

Performance Comparison of RANSAC and Other Model Estimation Methods in Panoramic Image Mosaic

Yawen Ren

School of Computer Information and Technology, Beijing Jiao Tong University,
Weihai, China

22722067@bjtu.edu.cn

Abstract. As computer vision technology advances, the significance of outlier processing algorithms increases across various applications. Despite this, there is a lack of comprehensive comparisons to assess the performance of these algorithms. This paper focuses on evaluating the principles, advantages, and disadvantages of three key models: RANSAC, least squares, and lmed. Through experimental testing in diverse scenarios, the study identifies the most suitable model for different contexts. Results indicate that RANSAC exhibits the highest robustness against outliers, while lmed performs effectively with a moderate number of outliers. These findings are crucial for selecting appropriate models tailored to specific applications, ultimately enhancing the quality of panoramic mosaic images. The comparative analysis presented in this paper aims to guide practitioners in choosing the best outlier processing techniques based on their specific requirements and application environments, thereby improving the robustness and accuracy of computer vision systems.

Keywords: Outlier processing, RANSAC, LMED models, panoramic mosaic images.

1. Introduction

Panoramic image stitching is an important research direction in the field of computer vision, he can stitch multiple images into a panoramic image, which is particularly important for commercial and artistic value, such as virtual reality, augmented reality and satellite image processing and many other applications. With the advent of digital cameras, the desire to create full panoramic images has become stronger, and the importance of automatic image stitching has come to the fore [1]. However, in the process of image stitching, the inaccuracy of feature point matching and the existence of external points often lead to the failure of the stitching result. Therefore, it is an important purpose of this study to find the most suitable method to effectively exclude the outer points.

In today's increasingly tight computing resources, maintaining high efficiency and high precision is a challenge for image Mosaic technology. Therefore, it is of great research significance and value to find out the algorithm with the best performance, which not only provides theoretical value for subsequent academic research, but also optimizes today's stitching technology. Common methods for model estimation include random sampling consistency algorithm (RANSAC), least square method and least median square method (LMedS). RANSAC is very robust and removes outliers by randomly selecting sample points and iteratively optimizing the model. In contrast, the least squares method estimates model parameters by minimizing the sum of squares of residuals, but is highly sensitive to

external points [2]. LMedS attempts to reduce the influence of the maximum outlier on the model estimation through the median statistic and enhance the robustness of the model.

Although the advantages and disadvantages of these model estimation methods have been explored, there is still a research gap in the performance discussion of these algorithms in practical applications. Therefore, this paper will compare the performance of different methods such as RANSAC, least square and LMedS in panoramic image Mosaic, and explore their efficiency in handling outliers and the quality of Mosaic results. Finally, the most suitable model estimation method is given by comparing the data of different algorithms.

2. Related work

In order to improve the accuracy and efficiency of image Mosaic, many scholars have proposed various feature matching and model estimation methods. At present, the main methods are feature point matching, model estimation and deep learning.

2.1. Feature point matching

SIFT (Scale invariant Feature Transform) and SURF (Accelerated Robust feature) are two algorithms that detect local feature points to achieve feature matching. Lowe's study showed that SIFT is invariant to rotation and scale changes, which significantly improves the stitching quality, but it is not sufficiently robust and unique in feature matching [3]. In addition, SURF algorithm proposed by Bay et al. provides a faster feature computation speed than SIFT, which relies on image Gaussian scale space analysis [4]. ORB is a more computationally efficient alternative that is widely used in real-time image stitching. It combines FAST keypoint detection with BRIEF descriptors to make the feature matching process more efficient. However, in complex scenarios, the matching accuracy may decrease.

2.2. Model estimation

Since its introduction, RANSAC has become a standard method for outlier detection and model estimation. Torr and Zisserman demonstrated that RANSAC can also effectively estimate the homologous matrix when dealing with scenes with high ratio exceptions, thereby improving the stitching effect [5]. In order to improve robustness to outliers, In addition, the least square method, as a classical estimation method, has been widely used because of its simple mathematics and high computational efficiency. However, when there are a large number of external points, the performance of least squares degrades, resulting in unreliable model estimates. To solve this problem, the researchers introduced more powerful methods, such as LMedS (minimum median squared), designed to improve the model's resistance to outliers. In order to improve the robustness of the outlier, LMedS selects the median as the estimator to reduce the influence of the maximum outlier on the model.

