2024 4th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS) | 979-8-3503-5998-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACCTCS61748.2024.00064

Real-time Multi-camera Video Stitching Algorithm with Adaptive Sensing of Illumination and Dynamic Objects

Min Lei
Department of Information and Communications Engineering
Xi'an Jiaotong University
Xi'an, China
2902301637@qq.com

Lijun He
Department of Information and Communications Engineering
Xi'an Jiaotong University
Xi'an, China
lijunhe@mail.xjtu.edu.cn

Fan Li
Department of Information and Communications Engineering
Xi'an Jiaotong University
Xi'an, China
lifan@mail.xjtu.edu.cn

***Abstract*—With the development of digital image processing technology, the demand and application of wide-view video continue to grow. However, the existing stitching algorithms may result in unnatural appearance, misalignment, and ghosting issues when stitching scenes with inconsistent brightness or dynamic objects. To address the above problems, we propose a real-time multi-camera video stitching algorithm with adaptive sensing of illumination and dynamic objects. Within the proposed illumination compensation module, brightness equalization across varying illumination conditions is attained through the application of gain compensation to image blocks. We propose a histogram-based dynamic object detection approach to dynamically detect the presence of moving objects along the stitching seam. This method facilitates the adaptive detection of dynamic entities by analyzing the histogram disparities between the local vicinity of the stitching seam and the broader image context. The experimental results show that it can effectively solve the problem of unnatural stitching results arising from the uneven brightness of the image, and can effectively detect the presence of dynamic objects near the stitching seam so that the effect of the video stitching has been improved.**

Keywords—video stitching, gain compensation, dynamic object detection, histogram

I. Introduction

In recent years, with the rapid development of camera technology and the continuous reduction of cost, multi-camera systems have been widely used in video surveillance, virtual reality, augmented reality, and other fields. Multi-camera system can provide richer information than a single camera. With the wide application of multi-camera systems, multi-camera video stitching technology came into being. Video stitching refers to a method of stitching video frames with overlapping regions acquired by multiple cameras from different viewpoints at the same moment into a wide-field-of-view video frame[1]. In practice, multi-camera video stitching technology faces the challenges posed by light variations and moving objects. The presence of light variations and moving objects can seriously affect the quality of multi-camera video stitching. Because the differences in cameras and light intensity can lead to uneven brightness within or between an image, the stitching image will have a contrast boundary. Meanwhile, if there are moving objects in the overlapping region of the video frames, the stitching result will have blurring and ghosting problems at the stitching seam.

For image video stitching, many different methods have been developed. Within the domain of image and video stitching technology, the two fundamental components are image alignment and image fusion[2]. In image alignment technology, the detection of the feature point is the most critical step. Harris angle point detection method was proposed in [3]. The algorithm is simple, but the detected feature points do not have scale invariance. The SIFT algorithm was proposed in [4]. The algorithm overcomes the shortcomings of the Harris algorithm and the proposed feature points have angle invariance and scale invariance. But the algorithm is computationally complex and the feature extraction speed is slow. A faster SURF algorithm was proposed in [5]. The algorithm uses Haar features and the concept of integral image, greatly improving the speed of the algorithm. The authors in [6] designed the ORB algorithm and the algorithm greatly improves the speed of the feature point detection at the expense of the cost of scale invariance. To improve the accuracy of feature matching, the authors in [7] proposed the RANSAC (Random Sample Consensus) algorithm to further eliminate the mismatched pairs during feature matching. In terms of image fusion, weighted fusion is the simplest class of image fusion algorithms. The algorithm has low time complexity but the fusion will appear the phenomenon of fuzzy edges. The authors in [8] proposed a multi-band fusion algorithm, which can avoid the blurring, ghosting, and other problems that occur in the process of stitching. But the amount of computation is large. However, none of the above methods take into account the interference of moving objects and light on video stitching. At the same time, there are some studies on the effect of moving objects and light on the video stitching effect. The authors in [9] designed the gain compensation for image stitching. But the authors did not consider the effect of moving objects. The authors in [10] complete the video stitching of smaller moving objects. The authors in [11] proposed an algorithm based on the background separation of the video stitching of moving objects. The method for the video of the stationary background of the video plays a good role. But the method can not work for the scenes where the camera is moving. In the case where the camera itself is in motion, how to effectively stitch video frames with moving objects and uneven light is still a problem to be solved in the field of video stitching.

