PRESCAN Adaptive Vehicle Image Real-Time Stitching Algorithm Based on Improved SIFT

Qian Li, College of Information Engineering, Shanghai Maritime University, China Yanli Xu, College of Information Engineering, Shanghai Maritime University, China* Pengren Ding, Shanghai Zhongzhixin Technology Co., China

ABSTRACT

Aiming at the problems of high latency, low accuracy, and obvious stitching gaps of traditional image stitching techniques in PRESCAN, this paper proposes an adaptive real-time vehicle image stitching algorithm based on improved scale-invariant feature transform (SIFT). Firstly, a multi-threaded structure is utilized to combine the improved adaptive features from accelerated segment test algorithm with SIFT descriptors for feature extraction. Secondly, a mismatching filtering strategy based on the BF algorithm has been introduced. Finally, image stitching is performed by combining random sample consensus algorithm and adaptive region stitching strategy. The results show that the recall rate is about 30.64%, the accuracy of matching is about 98.86%, and the total stitching time is about 22.82ms. It effectively improves the quality of feature point extraction and matching accuracy and significantly reduces the stitching time of images. It provides a real-time and accurate vehicle image stitching method for the field of perception in the autonomous driving laboratory environment.

KEYWORDS

Feature Extraction, Feature Matching, Image Processing, Image Stitching, Real Time

INTRODUCTION

With the continuous development of advanced driving assistance system (ADAS) technology in recent years, there is an increasing demand for real-time and accurate vehicle environment perception (Y. L. Zhang et al., 2022; Hu et al., 2019; Miao et al., 2020). Before ADAS technology can be safely implemented, simulation and validation of the relevant environment perception algorithms have become essential to ensure their safety and stability. Prescan is an autonomous driving scenario simulation software developed by Tass International, which has the advantages of simple operation, complete vehicle sensor models, and controlled weather visualization. It can customize various autonomous driving scenarios according to requirements and occupies an important place in the field

DOI: 10.4018/IJITSA.321754

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

of autonomous driving scenario simulation (C. S. Wang et al., 2021). At present, it is a hot topic for many technical researchers to achieve real-time and accurate vehicle image stitching in a laboratory environment to maximize the range of vehicle environment perception using existing information at a lower cost; thus, meeting the need to conduct tests of relevant environmental perception algorithms.

Considering the application scenario in this paper, the height of the vehicle camera is not always at the same level due to ground undulations and potholes during driving, so it is especially important to extract feature points with high rotation invariance. The scale-invariant feature transform (SIFT) algorithm proposed by Lowe (2004) performs better in terms of scale and rotation invariance in traditional image stitching techniques. Therefore, this paper is based on SIFT algorithm for the study of vehicle image stitching technology. Although the performance of scale invariance and rotation invariance of this algorithm is superior, there is high complexity with numerous computational efforts (Y. Y. Liu et al., 2022). To solve this problem, based on the SIFT algorithm, Y. L. Zhang and Xie (2021) extended the original 26 reference points in the Gaussian difference space to 32 reference points within the Manhattan distance, respectively. The descriptor model was reconstructed to reduce the traditional 128-dimensional SIFT feature descriptor to 64 dimensions, effectively reducing the complexity of the algorithm.

Y. Liu et al. (2022) used a bidirectional *k*-nearest neighbor matching algorithm to increase the correct matching rate and used the random sample consensus (RANSAC) algorithm to filter outliers to obtain a more accurate homography matrix. Yan and Ma (2022) used the singular value decomposition (SVD) algorithm to reduce the dimensionality of SIFT feature vector and the *k*-dimensional tree (*k*-d tree) algorithm for feature matching to obtain an image without stitching gaps. Yang et al. (2021) adopted the features from the accelerated segment test (FAST) algorithm to extract corner points and then used the speed-up robust feature (SURF) operator to determine the main direction to generate descriptors and added a pre-sampling step to the RANSAC algorithm to remove the mismatched pairs, which effectively improved the matching accuracy of the algorithm.