2.3. Deep learning method

With the rapid development of deep learning, researchers begin to explore its application in panoramic image Mosaic. For example, convolutional neural networks (CNNS) have been used to learn complex relationships between depth pixels and color pixels, improving the accuracy of standard benchmark datasets [6].

3. Method and technology

3.1. Image capture

In this study, the author used his own mobile phone camera to shoot. The phone has a 50-megapixel camera. The phone's optical anti-shake feature reduces blurring during filming. In terms of shooting angles, the overlap between the images is between 30% and 50%. Thus, the stitching precision is improved.

3.2. Algorithm

Before testing different model estimation methods, we need to extract and match the feature points of the experimental objects first. In the process of feature point extraction, we use Oriented FAST and Rotated BRIEF (ORB) algorithm. This algorithm has good rotational invariance, anti-noise, and high computational efficiency [7]. Compared with other feature extraction algorithms, such as SIFT and SURF, ORB is more efficient on mobile devices and can perform well in real-time applications [7]. In the process of feature point matching, Fast Library for Approximate Nearest Neighbors (FLANN) algorithm is used. This algorithm is suitable for the situation with a large number of feature points and is faster than the Brute force algorithm [8].

3.2.1. Feature point extraction

- ORB algorithm

ORB is an algorithm for feature point detection that is a fusion of the FAST key point detector and the BRIEF descriptor [7]. It uses FAST (Features from Accelerated Segment Test) algorithm to detect interest points quickly. A rotating BRIEF descriptor is used to encode the image area around the feature points. Therefore, ORB not only has rotation invariance, but also has better matching results under illumination changes (Table 1).

Table 1. ORB procedure

ORB algorithm
1. FAST algorithm is used to detect key points in images. Then apply the Harris Angle quantity to find the top N points [9].
2. Through the gradient information of pixel brightness, the main direction is calculated for each detected key point. The moment is then calculated to improve rotation invariance
3. The rotation matrix is calculated with the direction of patch , and the neighborhood of each key point is encoded with BRIEF descriptors to generate a set of binary feature descriptors.
4. According to a certain threshold and priority, the most representative feature points are selected for matching.

3.2.2. Feature point matching

- FLANN algorithm

FLANN supports a variety of nearest neighbor search algorithms, such as KD-tree, K-means clustering, etc. [10], which can dynamically select the optimal search mode according to data characteristics. FLANN is particularly suitable for dealing with large-scale feature database scenarios, such as image registration, feature matching, etc., especially in the case of image rotation, translation and scale change (Table 2).

Table 2. FLANN procedure

Content
1. The feature descriptors extracted by the ORB are input into FLANN to build the index. This process, through clustering and other efficient data structures, can greatly speed up queries and significantly reduce computational complexity compared to brute force searches.
2. Approximate search by index can quickly find the nearest neighbor in feature matching and improve the search speed.
3. Finally, the best matching pair of feature points is returned

3.2.3. Model estimation methods This paper selects three evaluation methods, namely RANSAC estimation method, least square method and LMedS orientation, and these methods will be tested later.

- RANSAC algorithm

RANSAC (Random Sample Consistency Algorithm) is a method used to estimate the parameters of a mathematical model and can be used to deal with situations where there are a large number of outliers in the data. It generates solutions by using a collection of observations (data points) that estimate the underlying model parameters as small as possible (Table 3).

Table 3. RANSAC procedure

RANSAC algorithm
<ol style="list-style-type: none"> 1. A minimum number of points are randomly selected from the input data set to estimate the model parameters. For plane fitting, select three points. 2. Solve the model parameters. Assuming a linear model $y=mx+b$, the slope m and intercept b can be calculated from the selected points. 3. Calculate the distance from each point to the model. Use a threshold value ϵ to determine whether a point is an inlier. If the distance from a point to the model is less than ϵ, the point is considered to be an interior point. $d_i = y_i - (mx_i + b) < \epsilon$ <p>Where d_i is the distance from the i th point to the model and (x_i, y_i) is the data point.</p> <ol style="list-style-type: none"> 4. Count the number of inner points, and record the set of inner points of the current model, if the proportion of the inner number to the total number of points in the set exceeds the predefined threshold τ, then reestimate the model parameters using all the identified inner number and terminate. 5. repeat steps 1 to 4(up to N times).