979-8-3503-5998-5/24/$31.00 ©2024 IEEE
DOI 10.1109/ACCTCS61748.2024.00064

To address the existing problems, we propose a real-time multi-camera video stitching algorithm with adaptive sensing of illumination and dynamic objects for moving multi-camera systems where the relative positions of the cameras are fixed with the following innovations:

(1) **Stitching seam updating based on adaptive sensing of moving objects:** Aiming at the problem of blurring and ghosting in the stitching result due to moving objects falling in the overlapping region in the image, a moving object detection algorithm based on histogram is proposed. The stitching seam is dynamically updated according to the detection result.

(2) **Illumination compensation based on image chunk brightness equalization:** To solve the problem of uneven brightness of the picture, we introduced the illumination compensation module. Through the function of the module to equalize the brightness of the picture, we realized the smooth transition of the stitching result.

II. Method

A. Video Stitching Framework

For the multi-camera system in which the relative positions between cameras are unchanged but the actual physical positions change, the multi-camera video real-time stitching algorithm proposed in this paper is shown in Fig. 1.

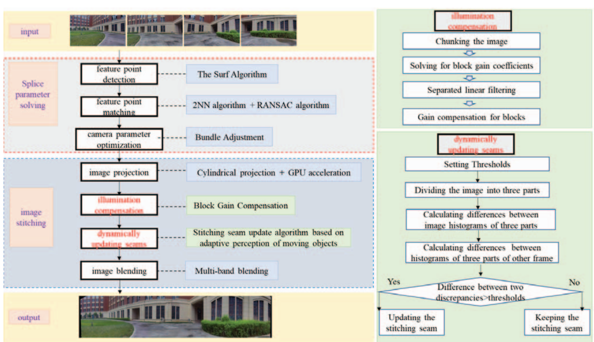


Fig. 1. Flowchart of multi-camera real-time video stitching

Considering that the relative position of each camera is fixed between multi-camera systems, the stitching parameters between video frames acquired at the same moment are fixed. Therefore, the algorithm in this paper only acquires the first video frame to solve the stitching parameters, and each subsequent video frame is automatically stitched using the solved stitching parameters. Solving the parameters of the camera is divided into the following steps. Firstly, use the Surf algorithm to select the image feature points. Then use the 2NN algorithm to coarsely match the pairs of feature points and use the RANSAC[7] algorithm to reject the mismatched points. Finally, use the bundle adjustment method[12] to globally optimize the camera parameters to obtain more accurate stitching parameters. Image stitching is mainly divided into the following steps. Firstly, the image is transformed to the same plane by cylindrical projection. Then, the block gain compensation is performed on the to-be-stitched pictures to achieve brightness equalization. Finally, the stitching seam updating algorithm based on the adaptive sensing of moving objects is used to determine whether there is a moving object in the vicinity of the existing stitching seams. If there is one, the algorithm will re-find the stitching seams. Then the final fusion of the image is performed using a multi-band fusion method. Next, we will focus on the illumination compensation module and stitching seam updating algorithm based on the adaptive perception of moving objects.

B. Illumination Compensation

If multiple images to be stitched have different exposures, this can result in a clear separation of light and dark in the stitching result. To make the stitching image more natural, we need to compensate for the exposure of each image so that each image has the same exposure. In the algorithm of this paper, we introduce the gain compensation method to compensate for the illumination of each picture. Gain compensation is to assign a gain coefficient to each image so that the lightness of the overlapping part of the image is equal or similar. It establishes the error function as (1)[9]:

$$e = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\substack{u_i \in R(i,j) \\ \tilde{u}_i = H_{ij}\tilde{u}_j}} [g_i I_i(u_i) - g_j I_j(u_j)]^2 \qquad (1)$$

Where g_i and g_j are the gain coefficients of the image i and the image j. $R(i,j)$ denotes the overlapping part of the image i and the image j. $I_i(u_i)$ denotes the average value I_{ij} of the intensity of the image i in the overlapping part $R(i,j)$.

$$I_{ij} = \frac{\sum_{u_i \in R(i,j)} \sqrt{R^2(u_i) + G^2(u_i) + B^2(u_i)}}{N_{ij}} \qquad (2)$$

Where R/G and B denote the intensity values of the red, green, and blue components of the color image, N_{ij} denotes the number of pixels in the overlapping part $R(i,j)$.