G. X. Zhang (2022) proposed an unmanned aerial vehicle (UAV) aerial image stitching algorithm based on semantic segmentation and oriented FAST and rotated BRIEF (ORB) to obtain foreground semantic information through semantic segmentation. The feature points are extracted using the quadtree decomposition idea and the classical ORB algorithm to delete the foreground feature points to achieve the matching of feature points. Then, the homography matrix and the weighted fusion algorithm are used to eliminate the stitching gaps. Y. B. Liu (2022) improved the efficiency of feature point detection by combining the binary robust invariant scalable keypoints (BRISK) algorithm with the matching factor and established a binary tree model to estimate the overlap area and to improve the accuracy of feature matching. H. P. Zhang (2022) proposed a local affine constraint-based image alignment method to establish feature descriptors for circular neighborhoods instead of square neighborhoods and used the local affine transform circular region search algorithm for more refined matching.

In summary, there has been abundant research conducted on image stitching techniques. However, research on real-time vehicle image stitching techniques in laboratory environments is relatively limited. Moreover, the real-time and stability performance in different backgrounds of these methods still need to be improved. Therefore, an adaptive SIFT-based real-time vehicle image stitching algorithm is proposed for the widely used Prescan autonomous driving scenario simulation software. Firstly, a low complexity and computational adaptive feature extraction algorithm is proposed based on the FAST and SIFT algorithms. Secondly, a high-accuracy matching strategy is applied based on the BF algorithm. Finally, a seamless stitching gap. Prescan constructs four weather scenarios of autonomous driving: sunny, cloudy, rainy, and snowy. The algorithm proposed in this paper is compared with SIFT and SURF algorithms. The results show that the algorithms in this paper can meet the requirements of real-time application scenarios and have good robustness in all kinds of weather conditions.

ALGORITHM MODEL

The real-time stitching algorithm for vehicle images proposed in this paper consists of two parts: First is the construction of scenes and the acquisition of datasets, which involves Prescan building the autonomous driving scenarios under different weather conditions and data acquisition carried out by the mounted vehicle camera. Second is the study of the vehicle image real-time stitching algorithm. As shown in Figure 1, the real-time stitching of vehicle images is completed through three modules: feature extraction, feature matching, and image stitching.

In the feature extraction module, combined with the application scenario of this paper, the improved adaptive FAST algorithm is used to detect the key points of the two-vehicle images to be stitched, and then it is combined with the SIFT descriptor to generate feature descriptors. In addition, for the two-vehicle images to be stitched, a multi-threaded parallel technique is used for feature extraction to reduce the time consumption of the algorithm. In the feature matching module, the Euclidean distance is used to describe the similarity between feature points. The brute-force (BF) matching algorithm and a strategy to filter out mismatches are used to find the pair with the highest mutual similarity. In the image stitching and fusion module, the RANSAC algorithm is used to remove outliers and to calculate the optimal homography matrix between the matched pairs. For the appearing stitching gaps, the authors combine the perspective transformation and the efficient adaptive area image stitching strategy to obtain an image without stitching gaps.

IMPROVED SIFT ALGORITHM FOR FEATURE EXTRACTION

In the traditional SIFT algorithm, the key points are detected by constructing a difference of Gaussian (DOG) space and comparing it with 8 adjacent points and a total of 26 points in the upper and lower layers to obtain discrete spatial extrema. Then, the Taylor expansion is used to derive the more accurate extreme points as the key points. This algorithm is highly complex, computationally intensive, and is not suitable for scenarios with high real-time requirements (Y. Liu et al., 2022). By analyzing the characteristics of SIFT feature extraction algorithm and the application scenario of this paper, an improved feature extraction algorithm for the adaptive FAST algorithm combined with the SIFT operator is proposed.

Improved Adaptive FAST Key Point Detection Algorithm

The conventional FAST algorithm is a simple and fast corner point detection algorithm proposed by Rosten and Drummond (2006). The algorithm defines a corner point as a pixel that may be a corner point if there are enough pixels in its surroundings that are in different grey areas. As shown in Figure 2, if the point to be processed is P, it will take 16 pixel points on the circumference of a circle with a radius of 3 pixels. Determine whether the pixel difference between $P_1 \sim P_{16}$ and P is greater than t: If it is, the point is considered a corner point.

Figure 1. Flow chart of the proposed algorithm



Figure 2. FAST feature point detection schematic



This algorithm has low complexity and good real-time performance, but the number of identified corner points is too large, and the phenomenon of corner point aggregation tends to occur (Jiao et al., 2022). In addition, the threshold value of t for determining whether it is a corner point is a fixed value set by humans based on experience, which causes the weak adaptability of this algorithm in different scenes. To solve the above problems, an improved adaptive FAST corner point detection algorithm is proposed. The specific steps are as follows.