For the number of iterations N , a suitable value can be set to ensure that the probability p (usually 0.99) of at least one random sample contains no outliers. Let u represent the probability that some of the data points are interior points, $v = 1 - u$. The minimum number m needs to be iterated N times,

$$1-p=(1-u_m)^N \quad (1)$$

And then

$$N=\frac{\log(1-p)}{\log(1-(1-v)^m)} \quad (2)$$

- least square algorithm

The Least Squares Method is a basic mathematical technique widely used in data analysis, statistics, and regression modeling to determine the best fit curve or line for a given set of data points [11]. It can find the best fit model by minimizing the sum of squares of error between the observed values and the predicted values of the model (Table 4).

Table 4. Least square algorithm procedure

least square algorithm
<ol style="list-style-type: none"> 1. Suppose there is a set of observed data points (x_i, y_i) where $i=1, 2, \dots, n$. And a linear model $y=mx+b$ to fit these data points, where m is the slope and b is the intercept. 2. For each observation point (x_i, y_i), the predicted value of the model is $\hat{y}_i = mx_i + b$. residual is defined as the difference between the observed and predicted values: $e_i = y_i - \hat{y}_i = y_i - (mx_i + b)$ 3. To eliminate the effect of positive and negative errors, we square each error: $e_i^2 = (y_i - (mx_i + b))^2$

Table 4. (continued).

4. The total square error (SSE) is the sum of the square errors of all observation points:

$$SSE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - (mx_i + b))^2$$

5. By changing the values of m and b, the total square error SSE is minimized. The optimal solution can be obtained by taking the partial derivative of SSE with respect to m and b and setting it to zero.

- LMedS algorithm

LMedS (Least Median of Squares) is a robust regression algorithm. In the presence of Gaussian noise, its efficiency can be improved. Monte Carlo acceleration technology can also reduce the high time complexity of LMedS algorithm. The basic idea is to reduce the impact on outliers by minimizing the median rather than the mean (Table 5).

Table 5. LMedS algorithm procedure

LMedS algorithm
1. N sample subsets are randomly drawn from the sample
2. The maximum likelihood is used for each subset to calculate the model parameters and the deviation from that model, usually a linear model.
3. Calculate the bias of the sample with the most bias among all samples of the model (i.e., Med bias)
4. Repeat steps 1 through 3 until a set maximum number of iterations is reached or the median change falls below a certain threshold
5. Finally, model parameters corresponding to the smallest Med deviation in N sample subsets are selected as the final output

Let X be the set of input points, y be the corresponding output value, and \hat{y} be the output value predicted by the model. The goal of LMedS is to minimize the following objective functions:

$$\hat{\theta} = \arg \min_{\theta} \text{median} ((y_i - \hat{y}_i(\theta))^2) \quad (3)$$

Where $\hat{y}_i(\theta)$ is the predicted value generated by the model parameter θ .

4. Experiment

4.1. Image selection

The data source of this study is the photos taken in different scenes in daily life, such as the light and dark, and the complexity of the pictures. The purpose is to explore the performance of each model estimation method in screening outliers under different scenes. RANSAC, least square method and LMedS method were used to evaluate the experiment. The evaluation index is the robustness of the algorithm and the calculation time. This report selects the second group of pictures for display, and introduces the experimental data of each group. Figure 1, Figure 2 Figure 3 and Figure 4 show the four sets of photos taken.



Figure 1. Bright complex picture group (Photo/Picture credit : Original)



Figure 2. Bright and simple picture group (Photo/Picture credit : Original)

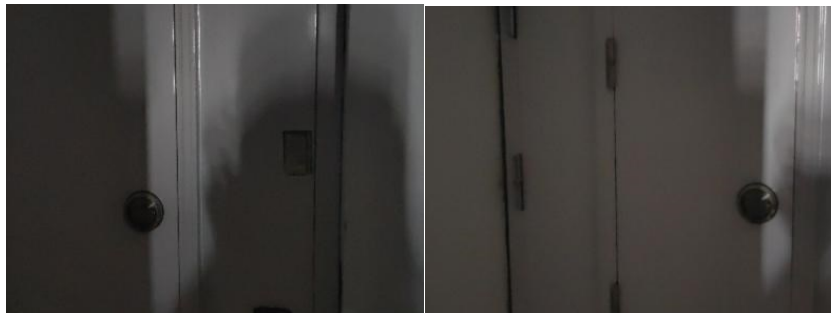


Figure 3. Dim complex picture group (Photo/Picture credit : Original)