In (1), if g is 0, the error function is 0. However, this is not the result we want, and we use its empirical formula as (3)[9],

$$e = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} N_{ij}\left[\frac{(g_i I_{ij} - g_j I_{ji})^2}{\sigma_N{}^2} + \frac{(1-g_i)^2}{\sigma_g{}^2}\right] \qquad (3)$$

In (3), σ_N and σ_g denote the standard deviation of the error and gain, respectively. The equation for $g_1, g_2, \ldots g_n$ is obtained by taking the above equation so that its derivative is zero. Solving this system of linear equations yields n gain coefficients. We build on the above gain compensation by separating the image into [32x32] chunks and compensating the image for gain in chunks. Each chunk has a gain coefficient. The result of the block compensation is that the image will be in the form of blocks, so we need to perform a smoothing filter between each block of the image to get a natural compensation result.

C. Stitching Seam Updating Algorithm Based on Adaptive Perception of Moving Objects

When stitching video frames, if it is based on a fixed stitching seam, we find that the stitching result will have ghosting and misalignment when a dynamic object falls on the stitching seam. However, if the finding of seams is done for each video frame, it will result in a low real-time performance of this video stitching algorithm. Therefore, we try to judge whether dynamic objects are falling on the stitching seam according to the image texture. If the camera is fixed, we can use the frame difference method and the background modeling method to easily distinguish the background and the moving foreground. However, when the camera is moving, the above method cannot detect the dynamic object well. In this paper, we introduce an innovative algorithm for updating stitching seams, grounded in histogram-based moving object detection. The algorithm is mainly based on the fact that if there is a dynamic object falling near the seam, the pixel value distribution near the seam is significantly different from the pixel value distribution of other parts. According to this feature, the following algorithm flow is designed :

Step 1: Set the threshold t and the width of the stitching seam margin w. The image width is W.

Step 2: Determine the range of horizontal coordinates $[x_{\min}, x_{\max}]$ of the pixel points of the current stitching seam;

Step 3: Divide the current image to be three parts, namely the left region, whose horizontal coordinate range is $[0, x_{\min} - w/2]$, the seam region, whose horizontal coordinate range is $[x_{\min} - w/2, x_{\max} + w/2]$, and the right region, whose horizontal coordinate range is $[x_{\max} + w/2, W-1]$.

Step 4: Compute the histograms of these three parts and calculate the histogram difference between the seam region and the left region as $dl1$,and the difference between the seam region and the right region as $dr1$.

The above four steps are the operations that need to be done after solving the camera parameters in the first frame. The following steps describe how to determine whether to update the stitching seam in each subsequent frame.

Step 5: Divide the current image into three parts according to step 3. The histogram distribution is counted separately. And the histogram differences between the stitching seam region and the left and right regions are calculated as dl, dr respectively.

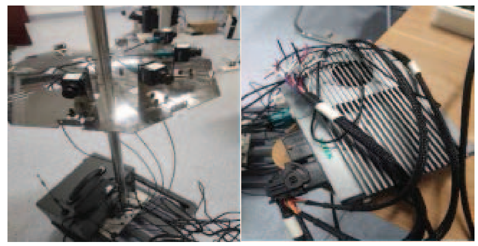
Step 6: Compare dl, dr with $dl1$, $dr1$. If the difference is greater than the threshold t, the stitching seam needs to be updated, as well as the range of the stitching seam $[x_{\min}, x_{\max}]$,and $dl1$, $dr1$.

The above shows the entire process of using histograms to detect whether a dynamic object falls on the stitching seam or not. Wherein the use of the image histogram of the region near the stitching seam and the difference between the image histograms of the two sides as a criterion for comparison is to prevent the change of the image histogram brought about by the scene switching. The algorithm can timely determine the appropriate need to update the stitching seam. The complexity is low, suitable for scenes with high real-time requirements.

