ll SN$, we derive $\frac{1}{C_{SM}^4} > \frac{1}{C_{SN}^4}$. After sorting, we have a greater probability of approaching the correct model parameters, which reduces the sampling time.



c1. Feature Point Distribution Map



c2. Model Transformation Effect Map

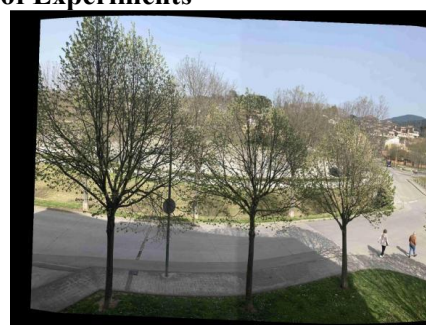


c3. Resulting Image of Street Scene Stitching

Figure 4. The Third Group of Experiments



d1. Feature Point Distribution Map



d2. Model Transformation Effect Map



d3. Resulting Image of Tree Stitching

Figure 5. The Fourth Group of Experiments

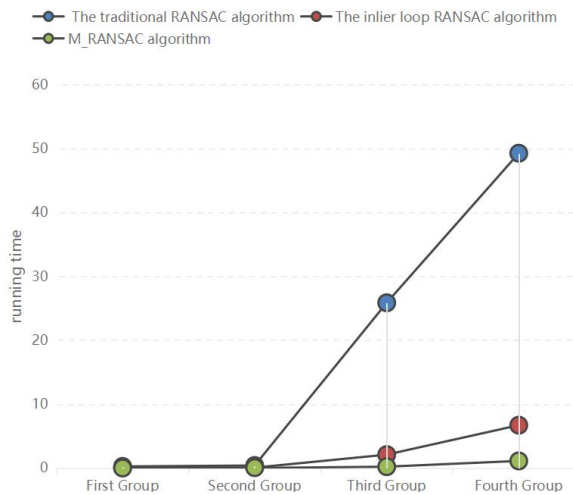


Figure 6. Algorithm Efficiency Diagram

From the perspective of data validation, in the nine stitching experiments of the fourth group, the M-RANSAC algorithm participated in an average of 1712 models for in-set validation, while the average number of models participating in the full data validation was 13. In contrast, the traditional RANSAC algorithm participated in an average of 479,318 models for full data validation, and the Inlier Loop algorithm participated in an average of 84,433 models. From this data analysis, it can be seen that the M-RANSAC algorithm preferentially extracts matching points from the higher quality

$$(M - M') \times (t_1 + t_2) + (M - M'') \times SN \times t_3 - M' \times SM \times t_3 \quad (12)$$

From the above analysis, it can be concluded that $SM \ll SN$, $M' < M$ and $M'' \ll M'$. The time for randomly selecting a sample is t_1 and the time for calculating the model parameters from a set of samples is t_2 , which does not vary with the complexity of the situation. Therefore, Equation (12) can be simplified as follows:

$$M \times SN \times t_3 \quad (13)$$

From the above expression, it is evident that when the situation becomes more complex—i.e., when the original dataset is large (SN increases), the error rate of the data is high (M increases), and the model is complex (the average time required to validate a model with a single data point t_3 increases)—the time difference between the algorithms will also increase, leading to a significant reduction in the efficiency of the RANSAC algorithm.

5. Conclusion

The M-RANSAC algorithm is an optimized

set and employs a pre-validation method within the set, filtering out most of the erroneous models. This significantly reduces the number of models participating in the full data validation, thereby decreasing the data validation time.

Now, from the perspective of time differences, let's assume that randomly selecting a sample takes t_1 , calculating the model parameter values

from a set of samples takes t_2 , and the average time required to validate a model with a single

data point is t_3 . Then, the computation time required for the traditional RANSAC algorithm is:

$$M \times (t_1 + t_2) + M \times SN \times t_3 \quad (10)$$

The computation time required for the M-RANSAC algorithm is:

$$M' \times (t_1 + t_2) + M' \times SM \times t_3 + M'' \times SN \times t_3 \quad (11)$$

Where $M' \times SM \times t_3$ is the time required for set data validation, and $M'' \times SN \times t_3$ is the time for the models passing the set data validation to participate in the full data validation. Using Equations (10) and (11), we can obtain the time difference between the two algorithms:

version of the Random Sample Consensus (RANSAC) algorithm that sorts and categorizes distance pairs of image feature points within the original dataset. By estimating model parameters from the optimized dataset through sampling and validation, it reduces the number of samples needed and the time required for data validation, thereby improving algorithm efficiency. In this experiment, four sets of images were used to compare and validate the efficiency of the improved algorithm. The experimental results showed that the M-RANSAC algorithm performs similarly to the traditional algorithm when dealing with simple models and a small amount of original data. However, when handling complex models and large datasets, M-RANSAC clearly outperforms the traditional RANSAC algorithm, significantly enhancing the efficiency of image stitching.