Bilinear Interpolation Redefines the Input Vehicle Image Size

Bilinear interpolation is used to redefine the input size of the onboard image to reduce the computational effort of the algorithm while ensuring that the image texture is not lost. Firstly, the vehicle image is transformed into a grayscale image, and it is assumed that the source image size is $A \times B$ and the target image size is $C \times D$. The relationship can be expressed by Equation 1, then the pixel points (c, d) of the target image corresponds to the pixel point (a, b) of the source image can be expressed by Equation 2.

$$\frac{A}{B} = \frac{C}{D}$$

$$\begin{cases} a = \frac{A}{C} \times (c+0.5) - 0.5 \\ b = \frac{B}{D} \times (d+0.5) - 0.5 \end{cases}$$

$$(1)$$

The function of 0.5 is to ensure that the geometric centers of the two images can coincide. After getting the corresponding source pixels location, find the four closest pixel points (a_1, b_1) , (a_1, b_2) , (a_2, b_1) , (a_2, b_2) to this coordinate point, and calculate the pixel value V(c, d) of the target pixel by Equation 3.

$$V(c,d) = a_1 \times Z_1 + a_2 \times Z_2 + a_3 \times Z_3 + a_4 \times Z_4$$
(3)

where Z_i (i = 1, 2, 3, 4) represents the grayscale value of the nearest four pixels, and a_i (i = 1, 2, 3, 4) represents the weight coefficient consisting of the weights corresponding to the vertical direction and the horizontal direction. The closer the distance is, the higher the weight factor will be. The pixel values of all pixel points in the target image can be obtained through the above steps.

OTSU Algorithm Was Used to Obtain the Value of T

The OTSU (Pham et al., 2021) algorithm can obtain the value of T corresponding to the maximum inter-class variance of the frame, which is the best threshold value to distinguish the foreground from the background. For the input vehicle image I(x, y), the proportion of foreground pixel points less than the threshold value T to the whole image pixel points is ω_0 , and the average gray level is μ_0 . The proportion of background pixel points greater than the threshold value T to the whole image pixel points is ω_1 , and the average gray level is μ_1 . The total average gray level of the image is μ , and the interclass variance is g. Then the threshold T corresponding to the maximum interclass variance can be obtained through the following Equation 4.

$$g = \omega_0 \times \left(\mu_0 - \mu\right)^2 + \omega_1 \times \left(\mu_1 - \mu\right)^2 \tag{4}$$

Accelerated Screening of Candidate Corner Points

First, according to Equation 5, we determine whether the absolute value of the difference between points P_1 , P_9 and pixel P is greater than the value of t. If the value is smaller, the point is discarded. If it is greater, we continue to compare pixel points P_1 , P_5 , P_9 , P_{13} to find out whether at least three points and the absolute value of the difference between the pixel points P is greater than t. If it is smaller than t, the point is discarded; otherwise, we continue to compare whether at least 9 of the 16 surrounding pixel points have an absolute value greater than t of the difference of pixel point P. If satisfied, the point is assigned to the set of candidate key points.

$$\left|V\left(P_{1}\right)-V\left(P\right)\right|>t\tag{5}$$

where V(P) represents the gray value of the point to be processed, and $V(P_i)$ represents the gray value of the surrounding pixels.

Calculation of the Threshold t

For empirical and fixed thresholds of the traditional FAST algorithm, the threshold T corresponding to different frames is obtained adaptively by introducing the OTSU algorithm, and then the threshold T is linearly scaled by Equation 6 to obtain the most suitable threshold t for this study.

$$t = k \times T \tag{6}$$

where k represents the scaling factor. Figure 3 shows the two indicators corresponding to different k values, (a) shows the number of feature points extracted from the two images to be stitched, *imageL* and *imageR*, for different values of k. (b) displays the recall rate corresponding to different k values (Z. B. Wang & Yang, 2020). The recall rate is the ratio of the number of correct matches to the number of all matches, and the higher the value, the better the quality of the extracted feature points. As shown in Figure 3(a), the number of feature points that meet the criteria decreases as the value of k is greater than 0.25, the number of extracted feature points is less than 500, which is not conducive to the subsequent matching of feature points. Therefore, the range of k is considered to be $0 \sim 0.25$. Combined with Figure 3(b), when the value of k is smaller than 0.25, the highest

recall rate is achieved at 0.22. That means the quality of extracted feature points is the best when k is 0.22. As a result, in this paper, the k value is chosen to be 0.22.

$$recall \ rate = \frac{correct \ matches}{matches} \times 100\%$$
(7)

where *correct* matches represents the number of correct pairs, and matches represents the number of all matches.