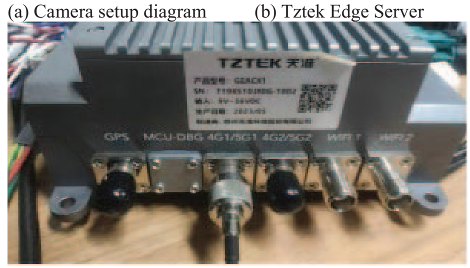
III. Experiments and analysis of results

We use the Tztek edge server GEACX1, which is an AI edge computing device developed based on an NVIDIA Jetson Orin embedded GPU module. It has 200 TOPS of arithmetic power and also carries four Sening imx390 cameras. All four cameras are fixed on a stand at the same horizontal height and the experimental setup is shown in Fig 2.

The experiment is realized based on OpenCV3.4.10, the system version is ubuntu18.04, the programming language is C++, and the programming environment is VSCode.



(a) Camera setup diagram (b) Tztek Edge Server

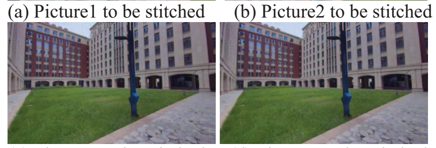


(c) The specific parameters of edge server

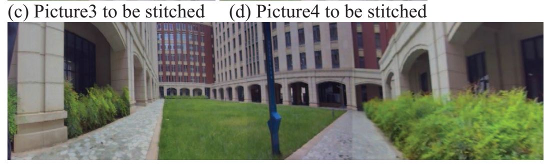
Fig. 2. Physical diagram of experimental equipment

A. Image Stitching Results

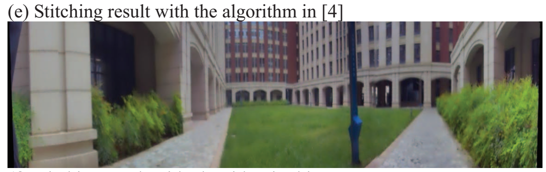
(a) Picture1 to be stitched (b) Picture2 to be stitched



(c) Picture3 to be stitched (d) Picture4 to be stitched



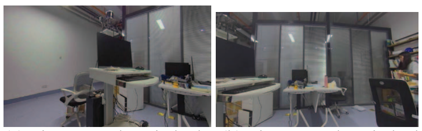
(e) Stitching result with the algorithm in [4]



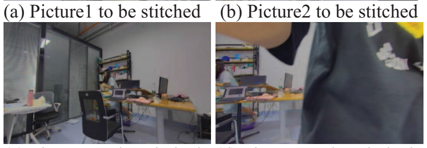
(f) Stitching result with algorithm in this paper

Fig. 3. Stitching results for static scenes

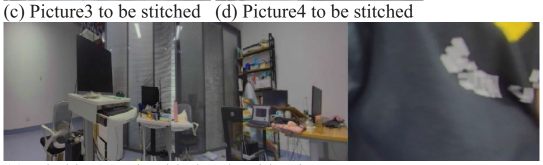
We use the above experimental equipment to collect images from multiple scenes for stitching. The stitching results prove that our algorithm works well for static scenes. And it is also robust to the presence of dynamic objects in the scene. We compared the results of this paper's algorithm with direct stitching using the sift algorithm[4] in a static scenario. As shown in Fig. 3, the results of stitching directly using the sift algorithm show a clear demarcation, while the stitching results of this paper's algorithm have a more natural and smooth transition. We also compared the stitching results of this paper's algorithm in scenes containing dynamic objects, which also outperform the results of stitching using the sift algorithm directly. As shown in Fig. 4, our algorithm works better at the stitching seam for dynamic objects.



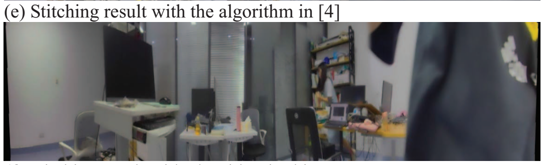
(a) Picture1 to be stitched (b) Picture2 to be stitched



(c) Picture3 to be stitched (d) Picture4 to be stitched



(e) Stitching result with the algorithm in [4]



(f) Stitching result with algorithm in this paper

Fig. 4. Stitching results for scenes with dynamic objects

B. Feature Point Detection Results and Comparison Experiments

We have performed feature point detection using the sift[4] algorithm surf[5] algorithm and orb[6] algorithm respectively and compared their results as shown in Figs. 5,6,7. It can be seen that the features extracted by the sift and surf algorithms are more detailed and evenly distributed, while a large part of the features extracted by the orb algorithm are densely concentrated in localized areas. According to TABLE I, it can be seen that the surf algorithm extracted feature points run faster compared to the sift algorithm, this is because the surf algorithm uses the concept of integral image in the feature extraction process, and at the same time, the dimension of the feature points will be reduced. Therefore, it is concluded that the surf algorithm performs better in feature extraction.