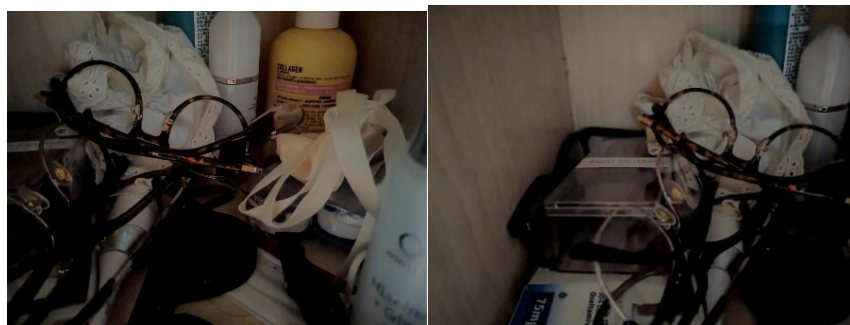


Figure 4. Dim simple picture group (Photo/Picture credit : Original)

4.2. Results

The second group screened outliers for display (Figure 5):



Figure 5. The second group of comparison, from top to bottom are RANSAC, least square, LMedS algorithm (Photo/Picture credit : Original)

4.3. Result analysis and discussion

Lowe's ratio $\text{testratio_thresh} = 0.6$ is set when FLANN algorithm is used for feature point screening. In order to unify variables, the threshold of RANSAC algorithm and LMedS algorithm is set to 6, and the least square method cannot set the threshold. The reprojection error in the experimental data can reflect the robustness of the algorithm. The larger the error, the lower the robustness (Table 6).

Table 6. Algorithm comparison in bright and complex case

Performance	RANSAC	Least square	LMedS
run time	0.006587	0.010068	0.014542
Mean Reprojection Error	17.33506	35.17632	17.33584

In the bright and complex first environment, the reprojection errors of the different algorithms are arranged as follows: $\text{RANSAC} < \text{LMedS} < \text{Least square}$, so their robust arrangements can also be obtained: $\text{RANSAC} > \text{LMedS} > \text{Least square}$. At the same time, the experiment also measured the

running time of each algorithm, which is arranged as follows: LMedS > Least square Least square > RANSAC. In summary, RANSAC's robustness and operating efficiency are superior to the other two algorithms in bright and complex environments. RANSAC is the environment of choice because it can process data sets containing outliers more efficiently and in less time.

Table 7. Algorithm comparison in bright and simple case:

Performance	RANSAC	Least square	LMedS
run time	0.002998	0.000993	0.004633
Mean Reprojection Error	15.41975	27.08029	18.27344

When encountering a bright and simple environment, the reprojection error shown in the experiment is as follows: RANSAC< LMedS< Least square, and the corresponding robustness is as follows: RANSAC> LMedS> Least square. the running time tested by each algorithm in order from small to large is: Least square < RANSAC < LMedS. In contrast, RANSAC is also the most suitable algorithm for screening in bright and simple environments, which can be described as the optimal algorithm in bright environments (Table 7).

Table 8. Algorithm comparison in dim and simple case

Performance	RANSAC	Least square	LMedS
run time/s	0.003004	0.001544	0.002614
Mean Reprojection Error	144.66176	205.12672	143.12581

Due to the dark and complex environment, the reprojection error and running time become much larger, and the reprojection error arrangement of the algorithm is shown: LMedS< RANSAC< Least square, so the robustness is: LMedS> RANSAC> Least square. and the running time ordering can also be obtained: Least square< LMedS< RANSAC. Although Least square algorithm has the shortest running time, its robustness is the worst, which will lead to a large error in the result, so in contrast, LMedS is more high-performance, if there are requirements for the running time, and there are not many outliers, it can try to use Least square algorithm, and finally, in the dim and complex case, LMedS has become the best algorithm (Table 8).