Non-Extreme Value Suppression Yields Local Optimal Points

Many of the key points obtained through the above steps are subject to feature point aggregation, increasing the incidence of mismatches. The use of non-extreme value suppression has been experimentally shown to be effective in suppressing this phenomenon. The response value of each candidate key point is calculated by Equation 8 and then compared with the adjacent left and right points. If the response value of this key point is the largest, it is an extreme value and shall be retained. Otherwise, the point is discarded.

$$Q = \sum_{i}^{16} \left| V\left(P_{i}\right) - V\left(P\right) \right| \tag{8}$$

where Q is the sum of the absolute value of the difference between the point to be processed and the surrounding 16 pixels' grayscale values.

SIFT Operator to Generate Feature Descriptors

The key points are obtained by the improved adaptive FAST algorithm and then combined with the SIFT operator to generate feature descriptors. In other words, each key point is described using a feature vector. First, the main direction of the key point is determined. The histogram is used to count the gradient of the neighboring pixels, and the main peak value is selected as the main direction of this feature point. The values of m(x, y) and $\theta(x, y)$ are calculated as follows (Li et al., 2020):





$$m(x,y) = \left[\left(L(x+1,y) - L(x-1,y) \right)^2 + \left(L(x,y+1) - L(x,y-1) \right)^2 \right]^{1/2}$$
(9)

$$\theta(x,y) = \tan^{-1} \frac{L(x,y+1) - L(x-1,y)}{L(x+1,y) - L(x,y-1)}$$
(10)

where m(x,y) represents the module value of feature points, $\theta(x,y)$ represents the direction of feature points, and L(x,y) represents the space value of the key point.

The next step is to generate feature descriptors. As shown in Figure 4, an 8×8 window is selected. Each small cell indicates the pixels in the neighborhood of the feature point, the direction of the arrow indicates the direction of the pixel gradient, the length of the arrow indicates the magnitude of the pixel point, and the gradient direction of each pixel in the cumulative four large cells can form a seed point, as shown in the right panel of Figure 4. A feature point is composed of 2×2 seed points, with each containing information on 8 direction vectors. Therefore, a feature point can be represented by $2 \times 2 \times 8$, a total of 32-dimensional feature vectors. Lowe (2004) suggested representing descriptors using 128-dimensional feature vectors. Specifically, each feature point is composed of 4×4 seed points, with each seed point containing eight directional components.

Multi-Threaded Concurrency Optimization Acceleration

When a computer is equipped with a multi-core processor, the use of multi-threading techniques can simplify the programming structure, improve the utilization of the computer's central processing unit (CPU) resources, better use of the computer hardware advantages, and avoid the waste of hardware resources (Jung et al., 2022). By analyzing the structure of the algorithm in this paper, the feature extraction of two images by the multithreading technique is proposed. The specific steps are as follows:

- 1. Define the function task(image, kps, feature) to implement the feature extraction function.
- 2. Read the data of two images to be stitched, imageL and imageR.
- 3. Open two threads t1 = threading. Thread (target = task, args = (imageL, kpsL, featureL))

and t2 = threading. Thread (target = task, args = (imageR, kpsR, featureR)), so that the two images to be stitched can be extracted in parallel.

4. Store the extracted feature vectors in *Queue()*, and then read the stored feature vectors in *get()* to transfer the data between multiple threads.

Figure 4. Generation of feature descriptors by SIFT operator



5. The obtained imageL, kpsL, imageR, and kpsR data are transferred to the feature matching module for feature point matching.

FEATURE MATCHING AND IMAGE STITCHING

BF Algorithm for Feature Matching

The main idea of the BF algorithm is to try all possibilities by enumeration and return the optimal match, but it is prone to false matches when there are more feature points (Ince, 2022). In this paper, the author adds the cross-validation and near-next-neighbor ratio strategy to improve the matching accuracy based on the BF algorithm. The specific steps are as follows.