(a)Figure a (b) Figure b

Fig. 5. Results of feature point detection using sift algorithm

(a)Figure a (b) Figure b

Fig. 6. Results of feature point detection using surf algorithm

(a)Figure a (b) Figure b

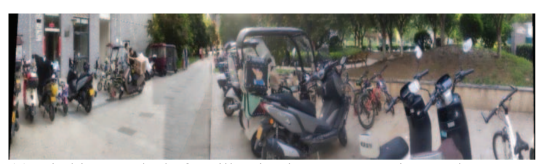
Fig. 7. Results of feature point detection using the orb algorithm

TABLE I. PERFORMANCE COMPARISON OF THREE FEATURE POINT DETECTION ALGORITHMS

Algorithm	**Image To Be Detected**	**Number Of Extracted Features**	**Time For Feature Extraction/(ms)**
sift	Figure a	1511	334.295
	Figure b	1006	197.693
surf	Figure a	1501	127.836
	Figure b	1337	13.265
orb	Figure a	1168	172.609
	Figure b	1061	54.743

C. Ablation Study of Illumination Compensation Effect

For the illumination compensation module encapsulated within the algorithm, a selection of indoor and outdoor scenes exhibiting disparate luminance levels was utilized to benchmark the efficacy of image stitching, pre- and post-integration of said module. The experiment proved that the block gain compensation has a good effect on equalizing the pictures with different brightness taken by multiple cameras, and it can eliminate the obvious light and dark boundaries after stitching due to the brightness difference. Fig 8 shows the effect of stitching before and after illumination compensation in an outdoor scene. Fig. 9 shows the effect before and after illumination compensation in an indoor scene.



(a) Stitching results before illumination compensation outdoor

(b) Stitching results after illumination compensation outdoor

Fig. 8. Comparison of stitching results before and after light compensation for outdoor scenes

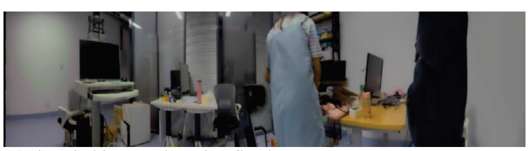
(a) Stitching results before illumination compensation indoor

(b) Stitching results after illumination compensation indoor

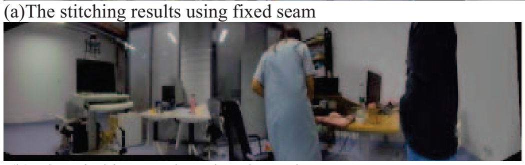
Fig. 9. Comparison of stitching results before and after illumination compensation for indoor scenes

D. Stitching Seam Dynamic Update Effect

In the use of a fixed stitching seam, if a dynamic object falls on the stitching seam, the stitching result will be misaligned and blurred near the seam, as shown in Fig. 10(a). As shown in Fig. 11, we can see that at this time, the stitching seam is just across the dynamic objects. When we use the algorithm in this paper, the histogram distribution of the region around the stitching seam of the first video frame with the rest of the image is shown in Fig. 12(a). The histogram of each region of the video frame that needs to update the stitching seam is shown in Fig. 12(b), where the red curve represents the histogram distribution of the left side of the image, the green curve represents the histogram distribution of the image near the stitching seam, and the blue curve represents the histogram distribution of the right side of the image. The algorithm detects the dynamic objects successfully and re-finds the stitching seam. The stitching result is shown in Fig. 11(b). As shown in Fig. 13, we can see that our stitching seam avoids dynamic objects at this time. The experiment proves that our algorithm is effective in determining whether there are dynamic objects near the stitching seam.

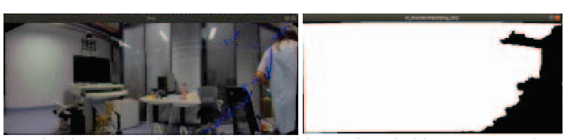


(a)The stitching results using fixed seam



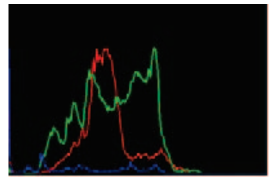
(b) The stitching results using dynamic seam

Fig. 10. The stitching results comparison by using fixed and dynamic stitching seams

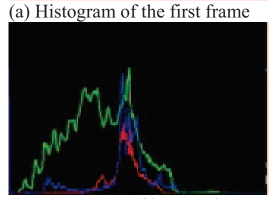


(a)Stitching seams through dynamic objects (b) Stitching Seam Mask

Fig. 11. Stitching seams and masks through dynamic objects

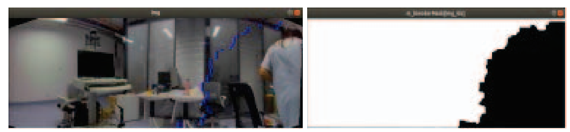


(a) Histogram of the first frame



(b) Histograms of images that need to be updated with stitching seams

Fig. 12. Histogram distribution



(a) Updated stitching seams (b) Stitching Seam Mask

Fig. 13. Updated stitching seams and masks

IV. Conclusion

This paper proposes a real-time video stitching algorithm that can be applied in practical engineering for moving multi-camera systems composed of multiple cameras with fixed relative positions. The proposed algorithm tackles the challenge of inconsistent luminance across images during the stitching process, which manifests as conspicuous shadows in the resultant panorama. To address this, a block gain compensation technique is introduced, effectively normalizing the brightness of individual images to yield a more seamless and natural-looking composite. Furthermore, the algorithm delineated in this work offers a robust solution to mitigate the perturbations caused by dynamic entities during video stitching. The algorithm compares histogram distributions between the region proximal to the stitching seam and the remaining image areas. This enables the effective identification of dynamic objects adjacent to the stitching seam and ascertains the necessity for seam updates.

Acknowledgment

This study was supported in part by the National Key R&D Program of China under Grant 2022ZD0115802, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2023-JC-JQ-51, the Central Guidance on Local Science and the Technology Development Fund under Grant 2022ZY1-CGZY-01HZ01, the Key R&D Project in Shaanxi Province under Grant 2023-ZDLNY-65.

References

[1] Wei L Y U, Zhong Z, Lang C, et al. A survey on image and video stitching[J]. Virtual Reality & Intelligent Hardware, 2019, 1(1): 55-83.

[2] Zhang Hang, Zhao Mingchang. Panoramic Image Stitching Using Double Encoder–Decoders[J]. SN Computer Science,2021,2(2).

[3] Harris C, Stephens M. A combined corner and edge detector[J]. Proc Alvey Vision Conf, Univ. Manchester, 1988(3):147-151.

[4] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.

[5] Bay H, Tuytelaars T, Van Gool L, et al. SURF: Speeded up Robust

Features[C]. In: European Conference on Computer Vision, 2006: 404-417.

[6] Rublee E, Rabaud V, Konolige K, et al. ORB: An Efficient Alternative to SIFT or SURF[C]. In: International Conference on Computer Vision, 2011: 2564-2571.

[7] D. Capel, A. Zisserman. Automated mosaicking with super-resolution zoom[C]. Computer Vision and Pattern Recognition Conference on IEEE, Venice, 1998:885-891.

[8] Burt P J, Adelson E H. A Multiresolution Spline with Application to Image Mosaics[J]. ACM Transactions on Graphics, 1983, 2(4):217-236.

[9] Brown M , Lowe D G .Automatic Panoramic Image Stitching using Invariant Features[J].International Journal of Computer Vision, 2007, 74(1):59-73.DOI:10.1007/s11263-006-0002-3.

[10] Irani M, Anandan P. Parallax geometry of pairs of points for 3D scene analysis[C]. European Conference on Computer Vision. Springer, Berlin, Heidelberg, 1996.

[11] Hui S, Kankanhalli M S, Srinivasan S H, et al. Mosaic-based view enlargement for moving objects in moving pictures[C]. IEEE International Conference on Multimedia & Expo. IEEE, 2004.

[12] Li Y, Fan S, Sun Y, et al. Bundle adjustment method using sparse BFGS solution[J]. Remote Sensing Letters, 2018, 9(8):789-798.