The significant advantages of the M-RANSAC algorithm indicate its broad application prospects in the fields of computer vision and image processing. As big data and artificial intelligence technologies continue to develop,

image stitching technology will find applications in more areas, such as medical image analysis, remote sensing image processing, virtual reality, and augmented reality. The efficiency of the M-RANSAC algorithm will make it an important tool in these fields, helping researchers and enterprises process and analyze image data more rapidly.

Acknowledgments

This work is supported by the Hunan Scientific Research Excellent Youth Project (Grant Nos. 24B1189).

References

- [1] Fu M, Liang H, Zhu C, et al. Image Stitching Techniques Applied to Plane or 3-D Models: A Review. *IEEE Sensors Journal*, 2023, 23(8): 8060-8079.
- [2] Soltanpour S, Joslin C. A Survey on Feature-Based and Deep Image Stitching//*Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. 2025: 777-788.
- [3] Yan Y, Xu L, Liu Y, et al. Larynx Ultrasound Image Stitching Based on Multiconstraint Super-Pixel Feature. *IEEE Transactions on Instrumentation and Measurement*, 2023, 72: 1-11.
- [4] Lang W, Wen Y, Bao Li. Multi-scenes image stitching based on autonomous driving. in *Proc. IEEE 4th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, 2020: 694–698.
- [5] Zhu S, Zhang Y, Zhang J, et al. ISGTA: an effective approach for multi-image stitching based on gradual transformation matrix. *Signal, Image and Video Processing*, 2023, 17(7): 3811-3820.
- [6] Wu H, Bao C, Hao Q, et al. Improved Unsupervised Stitching Algorithm for Multiple Environments Super UDIS. *Sensors*, 2024, 24(16): 5352.
- [7] Li Y, Wei B, Zhong Y, et al. A bibliometric analysis of global research on spinal cord injury: 1999–2019. *Spinal Cord*, 2021, 60(4): 281-287.
- [8] Jiayi M, Xingyu J, Aoxiang F, et al. Image matching from handcrafted to deep features: a survey. *International Journal of Computer Vision*, 2021, 129(1): 23-79.
- [9] Atokari M, Viitanen M, Mercat A, et al. Parallax-Tolerant 360 Live Video Stitcher//2019 IEEE Visual Communications and Image Processing (VCIP). IEEE, 2019: 1-4.
- [10] Khamiyev I, Issa D, Akhtar Z, et al. Panoramic image generation using deep neural networks. *Soft Computing*, 2023, 27(13): 8679-8695.
- [11] Jia Y, Wang R, Jiang X. A Real-Time Image Stitching and Fusion Algorithm Circuit Design Based on FPGA. *Electronics*, 2024, 13(2): 271.
- [12] Lee H, Lee S, Choi O. Improved method on image stitching based on optical flow algorithm. *International Journal of Engineering Business Management*, 2020, 12: 184797902098092.
- [13] Xiao J. Research of different feature detection and matching algorithms on panoramic image. *Applied and Computational Engineering*, 2023, 15(1): 38-51.
- [14] Raguram R, Chum O, Pollefeys M, et al. USAC: A Universal Framework for Random Sample Consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(8): 2022-2038.
- [15] Zhang J, Liu S, Sun L, et al. Plane-based content-preserving warps for image stitching. *IEEE Transactions on Visualization and Computer Graphics*, 2016, 22(7): 1848-1859.
- [16] Hongxing C, Jiahang J. Image Stitching Based on Corner Point Transformation Matrix. *Science Technology and Engineering*. 2008, 8(16):4536-4529.
- [17] Yao L, Lizhuang M. A Fast and Robust Image Stitching Algorithm. *Proceeding of the 6th World Congress on Intelligent Control and Automation Dalian, China*. October, 23rd, 2006.
- [18] Lifang W, Linlin H, Lin P, Lun Y. The Retinal Image Mosaic Based on Invariant Feature and Hierarchical Transformation Models The 2nd International Congress on Image and Signal Processing. Tianjin, China, October 30th, 2009.
- [19] Cheng B, Tang X. Research on Harris scale-invariant keypoint detector. *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*, 2010, 11(10): 755-762.
- [20] Bangsheng C, Xiaowei T. Study of Harris scale invariant keypoint detector. *Journal of Zhejiang University (Engineering Science)*,

- 2009, 43(5):855-859.
- [21]David G. Lowe. Distinctive Image Features Form Scale-Invariant Keypoints. International Journal of Computer Vision, 2004, 60(2):91-110.
- [22]Krystian Mikolajczyk, Cordelia Schmid. Scale & Affine Invariant Interest Point Detectors. International Journal of Computer Vision, 2004, 60(1):63-86
- [23]Barbara Zitova, Jan Flusser. Image registration methods: a survey. Image and Vision Computing, 2003, 21(11):977-1000.
- [24]Jiangning W, Baolong G, Zongzhong F. An Image Mosaic Method Based on Interest Point Matching. Journal of Computer Applications, 2006, 26(3): 610-612.