Calculating the Euclidean Distance

After obtaining the feature descriptors of the two images, the similarity between the feature points of the two images is described based on the Euclidean distance. The calculation formula is shown as follows:

$$d = \left\| M, N \right\| = \sqrt{\sum_{i=1}^{n} \left(M_{i} - N_{i} \right)^{2}}$$
(11)

where d is the Euclidean distance, M and N are the feature points in the two images, and n is the dimension of each feature point. The smaller the Euclidean distance d is, the more similar the pair of feature points are, and the more likely they are the same feature point.

Cross-Validation

By cross-verifying filter error matching pairs, we assume that M_i is a feature point in the first image and calculate the Euclidean distance between M_i and each feature point in the second image in turn. At this point, if the feature point corresponding to the minimum distance returned is N_i , use the feature point N_i to match with each feature point in the first image in turn, and if the feature point corresponding to the minimum distance returned is M_i , the cross-validation is passed. Consider M_i and N_i as a pair of candidate matches. Otherwise, they are discarded, until all matching pairs complete the cross-validation.

Near-Next-Neighbor Ratio

The candidate matching pairs are obtained by the above steps, and then the mismatching pairs are further filtered by the near-next-neighbor ratio strategy to improve the matching accuracy. Assuming that the returned nearest distance is d_1 and the next nearest distance is d_2 , the matching pairs with a higher possibility of mismatch are filtered by Equation 12.

$$\frac{d_1}{d_2} < ratio \tag{12}$$

where the larger the ratio of d_1 and d_2 is, the closer the nearest feature is to the next closest feature, and the more likely it is to be a false match. Conversely, the smaller the ratio is, the more distant the nearest feature is from the next closest feature, and the less likely it is to be a false match. *ratio* is a threshold value. If the value is too large, it will lead to an increase in the possibility of a false match. If this value is too small, it may result in a low number of matched pairs obtained. The value of *ratio* should be set reasonably. In this paper, the value is taken as 0.65.

Image Stitching

The prerequisite for image stitching is to obtain a highly accurate homography matrix H to avoid misalignment in the stitching process. Suppose the two images to be stitched are $\operatorname{Im} ageL$ and $\operatorname{Im} ageR$. Firstly, the authors use the RANSAC (Vasuhi et al., 2021) algorithm to obtain the homography matrix H, then perform the perspective transformation on $\operatorname{Im} ageR$, project the feature points in $\operatorname{Im} ageR$ to the space where $\operatorname{Im} ageL$ is located, and finally, perform seamless on-board image stitching by adaptive regions. The specific steps are as follows.

RANSAC Algorithm Calculates the Homography Matrix H

First, four pairs of sample data are randomly selected from the obtained matching pairs. These four pairs cannot be co-linear, and the homography matrix H_i between these four pairs of matching pairs is obtained by calculation, and the corresponding model D_i is obtained. The projection error between the remaining matched pairs and the model D_i is calculated, and if the difference is less than the empirical value, it is recorded as the inner point. So, the cycle continues until the specified number of iterations is reached, and the model with the most internal points is selected as the best matching model. Finally, the least square method is used to fit the best-matching model with the corresponding set of interior points to obtain the optimized homography matrix H.

Perspective Transformation

Assuming that Im ageR source pixel points are (u, v) and the pixel points after perspective transformation are (u', v'), the relationship can be expressed as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$
(13)

Through calculation:

$$\begin{cases} u' = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}} \\ v' = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}} \end{cases}$$
(14)

where h_{11} and h_{12} are rotation transformation parameters, h_{21} and h_{22} are scale transformation parameters, h_{13} and h_{23} are translation transformation parameters, h_{31} and h_{32} are perspective transformation parameters, and the value of h_{33} is 1.

Adaptive Region for Image Stitching

First, we calculate the overlapping region mask of $\operatorname{Im} ageL$ and $\operatorname{Im} ageR$ after perspective transformation, as shown in Figure 5. It can be seen that the feature points of $\operatorname{Im} ageR'$ in the region mask overlap with those of $\operatorname{Im} ageL$ when the homography matrix H reaches a certain degree of

accuracy. The experimental results show that by directly stitching the Im ageL and the non-overlapping areas of Im ageR', the vehicle stitching image without stitching gaps and misalignment can be obtained.

EXPERIMENTAL ANALYSIS

The configuration of the experimental environment in this paper consists of two computers: One is a Windows 10 x64 system with Prescan scenario simulation software installed for building autonomous driving scenarios and collecting vehicle image data. The other is a ubuntu 18.04 system with Python 3.6.9 and OpenCV 3.4.1, AMD Ryzen5 4600H CPU, NVIDIA GeForce GTX 1650 GPU, and 16G RAM for testing and validating the Prescan vehicle image stitching algorithm.

Data Collection

Prescan is used to build an autonomous driving scenario with a town as the background, and by changing the parameters of the weather in the scenario simulation software, the vehicle images are collected under four types of weather conditions: sunny, cloudy, rainy, and snowy. Figure 6 shows a set of data to be measured under different weather conditions, which is used to verify the robustness of each module of the algorithm proposed in this paper.

Performance Analysis of Feature Matching

To verify the advantages of this paper's algorithm in the feature matching module, it is compared with the traditional SIFT and SURF algorithms in terms of the number of extracted feature points, recall rate, correct matching rate, and matching time under different weather conditions. The correct

Figure 5. Overlapping area



Figure 6.

Image data to be tested: (a) sunny, (b) cloudy, (c) rainy, (d) snow





10

matching rate is the ratio of the number of correct matching pairs to the number of final matching pairs, and the higher the value, the higher the efficiency of matching. The calculation formula is as follows:

correct matching rate =
$$\frac{correct \ matches}{matches} \times 100\%$$
 (15)

where *correct* matching rate is the correct match rate, *correct* matches is the number of correct matches, and *matches*' is the number of matches left after filtering out mismatches.

Figures 7–9 show the visual effect plots of SIFT, SURF, and the algorithm of this paper for feature matching under different weather conditions, respectively. From Figures 7–8, it can be seen that the traditional SIFT and SURF algorithms show significant mismatching pairs when performing feature matching, and from Figure 9, it can be seen that the algorithm proposed in this paper does not have significant mismatching occurrence, although it obtains fewer matching pairs.

Table 1 shows the indicator data of each algorithm in each weather condition. From the number of feature points extracted on sunny and cloudy days, it can be seen that the traditional SIFT and SURF algorithms are susceptible to the influence of illumination. In this paper, the algorithm uses

Figure 7. Feature matching result of SIFT algorithm: (a) sunny, (b) cloudy, (c) rainy, (d) snow







International Journal of Information Technologies and Systems Approach Volume 16 • Issue 3

Figure 9.

Feature matching result of proposed algorithm: (a) sunny, (b) cloudy, (c) rainy, (d) snow



the improved adaptive FAST algorithm for feature point detection, which improves the number of extracted feature points and effectively overcomes the influence of illumination on feature point extraction. In rainy and snowy weather conditions, the trailing phenomenon caused by the vehicle driving at a high speed is more serious, resulting in fewer feature points identified, but compared with the traditional SIFT and SURF algorithms, the proposed algorithm in this paper still has the best recall rate, which is improved by 34.52% and 51.92% to about 30.64%, indicating that the proposed algorithm has effectively improved the quality of extracted feature points.

Weather	Parameters	SIFT	SURF	Proposed
Sunny	feature number (left/right)	1040/1454	3193/4580	410/576
	recall rate (%)	19.19	17.29	27.60
	correct matching rate (%)	94.26	96.94	98.76
	time (ms)	267.86	567.95	26.90
Cloudy	feature number (left/right)	459/732	1870/2685	448/698
	recall rate (%)	21.17	20.37	23.64
	correct matching rate (%)	98.10	96.47	98.99
	time (ms)	232.97	465.88	27.43
Rainy	feature number (left/right)	753/975	2623/3670	256/296
	recall rate (%)	22.26	18.31	35.81
	correct matching rate (%)	95.18	97.25	99.07
	time (ms)	247.02	524.49	17.81
Snow	feature number (left/right)	534/625	1698/2065	167/200
	recall rate (%)	28.48	24.70	35.50
	correct matching rate (%)	96.22	97.14	98.61
	time (ms)	237.10	433.82	15.10

Table 1. Feature matching results of each algorithm

It can be seen from the correct matching rate in each weather condition that the performance of the algorithm in this paper is optimal in all cases. Compared with the traditional SIFT and SURF algorithms, the average correct matching rate is optimized by 3.04% and 1.97%, reaching about 98.86%, indicating that the algorithm has effectively improved the matching accuracy of the feature matching module. The performance of the algorithm in this paper is also optimal in all the weather conditions, as can be seen from the matching elapsed time in each weather condition. Compared with the traditional SIFT and SURF algorithms, the average reduction is 91.14% and 95.62%, which is about 21.81 ms. It shows that the algorithm matching elapsed time is significantly reduced by the optimization of the algorithm and the multi-threaded structure.

Image Stitching Performance Analysis

To verify the advantages of the algorithm in the image stitching module of this paper, it is compared with the traditional SIFT and SURF algorithms in terms of the visual effect of stitching and the time of stitching, Figures 10–12 show the visual effect of stitching for each algorithm. Figures 10–11 show that the image stitching results of the traditional SIFT and SURF algorithms have obvious stitching gaps and color differences. Figure 12 shows that by analyzing the conversion relationship between the images to be stitched and using the proposed adaptive stitching strategy, the above problems are effectively solved, and a vehicle stitched image without obvious stitching gaps and misalignment is obtained.

Figure 10. Image stitching results of SIFT algorithm: (a) sunny, (b) cloudy, (c) rainy, (d) snow



Figure 11. Image stitching results of SURF algorithm: (a) sunny, (b) cloudy, (c) rainy, (d) snow



International Journal of Information Technologies and Systems Approach Volume 16 • Issue 3

Figure 12.

Image stitching results of proposed algorithm: (a) sunny, (b) cloudy, (c) rainy, (d) snow



Figure 13 shows the running time of each algorithm when image stitching is carried out under different weather conditions. The algorithm in this paper has the shortest average time (about 22.82 ms) for vehicle image stitching in each weather condition, which is optimized by about 93.16% and 96.04% compared with the traditional SIFT and SURF algorithms. In summary, the algorithm in this paper can complete the vehicle image stitching task in real time and accurately in all weather conditions with good robustness.





CONCLUSION AND OUTLOOK

The proposed adaptive real-time image stitching algorithm was adopted to solve the problems of single application scenes, low matching accuracy, poor real-time performance, and obvious stitching gaps of traditional image stitching technology in Prescan test scenes. Specific improvements are as follows. First, the OTSU algorithm is introduced to adaptively represent the grayscale features of different images by analyzing the application scenes, which overcomes the problem of thresholds fixation in vehicle images with different brightness and different textures and improves the adaptability in different scenes. Then, the combination of the improved FAST algorithm with SIFT descriptors for feature extraction in parallel provides a low-complexity and efficient feature extraction algorithm. After that, the cross-validation and the near-next-neighbor ratio strategy are used to filter out the mismatched pairs to improve the matching accuracy. Finally, the adaptive region stitching strategy is used to generate vehicle images without stitching gaps. The results show that compared with the traditional SIFT and SURF algorithms, the performance is optimal under different weather conditions. A low-complexity and high-accuracy image stitching technique with good robustness is provided for laboratory test environments and other virtual scenarios. Currently, only two vehicle images have been stitched, and in future work, the panoramic stitching technique for three and more vehicle images will be investigated.

DATA AVAILABILITY

The data used to support the findings of this study are included within the article.

CONFLICTS OF INTEREST

The author declares that there is no conflict of interest regarding the publication of this paper.

FUNDING STATEMENT

This work was partially supported by the National Natural Science Foundation of China (62271303), the Natural Science Foundation of Shanghai (20ZR1423200), the Innovation Program of Shanghai Municipal Education Commission of China (2021-01-07-00-10-E00121), the Shanghai Sailing Program No. 20YF1416700.

REFERENCES

Hu, Y., Chen, Q., Feng, S. J., Tao, T. Y., Asundi, A., & Zuo, C. (2019). A new microscopic telecentric stereo vision system-Calibration, rectification, and three-dimensional reconstruction. *Optics and Lasers in Engineering*, *113*(4), 14–22. doi:10.1016/j.optlaseng.2018.09.011

Ince, I. F. (2022). Robust image matching for information systems using randomly uniform distributed SURF features. In Applications of Computational Science in Artificial Intelligence. IGI Global.

Jiao, S. M., Zhong, Y. F., Ding, H., Yao, X., & Bai, J. P. (2022). Research on an environmental adaptive feature extraction algorithm. *Measurement Science & Technology*, *34*(2), 025405. doi:10.1088/1361-6501/aca173

Jung, M., Kim, D. Y., & Kim, S. (2022). A system architecture of a fusion system for multiple LiDARs image processing. *Applied Sciences (Basel, Switzerland)*, *12*(19), 9421. doi:10.3390/app12199421

Li, Y. L., Huang, J. Q., Deng, F. M., Lu, R. H., & Yao, M. (2020). An image stitching algorithm based on SIFT and DAISY descriptor. *Proceedings of the Annual International Conference on Data Science and Business Analytics*, 271–274. doi:10.1109/ICDSBA51020.2020.00077

Liu, Y., Tian, J. W., Hu, R. R., Yang, B., Liu, S., Yin, L. R., & Zheng, W. F. (2022). Improved feature point pair purification algorithm based on SIFT during endoscope image stitching. *Frontiers in Neurorobotics*, *16*, 840594. doi:10.3389/fnbot.2022.840594 PMID:35242022

Liu, Y. B., Qin, J. H., Zhu, M. Y., & Huang, T. T. (2022). Fast stitching method for multi-view images of cupping spots. *Signal, Image and Video Processing*. Advance online publication. doi:10.1007/s11760-022-02402-0

Liu, Y. Y., He, M., Wang, Y. Y., Sun, Y., & Gao, X. B. (2022). Farmland aerial images fast-stitching method and application based on improved SIFT algorithm. *IEEE Access : Practical Innovations, Open Solutions, 10*, 95411–95424. doi:10.1109/ACCESS.2022.3204657

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60), 91–110. doi:10.1023/B:VISI.0000029664.99615.94

Miao, H. S., Xiao, C. Y., Li, Y. C., Zou, S., & Huang, W. (2020). Machine vision system for key-cap flatness measurement of computer keyboards. *Optical Engineering (Redondo Beach, Calif.)*, 59(3), 033107. doi:10.1117/1. OE.59.3.033107

Pham, N. T., Park, S., & Park, C. S. (2021). Fast and efficient method for large-scale aerial image stitching. *IEEE Access : Practical Innovations, Open Solutions*, 9, 127852–127865. doi:10.1109/ACCESS.2021.3111203

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. *Proceedings of the Ninth European Conference on Computer Vision*, 430–443.

Vasuhi, S., Samydurai, A., & Vijayakumar, M. (2021). Multicamera video stitching for multiple human tracking. *International Journal of Computer Vision and Image Processing*, *11*(1), 17–38. doi:10.4018/IJCVIP.2021010102

Wang, C. S., Liu, D. Y., & Hsu, K. S. (2021). Simulation and application of cooperative driving sense systems using Prescan software. *Microsystem Technologies*, 27(4), 1201–1210. doi:10.1007/s00542-018-4164-z

Wang, Z. B., & Yang, Z. K. (2020). Review on image-stitching techniques. *Multimedia Systems*, 26(4), 413–430. doi:10.1007/s00530-020-00651-y

Yan, S., & Ma, M. X. (2022). Image stitching based on an Improved SIFT Algorithm. *Proceedings of the Seventh International Conference on Electronic Technology and Information Science*, 1–5.

Yang, Q. N., Qiu, C. G., Wu, L. T., & Chen, J. J. (2021). Image matching algorithm based on improved FAST and RANSAC. *Proceedings of the 2021 IEEE International Conference on Mechatronics and Automation*, 142–147. doi:10.1109/ICMA52036.2021.9512798

Zhang, G. X., Qin, D. Y., Yang, J. Q., Yan, M. Y., Tang, H. P., Bie, H. Z., & Ma, L. (2022). UAV low-altitude aerial image stitching based on semantic segmentation and ORB algorithm for urban traffic. *Remote Sensing*, *14*(23), 6013. doi:10.3390/rs14236013

Zhang, H. P., Leng, C. C., Yan, X., Cai, G. R., Pei, Z., Yu, N., & Basu, A. (2022). Remote sensing image registration based on local affine constraint with circle descriptor. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1–5. doi:10.1109/LGRS.2020.3027096

Zhang, Y. J., & Xie, Y. Q. (2021). Adaptive clustering feature matching algorithm based on SIFT and RANSA. *Proceedings of the Second International Conference on Electronics, Communications and Information Technology*, 174–179.

Zhang, Y. L., Lai, Y. K., & Zhang, F. L. (2022). Content preserving image stitching with piecewise rectangular boundary constraints. *IEEE Transactions on Visualization and Computer Graphics*, 27(7), 3198–3212. doi:10.1109/TVCG.2020.2965097 PMID:31944958