Table 9. Algorithm comparison in dim and complex cases

Performance	RANSAC	Least square	LMedS
run time/s	0.003998	0.001001	0.003010
Mean Reprojection Error	51.13642	63.34874	51.38120

The last case is a dark and simple environment, and the experimental results in this environment are a little complicated. First, the gravity projection error is arranged as follows: RANSAC< LMedS< Least square, and then the robust ranking is obtained: RANSAC> LMedS> Least square. The running time will also be compared: RANSAC > LMedS> Least square (Table 9). RANSAC algorithms, although slower in terms of run time, are usually the best choice due to their superior performance in robustness, especially in the case of large noise and outliers. If running time is a critical factor when using the algorithm, and a high reprojection error is acceptable, the use of least squares may be considered. If these two kinds of performance are considered at the same time, the compromise LMedS algorithm can be chosen. In general, the RANSAC algorithm is the best solution, because the difference in time is not very large.

4.4. Experimental discussion summary

In this experiment, the experimental images are divided into four different situations, which are: bright complex environment, bright simple environment, dark complex environment, dark simple environment.

The test is on the reprojection error (i.e., robustness) and runtime of each algorithm. During testing, RANSAC algorithm is the most robust in most environments, but it is less efficient in dim environments. In addition, LMedS is moderately superior in both robustness and efficiency in all cases, and is even better than RANSAC in dim and complex environments. At last, the efficiency of least squares algorithm is always high, but its robustness is relatively low, so the error of experimental results is large.

5. Conclusion

This paper discusses the performance of different algorithms in different environments by obtaining image database from mobile devices. The experimental environment includes bright complex environment, bright simple environment, dark complex environment and dark simple environment. The results show that RANSAC algorithm shows high robustness in most environments, but low efficiency in dim environments. The LMedS algorithm showed moderate superiority in all cases and was superior to RANSAC in dim and complex environments. In contrast, the least square algorithm always maintains a high efficiency, but its experimental results have a large error. In future investigations, researchers could consider using a combination of Least Squares to quickly obtain preliminary results, and then using RANSAC to further optimize and improve robustness, especially when working with more complex data sets.

References

- [1] DelSole, T., & Tippet, M. (2022). Linear Regression: Least Squares Estimation. In *Statistical Methods for Climate Scientists* (pp. 185–209). Cambridge University Press. <https://doi.org/10.1017/9781108652078.010>
- [2] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91-110. <https://doi.org/10.1023/B:VISI.000.0029664.99615.94>
- [3] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3), 346-359. <https://doi.org/10.1016/j.cviu.2007.09.014>
- [4] Verma, C. S., & Mon-Ju. (2009). Panoramic Image Mosaic. Retrieved from https://pages.cs.wisc.edu/~csverma/CS766_09/ImageMosaic/doc.pdf
- [5] Chen, Y., Wang, G., & Wu, L. (2019). Research on feature point matching algorithm improvement using depth prediction. *The Journal of Engineering*, 2019(23), 8905–8909. <https://doi.org/10.1049/joe.2018.9142>
- [6] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. 2011 International Conference on Computer Vision (pp. 2564-2571). <https://doi.org/10.1109/ICCV.2011.6126544>
- [7] Meer, P., Mintz, D., Rosenfeld, A., & Kim, D. Y. (1991). Robust Regression Methods for Computer Vision: A Review. *International Journal of Computer Vision*, 6(1), 59-70.
- [8] Karami, E., Prasad, S., & Shehata, M. (2023). Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. <https://arxiv.org/pdf/1710.02726>
- [9] Suju, A., & Jose, H. (2017). FLANN: Fast approximate nearest neighbor search algorithm for elucidating human-wildlife conflicts in forest areas. 2017 International Conference on Signal Processing and Communication Networks (ICSCN) (pp. 1-6). <https://doi.org/10.1109/ICSCN.2017.8085676>
- [10] GeeksforGeeks. (2024). Least Square Method Definition graph and formula. Retrieved from <https://www.geeksforgeeks.org/least-square-method/>
- [11] Derpanis, K. G. (2010). Overview of the RANSAC Algorithm. Retrieved from http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf