IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 22, 2019, accepted September 4, 2019, date of publication September 9, 2019, date of current version September 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940137

Multi-Threshold Corner Detection and Region Matching Algorithm Based on Texture Classification

ZETIAN TANG[1], ZHAO DING[1], RUIMIN ZENG[1], YANG WANG[1], JUN WEN[2], LIFENG BIAN[3], AND CHEN YANG[1]

[1]Guizhou Key Laboratory of Micro-Nano-Electronics and Software Technology, Engineering Center of the Ministry of Education of Semiconductor Power Device Reliability, College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China
[2]School of Computer and Electrical Information, Guangxi University, Nanning 530004, China
[3]Key Laboratory of Nanodevices and Applications, Suzhou Institute of Nano-Tech and Nano-Bionics, Chinese Academy of Sciences, Suzhou 215123, China

Corresponding author: Chen Yang (eliot.c.yang@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61604046, in part by the Guizhou Science and Technology Plan Project under Grant [2017]5788 and Grant [2018]5781, in part by the Engineering Research Center Open Fund, Semiconductor Power Device Reliability, Ministry of Education under Grant 20176103, and in part by the fund of the Key Laboratory of Nanodevices and Applications, Suzhou Institute of Nano-Tech and Nano-Bionics under Grant 18ZS08.

ABSTRACT In order to address the unreasonable distributed corners in single threshold Harris detection and expensive computation cost incurred from image region matching performed by normalized cross correlation (*NCC*) algorithm, multi-threshold corner detection and region matching algorithm based on texture classification are proposed. Firstly, the input image is split into sub-blocks which are classified into four different categories based on the specific texture: flat, weak, middle texture and strong regions. Subsequently, an algorithm is suggested to decide threshold values for different texture type, and interval calculation for the sub-blocks is performed to improve operation efficiency in the algorithm implementation. Finally, based on different texture characteristics, Census, interval-sampled *NCC*, and complete *NCC* are employed to perform image matching. As demonstrated by the experimental results, corner detection based on texture classification is capable to obtain a reasonable corner number as well as a more uniform spatial distribution, when compared to the traditional Harris algorithm. If combined with the interval classification, speedup for texture classification is approximately 30%. In addition, the matching algorithm based on texture classification is capable to improve the speed of 26.9%∼29.9% while maintaining the comparable accuracy of *NCC*. In general, for better splicing quality, the overall stitching speed is increased by 14.1%∼18.4%. Alternatively, for faster speed consideration, the weak texture region which accounts for a large proportion of an image and provides less effective information can be ignored, for which 23.9%∼28.4% speedup can be achieved at the cost of a 1.9%∼3.9% reduction in corner points. Therefore, the proposed algorithm is made potentially suited to uniformly distributed corner point calculation and high computation efficiency requirement scenarios.

INDEX TERMS Harris, texture classification, interval categorization, classification matching.

I. INTRODUCTION

Image stitching represents a process of transforming partial views into larger-scale views by features such as corner detection, registration, fusion and the likes. It is regarded as a significant part of image processing and is extensively used in motion detection, resolution enhancement, remote sensing and medical imaging [1].

At present, a majority of the stitching algorithms can be classified into two groups: gray scale-based and feature-based approaches, where the feature-based registration one is preferred by researchers [2]–[4] for its affine invariance, stability and excellent robustness. Corner points which provide maximum curvature in micro regions of an image are usually taken as a comprehensive object descriptor, and the Harris algorithm is commonly applied as prior step to provide corner information as guidance for the following steps for image understanding [5], [6]. Nevertheless, the number of

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou.

Harris corner points derived from images varies significantly depending on the threshold value setting, and the value needs to be adjusted for different images [7]. To eliminate the need for manually adjustment, a variety of different adaptive threshold setting algorithms have been developed in the literature over the past decade. Li *et al.* [8] proposed an adaptive threshold factor 'ρ' to adjust it to a reasonable value and the Forstner operator is involved to identify the best feature point. Cui *et al.* [9] suggested a Harris corner detection algorithm based on Barron operator which is used to calculate the image gradient, then centre B-spline function is applied to smooth the image, and finally non-maximum inhibition and corner sieving are performed to determine the real corner points. The algorithm demonstrates strong anti-noise ability and is effective in the extraction of corner points. An adaptive corner detection algorithm based on iterative threshold calculation algorithm is put forward by Wang *et al.* [10] to avoid cluster and pseudo corner, based on which desirable results have been achieved in threshold setting and feature extraction. Shen *et al.* [11] came up with a method to split an image into several independent blocks and adopted an iterative method to determine the appropriate threshold for each block. Changan and Chilveri [12] proposed a Harris corner detection algorithm for stereo image feature matching and a threshold operator has been raised to obtain the upper and lower threshold value.

Nevertheless, the detection with a single threshold usually leads to an unreasonable distribution of corner points and iterative searching incurs a substantial amount of computation cost. In order to address this problem, an adaptive multi-threshold calculation approach based on image texture complexity analysis and classification is proposed to generate more uniformly distributed corner points. Firstly, images can be segmented into four types of regions based on texture complexity for all subsequent calculations. Secondly, the approach to calculating different threshold value is proposed for each texture region. Subsequently, Census algorithm, interval sampling *NCC* and complete *NCC* algorithm are employed in matching process for different regions rather a single *NCC* for computation efficiency. Finally, fine matching is performed with the assistance of RANSAC, and image fusion is performed for image stitching. The experimental results indicated in section III demonstrates corner point distribution and computation performance by the algorithm proposed in the paper.

II. FUNDAMENTAL

A. SINGLE THRESHOLD VALUE

Harris corner detection algorithm demonstrates the desirable characteristics of high speed and high precision. Nevertheless, its threshold setting could make a direct impact on detection effect. To obtain corner points, single threshold value is usually employed and compared to the corner response function (*CRF*), which is indicated in equation (1):

$$CRF = det(M) - k(trace(M))^2 \tag{1}$$

where, $det(M)$ represents the determinant of the matrix $M(x, y)$, $trace(M)$ indicates the trace of the matrix, k denotes the empirical constant which ranges from 0.04 to 0.06, and $M(x, y)$ refers to the autocorrelation matrix of the pixel (x, y), as shown in equation (2):

$$M(x, y) = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \tag{2}$$

$$w(x, y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{3}$$

where: $w(x, y)$ indicates a Gaussian window factor, as shown in formula (3). I_x and I_y represent the derivatives in the horizontal and vertical directions of the point in the image, respectively.

However, the texture complexity varies from region to region in an image. If a single threshold is employed, it has a great possibility that the corner points are subjected to suppression in the simple texture region and are made excessive in the complex textures region, which tends to result in unbalanced distributed corner points. With a low threshold value, it is prone to the generation of pseudo corner points and corner points cluster (as shown in Figure 1(a)). Otherwise, a high threshold value will lead to much sparser distributed corner points and makes it inadequate to describe the image features (as shown in Figure 1(b)).

B. MULTI-THRESHOLD STITCHING BASED ON TEXTURE CLASSIFICATION

Herein, a dynamically multi-threshold calculation algorithm is proposed base on texture complexity to prevent the adverse consequences resulting from the mismatch between threshold and texture complexity. In the subsequent image stitching process, the corresponding algorithms are applied for different textures to improve the speed. To analyze the crucial factor of texture in the algorithm, the input image is segmented into multiple sub-blocks. The selection of the sub-block size is detailed in Section III.A based on experiment and 5 × 5 pixel block is employed for the following discussion.

In each sub-block, the texture complexity can be described by number 'V' according to the gray histogram peaks. The larger V value, the richer the image texture represented by the sub-block. Otherwise, the flatter the texture is.

In this paper, four types of texture regions are involved for multiple threshold value setting and subsequent matching processing:

Flat Region ($V = 1$): there is only a single gray value for the sub-block and no texture variation. This type of region is incapable to provide any effective information for corner detection. Thus, no corner points detection processing is required for time saving.

Weak Texture Region ($1 < V \leq 5$): texture variation is extremely small which can provide less information on corner points. Therefore, a smaller threshold is conducive to increasing the number of corner points.

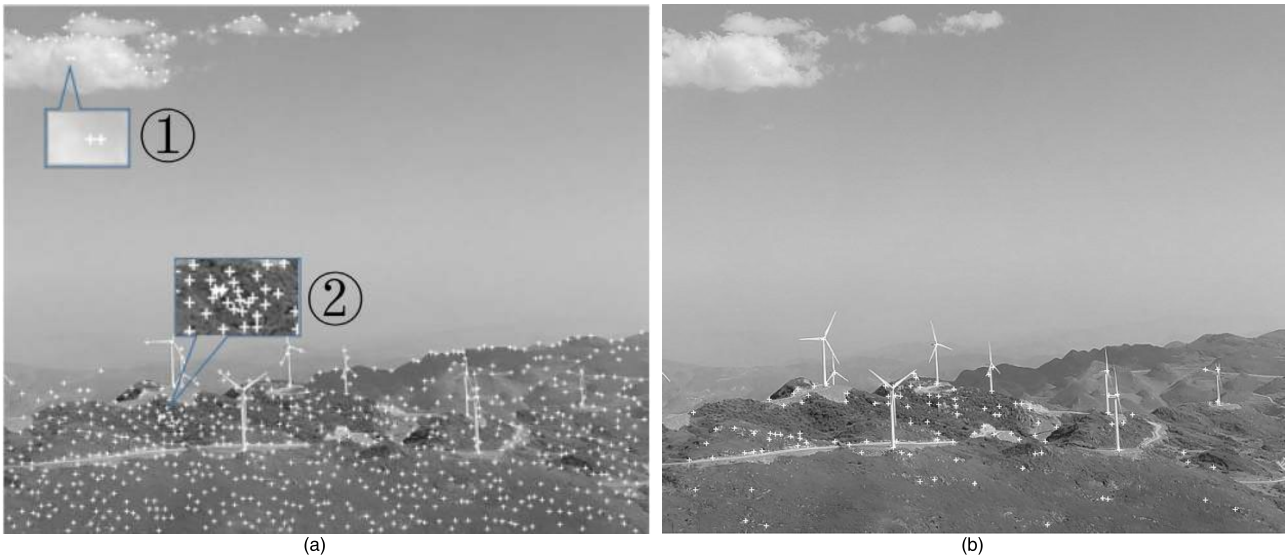


FIGURE 1. Impact on corner detection by threshold value: (a) Small threshold; (b) Excessive threshold. ① : Pseudo corner points; ② : Corner clusters.

Middle Texture Region ($5 < V \leq 15$): the texture variation is noticeable, and the effective corner points have a higher likelihood of being detected with a medium threshold value.

Strong texture region ($V > 15$): the texture varies significantly with a high probability to provide points information; however, a larger threshold is deemed necessary to prevent corner points clustering.

After the Harris corner detection, corner points matching is required, in which *NCC* is one of the most commonly used algorithms, and the correlation coefficient is calculated to estimate similarity of the corner points as shown in formula (4):

$$NCC = \frac{\sum_i \left[I_1(x_i, y_i) - \bar{I}_1\right]\left[I_2(x_i, y_i) - \bar{I}_2\right]}{\sqrt{\sum_i \left[I_1(x_i, y_i) - \bar{I}_1\right]^2}\sqrt{\sum_i \left[I_2(x_i, y_i) - \bar{I}_2\right]^2}} \tag{4}$$

where: I_1 and I_2 represent the associated windows of the corner points of two images, respectively. $\bar{I}_1$ and $\bar{I}_2$ denote their average gray value.

The *NCC* algorithm exhibits various advantages such as strong anti-noise ability and high precision. Despite this, it requires heavy computation [13], [14]. In order to achieve an optimal calculation efficiency in the corner points matching stage, different matching algorithms are applied to different texture region based on corresponding characteristics:

Flat Region: no matching processing is necessary.

Weak Texture Region: Census algorithm is classed as a non-parametric transform matching method [15], which is capable of excellent performance in large image noise and illumination variation scenarios, which makes it suitable for the weak texture region. In weak texture region, corner point distribution is sparse and beneficial to avoid repetitive or similar textures which are sensitive in Census algorithm [16]. The overall flow for the algorithm is as follows: 1) A 0/1 sequence is generated for a corner point window, where 0/1 represents a given point that is large or less than the central pixel respectively. 2) Similarity for corner points is determined by Hamming distance of the sequence.

Middle Texture Region: for large matching window, it is unnecessary for all pixels to be involved in *NCC* operation [17], [18]. In order to expedite the calculation, the middle texture region takes the same interval sampling approach, by which the black dots is the pixels participating in the calculation and the white dots is discounted as shown in Figure 2.

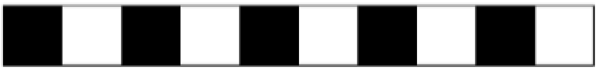


FIGURE 2. Interval sample method.

Strong Texture Region: The texture information of this region is complex, and each pixel has a significant influence on the similarity calculation, which makes the complete *NCC* operation imperative.

To avoid mismatching, the Random Sample Consensus (RANSAC) is adopted to perform fine matching, and the projection transformation matrix between two images is calculated. Finally, the image is stitched by image fusion.

C. ALGORITHM FLOW

The algorithm flow is illustrated in Figure 3, which consists of the four stages: preprocessing to segment image sub-block for following steps, interval texture classification, multi-threshold corner point calculation, corner points matching and image fusion based on texture classification. The multi-threshold corner detection algorithm mainly focuses on the improvement made to the second and third stage.

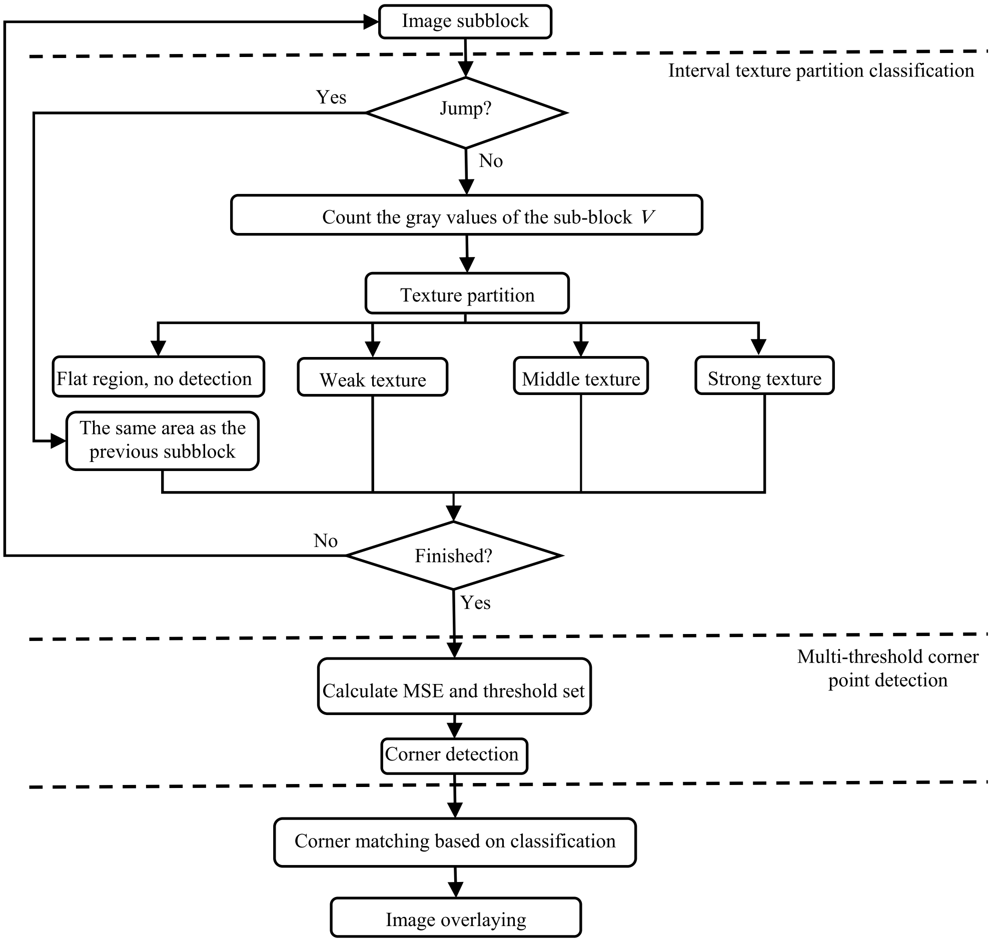


Image subblock
Interval texture partition classification
Yes
Jump?
No
Count the gray values of the sub-block V
Texture partition
Flat region, no detection
Weak texture
Middle texture
Strong texture
The same area as the previous subblock
No
Finished?
Yes
Multi-threshold corner point detection
Calculate MSE and threshold set
Corner detection
Corner matching based on classification
Image overlaying

FIGURE 3. Harris corner detection flow chart based on image classification.

1) INTERVAL TEXTURE CLASSIFICATION

As the Harris corner points detection are closely associated with the texture complexity, which is proposed to determine the multi-threshold value and the following steps. Traditionally, texture information is usually provided by descriptors for material classification, object recognition and natural scene identification applications [19]–[22].

As the number of gray levels in a sub-block is closely related to the texture gradient of a local region, texture classification based on the gray histogram for each sub-block is a significant factor for the following flows. Though block-by-block calculation provides accurate complexity information about each sub-block, the amount of calculation required is relatively large.

For images in practice, it has higher possibility for two adjacent sub-blocks classified into the same texture region. Taking advantage of this feature, an alternating interval calculation approach is proposed for texture classification as shown in Figure 4.

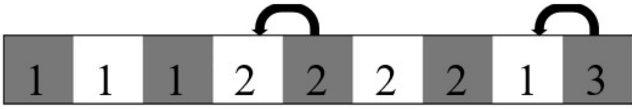


FIGURE 4. Interval texture classification.

In the scan for texture classification, the calculation pointer jump to the interval sub-block unit after calculating a sub-block (as shown in the dark area of Figure 4). Then, a judgment is made as to whether two adjacent calculated dark sub-blocks belong to the same texture type. If yes, the intermediate white sub-blocks have a higher probability of falling within the same texture region with its left and right adjacent sub-blocks, and the texture characteristics are directly assigned. Otherwise, there is a necessity to return and calculate the texture complexity for the white block.

The process will continue to scan the entire image until all sub-blocks are classified and marked as either flat, weak, middle or strong texture regions.

2) MULTI-THRESHOLD CORNER POINTS DETECTION

In the stage, multi-threshold values are set in the order of strong, middle and weak texture region.

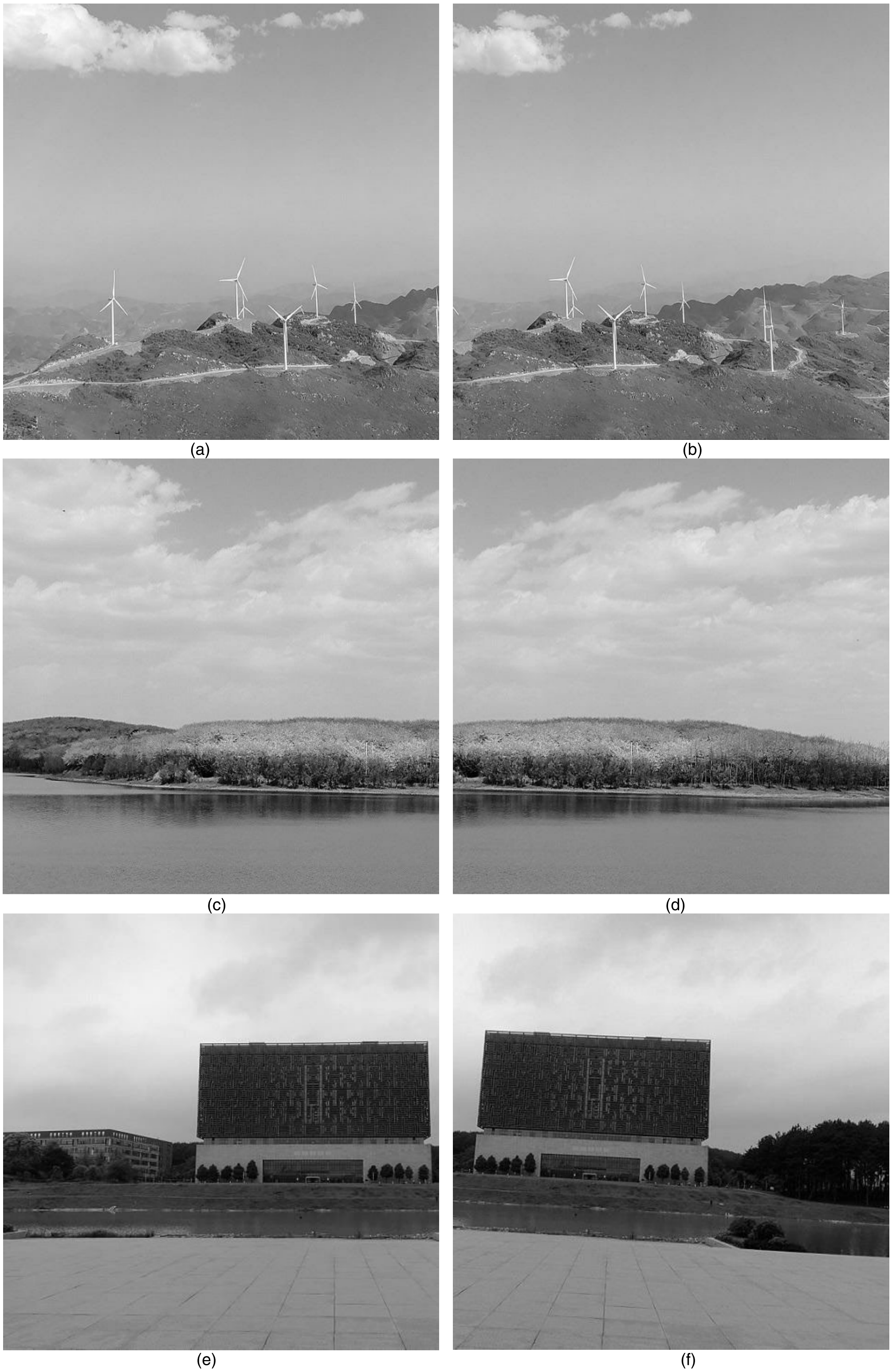


FIGURE 5. The original images: (a) and (b) Image of the mountain; (c) and (d) Image of the island; (e) and (f) Image of the building.

Strong Texture Region: the region represents strongest texture variation and it is necessary to impose limit on the maximum corner point number to avoid cluster. Firstly, a smaller threshold T_0 is adopted to generate excessive corner points. Secondly, the Nth largest *CRF* is taken as threshold for the region to reduce corner point number, where N is

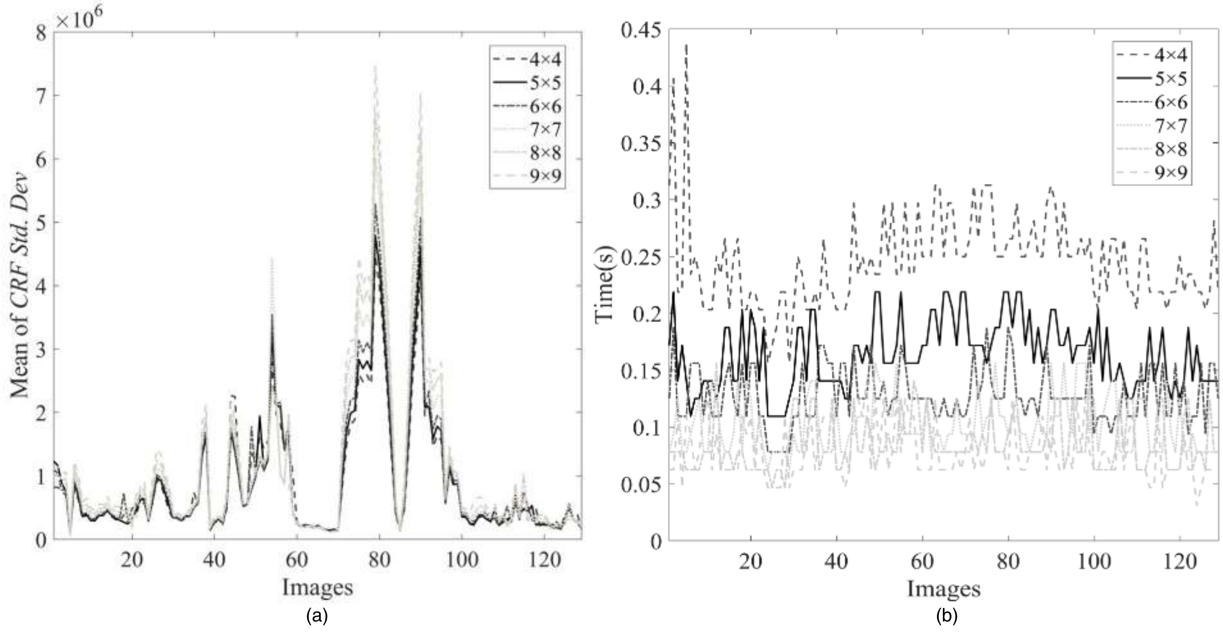


FIGURE 6. Impact of Sub-block on: (a) the mean of CRF of Std. Dev in four regions; (b) classification time.

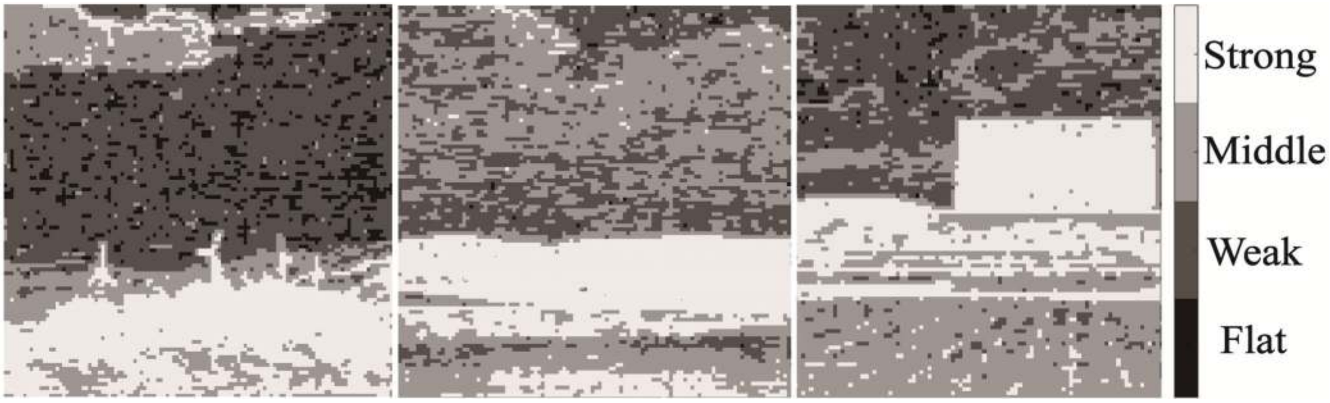


FIGURE 7. Results for Interval texture classification.

parameter in relation to the strong region area defined by linear equation (5):

$$n = aS + b \tag{5}$$

where, S represents the total area of the strong regions. In addition, coefficients a, b are determined by experiments conducted on different scenes, and the recommended values are 0.0053 and 43.12 respectively.

Middle and Weak Texture Region: the average mean square error (MSE) (eq. (6)) is employed to determine threshold value for the two regions.

$$D = \frac{1}{k}\sum_{s=1}^{k}\sqrt{\frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n}(I_{i,j} - \bar{I})^2} \tag{6}$$

where k represents the number of sub-blocks, and $m \times n$ indicates the sub-block size, $I_{i,j}$ denotes the gray value of each pixel in the sub-block, and $\bar{I}$ refers to the average value of the sub-block gray values.

Eq. (7) is the empirical equation to decide the thresholds.

$$T_n = \begin{cases} 0.1 \times T_u & 0 < D_n \le \dfrac{D_u}{3} \\ 0.4 \times T_u & \dfrac{D_u}{3} < D_n \le \dfrac{2 \times D_u}{3} \\ T_u & \dfrac{2 \times D_u}{3} < D_n \le D_u \end{cases} \tag{7}$$

where: T_n and D_n represent the threshold and average MSE for the two regions, respectively. T_u and D_u denote the parameters of strong texture regions if the equation is applied to calculate threshold for middle regions, or the parameters of middle texture regions to calculate the threshold for weak regions.

Multi-threshold corner detection can be effective in increasing the corner points of the weak texture area, maintaining the corner points of the middle texture area, and reducing the corner points of the strong texture area to avoid clustering, which contributes to a more uniform distribution of the corner points.

3) IMAGE REGISTRATION AND FUSION

In the next stage, corner points are registered and performed by registration algorithm, precision matching was performed by RANSAC method, and finer adjustment was made to the weighted smoothing algorithm [23] to achieve smooth transitions between the two images.

III. EXPERIMENTAL RESULTS AND ANALYSIS

Figure 5 presents an original image used in three sets of stitching, wherein (a) and (b) are 515 pixels × 607 pixels; (c) and (d) are 551 pixels × 521 pixels; (e) and (f) are 478 pixels × 467pixel.

A. SUB-BLOCK SIZE ANALYSIS

The choice of sub-block size must give consideration to both the detected corner point distribution and computational efficiency.

In respect of corner point distribution, as the *CRF* value is indicative of the local texture complexity, the sub-block size selection is expected to reduce the fluctuations, which are assessed by using the mean of the *CRF* standard deviation (*Std. Dev*) in each region. Figure 6(a) indicates the average *Std. Dev* variation for images in panorama dataset [24]. It can be found out that a larger block size results in a large average *Std. Dev.*, which suggests a large fluctuation in *CRF*.

In terms of computational efficiency as shown in Figure 6(b), a large block size is conducive to lowering the time cost of texture classification.

Therefore, it is recommended to choose a 5 × 5 or 6 × 6 sub-block size. In this paper, 5 × 5 sub-blocks are employed for discussion simplification.

B. RESULTS FOR INTERVAL TEXTURE CLASSIFICATION

Taking Figure 5 (a), (c), (e) as an example, the results for the interval texture classification are indicated in Figure 7. It can be seen that the strong texture region represents the strongest variation in an image, while the flat one fails to provide any corner points information.

As mentioned above, an alternative calculation strategy is applied to perform sub-block classification calculation. Figure 8 shows the time enhancement and classification accuracy rate when the jump step is set to 1 and 2 as compared to the non-jump calculation. It can be seen that the interval calculation approach is capable to provide about 30% time saving compared to the non-jumping one. Nevertheless, not obvious further improvement is made if the jumping step is greater than 1. Besides, a large jumping step would cause the accuracy to decline. Therefore, it is recommended to apply interval calculation method with step = 1, with both efficiency and accuracy taken into account.

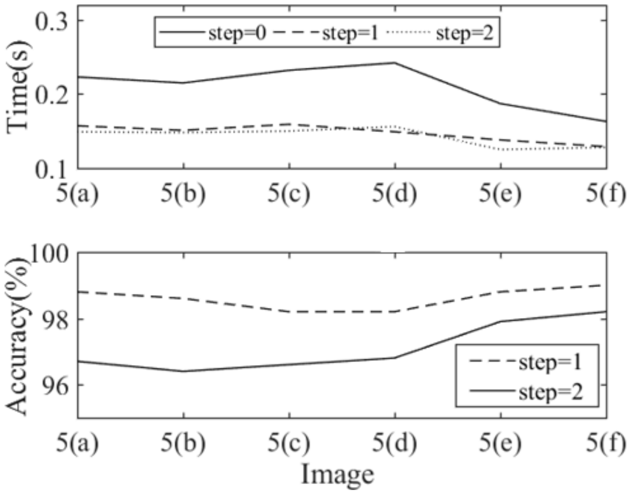


FIGURE 8. Time and classification accuracy statistics chart.

C. CORNER POINTS DISTRIBUTION

Figure 9 illustrates corner point distribution calculated by the proposed (Figure 9(c), (f), (i)), the traditional Harris algorithm (Figure 9(a), (d), (g)) and Cui's algorithm [9] (Figure 9(b), (e), (h)). For the traditional Harris and Cui's algorithms, corner points show a tendency to concentrate at the complex texture regions, while simple texture regions contain a smaller number of corner points. By reducing corner points at the complex texture regions and increasing those at the simple texture regions, the proposed algorithm is capable to balance distribution of corner points. As a result, the algorithm can provide a reasonable number of corner points, and the distribution is made more uniform and reasonable, which is conducive to improving the quality of image stitching.

Figure 10(a) illustrates the proportion of different texture regions calculated of images listed in Figure 5 based the algorithm in this paper. It can be seen that the proportion of flat regions incapable to provide useful information is low, while the strong texture regions which represent largest variation account for about 20% to 30% of an image. Besides, a major part of the image can be classified into the weak and middle texture regions.

Furthermore, different regions make different contribution to corner points Figure 10(b). Though it accounts for about 20% to 30% of an image, the strong texture regions provide over 70% of the corner information. The weak texture regions which generally occupy a large proportion of the image can only provide less corner points information, for which calculation for the weak regions could be omitted for high speed applications.

D. TIME CONSUMPTION

Following detection, the corner points need to be matched. Multiple matching algorithms premised on the texture

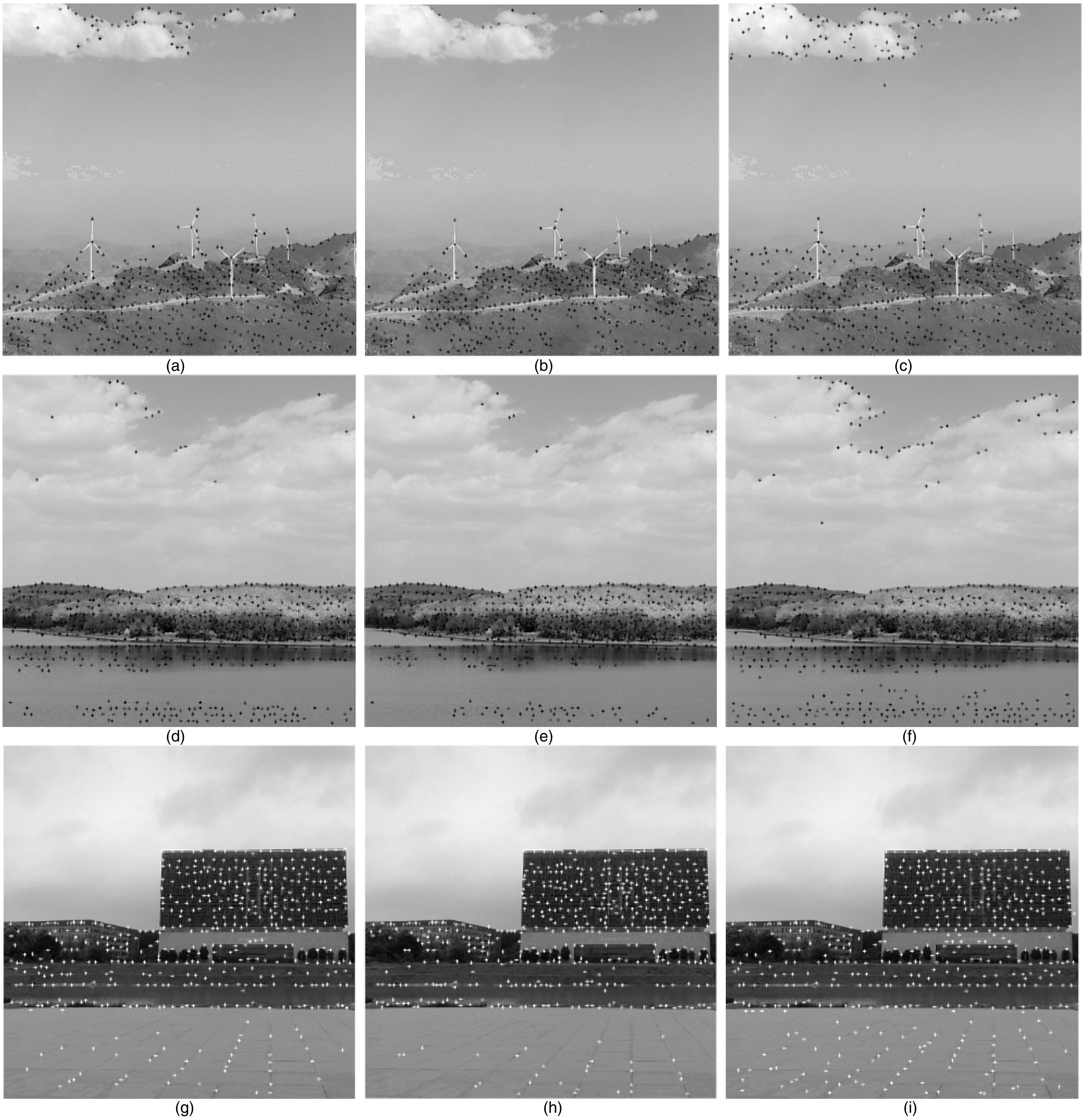


FIGURE 9. Corner detection results: (a), (d) and (g): Traditional Harris algorithm; (b), (e) and (h): Cui's algorithm; (c), (f) and (i): The proposed algorithm.

TABLE 1. Comparison between the multiple matching algorithms based on texture and *NCC*.

Image	Corner matching based on classification			*NCC*		
	Number of corner	Time(s)	RANSAC	Number of corner	Time(s)	RANSAC
5(a),(b)	197	2.556	170	207	3.568	178
5(c),(d)	123	2.582	83	124	3.687	85
5(e),(f)	66	2.119	25	64	2.901	26

classification are compared against the single *NCC*, as listed in Table 1. In terms of the number of the matched corner points, the multiple algorithms show similarity to *NCC*. Besides, the computation time for the proposed algorithm is reduced as it only matches corner points belonging to the same texture regions rather than the whole image.

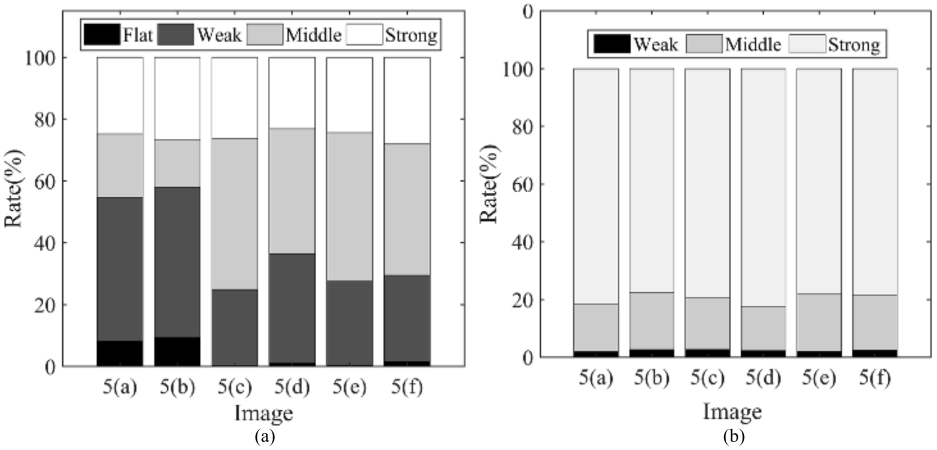


FIGURE 10. Proportion of: (a) Different texture region; (b) Corner point contribution.

TABLE 2. Times comparison table of each stage.

algorithm	Image	Corner number	Reading image (s)	Texture classification (s)	Corner detection (s)	Matching (s)	Fusion (s)	Overall (s)
Harris+*NCC*	5(a),(b)	493,532	0.044	0	0.603	3.579	0.644	4.870
	5(c),(d)	551,472	0.046	0	0.492	3.619	0.525	4.682
	5(e),(f)	466,443	0.046	0	0.459	2.884	0.495	3.884
Cui's	5(a),(b)	557,468	0.044	0	0.670	3.633	0.575	4.922
	5(c),(d)	526,512	0.046	0	0.617	3.637	0.557	4.857
	5(e),(f)	452,458	0.046	0	0.527	2.877	0.484	3.934
Detect weak texture region	5(a),(b)	521,504	0.044	0.237	0.627	2.559	0.478	3.945
	5(c),(d)	552,490	0.046	0.265	0.609	2.582	0.496	3.998
	5(e),(f)	478,433	0.046	0.209	0.477	2.119	0.493	3.344
Not detect weak texture region	5(a),(b)	511,491	0.044	0.238	0.460	2.181	0.562	3.485
	5(c),(d)	541,480	0.046	0.274	0.475	2.168	0.485	3.448
	5(e),(f)	466,416	0.046	0.229	0.406	1.758	0.515	2.954

TABLE 3. Evaluation results summary: SSIM.

algorithm	diamondhead		goldengate		halfdome		rio		shanghai	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
Harris+*NCC*	0.9060	0.0723	0.9272	0.0721	0.8435	0.0801	0.9270	0.0679	0.9218	0.0546
Cui's algorithm	0.9155	0.0637	0.9043	0.0726	0.8610	0.0779	0.9459	0.0462	0.9215	0.0530
Detect weak texture region	0.9592	0.0419	0.9852	0.0133	0.9392	0.0455	0.9760	0.0344	0.9729	0.0251
Not detect weak texture region	0.9245	0.0643	0.9530	0.0357	0.9024	0.0516	0.9585	0.0422	0.9434	0.0335

Furthermore, as the Census matching algorithm for weak texture regions, interval sampling for general texture regions and omitted calculation for flat texture regions can all contribute to computation time, as a result of which the overall speed is increased by 26.9% $\sim$ 29.9% when compared to single *NCC*.

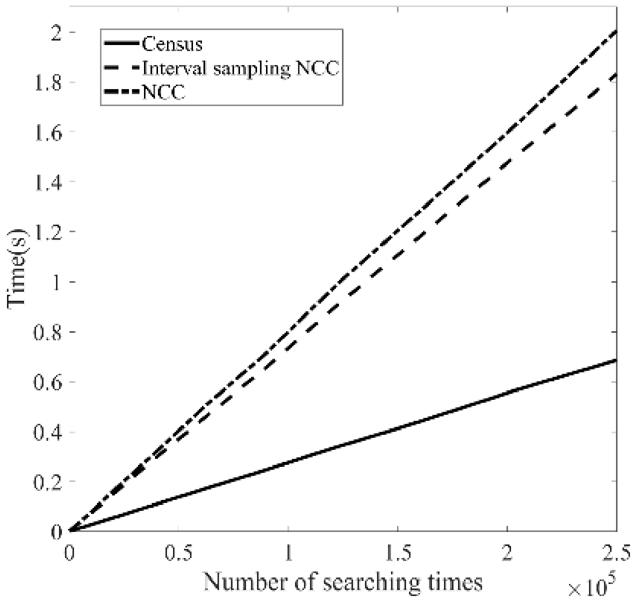


FIGURE 11. Calculation time *vs* number of searching times for *NCC*, Interval sampling *NCC* and Census respectively.

The proposed algorithm as a whole is compared with the Harris+*NCC* image stitching algorithm, as shown in Table 2. It can be seen that the overall speed improves by 14.1%~18.4% when detecting the weak texture regions for higher image splicing quality purpose. Furthermore, the overall speed is increased by 23.9%~28.4% by discounting the weak texture regions. In contrast, the number of corner points decreases by 1.9%~3.9%.

The proposed algorithm is compared against the Harris + *NCC* and Cui algorithms, as shown in Table 2. It can be seen that the matching is the most time costive stage in the whole process. For the most costive *NCC* matching stage, it requires $O(M \times N)$ to conduct search to find out the matching points, for which M and N are the detected corner points of two images waiting to be stitched. However, the proposed texture segmentation limits the searching area, so that the searching complexity are $O(M_w \times N_w)$, $O(M_m \times N_m)$ and $O(M_s \times N_s)$ for weak, middle and strong texture regions respectively. According to the inequality (8), the the overall speed is improved even if an additional texture classification stage is introduced.

$$\begin{aligned} M \times N &= (M_f + M_w + M_m + M_s)(N_f + N_w + N_m + M_s) \\ &> M_w N_w + M_m N_m + M_s N_s \end{aligned} \tag{8}$$

where, the subscripts f, w, m and s represent flat, weak, middle and strong texture regions respectively.

Besides, for a given correlation window size, the computational complexity for the *NCC*, interval *NCC* and Census are $O(1)$, as a result of which the matching time is linear to the searching times as demonstrated in Figure 11. It also can be seen that computational efficiency is boosted in the above order.

E. IMAGE STITCHING RESULTS AND QUALITY ANALYSIS

The image matching results are presented in Figure 12 in comparison to the *NCC* and Cui algorithms. It can be seen that the proposed algorithm restricts matched points in complex texture area and increases feature points in the weak and middle ones, thus providing a more balance corner point distribution.

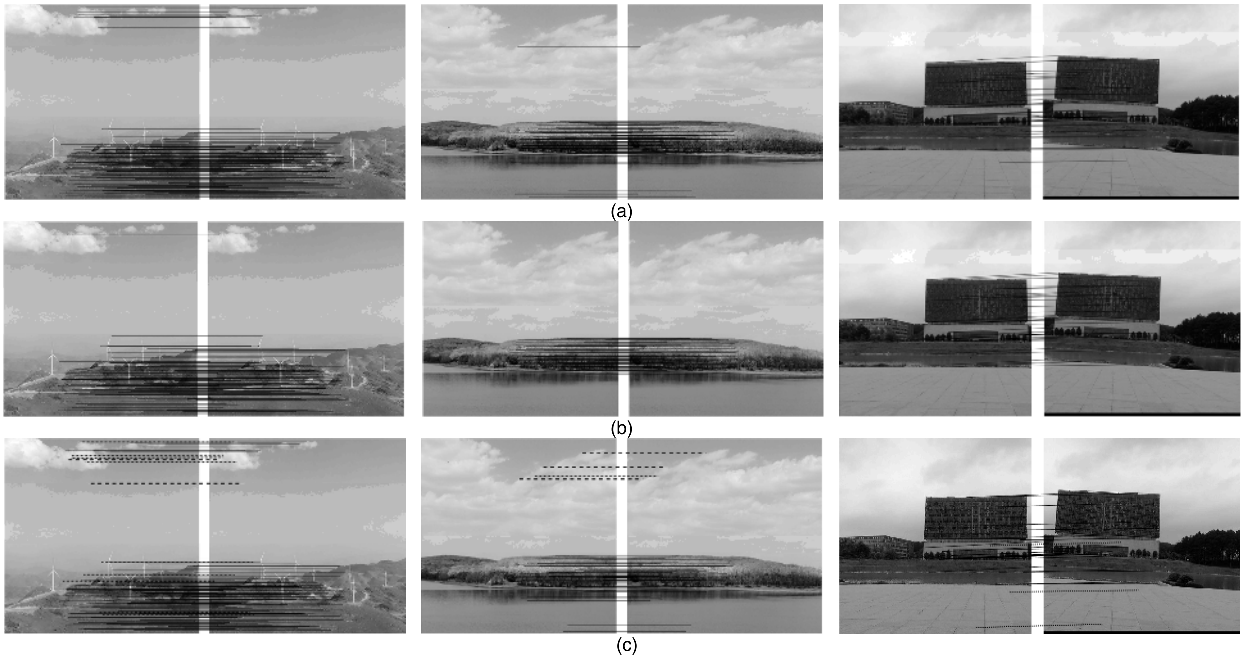
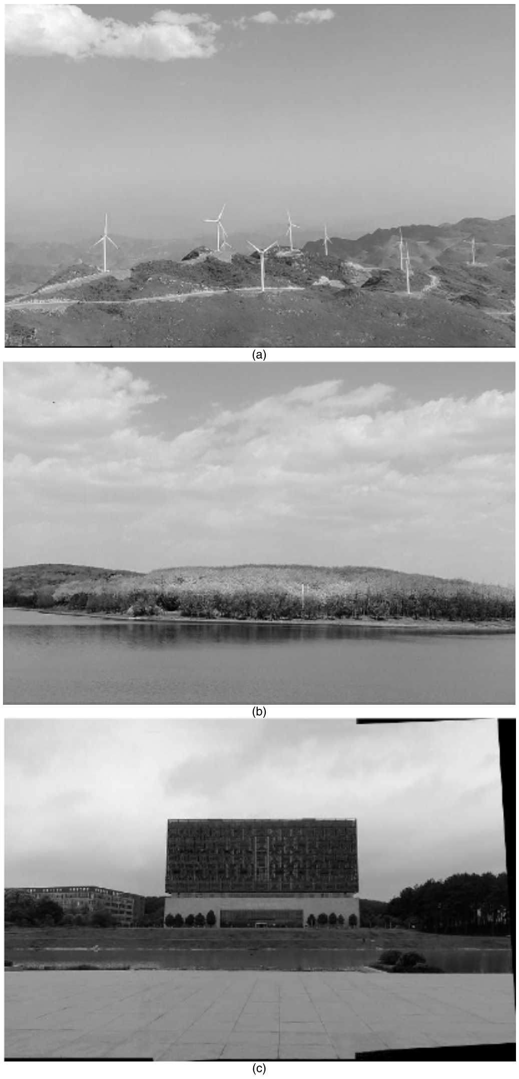


FIGURE 12. The results of matching features detected by:(a) Harris+*NCC* algorithms; (b) Cui's algorithm; (c) The proposed algorithm, dot line: Middle texture regions, dashed line: Weak texture regions.

TABLE 4. Evaluation results summary: PSNR.

algorithm	diamondhead		goldengate		halfdome		rio		shanghai	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
Harris+*NCC*	26.5391	4.1681	26.6005	1.9873	23.1037	2.5117	26.3509	4.1988	26.4096	2.6655
Cui's algorithm	27.0370	3.6924	25.7648	2.0852	23.3428	1.9333	27.0032	3.9096	26.4327	3.2511
Detect weak texture region	30.8891	5.8389	33.3983	3.6956	27.0209	3.6836	31.5565	4.7240	32.0420	4.1739
Not detect weak texture region	27.2253	4.9408	28.3718	3.4721	24.7768	2.4603	28.4361	3.6287	27.8270	2.7575


(a)

(b)

(c)

FIGURE 13. Image stitching results: (a) Image of the mountain; (b) Image of the island; (c) Image of the building.

The stitching result is indicated in Figure 13. Furthermore, in order to assess the stitching quality proposed by using this algorithm, SSIM [25] and PSNR are employed to conduct test on images with no scale transformations in the panorama dataset. The results are shown in Table 3 and 4. It is noteworthy that the stitching quality will be relatively poorer, if the corner points in weak texture regions fail to be detected for computational efficiency, which also indicates that feature points from the regions also contributes to the image quality.

IV. CONCLUSION

In this paper, multi-threshold corner detection and region matching algorithm based on texture classification are proposed aiming at the unreasonable corner point distribution caused by a single threshold value. In the multi-threshold value calculation stage, the texture features of image sub-blocks are judged based on gray histogram and then classified into four types, including strong, middle, weak and flat texture regions. Furthermore, an interval calculation and prediction strategy is introduced to the classification process for improvement to computation efficiency. In the subsequent image stitching stage, Census algorithm, interval sampling *NCC* and complete *NCC* algorithm are employed to match corner points based on the attributes of different texture region, which can improves the speed by 26.9%∼29.9% when compared to single *NCC*.

Due to distinct treatment for different texture regions, the proposed algorithm is capable to provide more uniform distributed corner points, and the overall stitching speed of the algorithm improves by 14.1%∼18.4%. Furthermore, the computation speed can increase by 23.9%∼28.4% for sacrificing 1.9%∼3.9% corner points if the weak texture region is discounted. Therefore, the algorithm has a massive potential of applications where more uniformly distribution of corner points and high computation efficiency are required.

REFERENCES

[1] D. Ghosh and N. Kaabouch, "A survey on image mosaicing techniques," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 1–11, Jan. 2016.

[2] K. Zhang, X. Li, and J. Zhang, "A robust point-matching algorithm for remote sensing image registration," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 2, pp. 469–473, Feb. 2014.

[3] Y. Wu, W. Ma, M. Gong, L. Su, and L. Jiao, "A novel point-matching algorithm based on fast sample consensus for image registration," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 43–47, Jan. 2015.

[4] L. Yang, J. Cao, L. Tang, B. Gao, H. Wang, and H. Gao, "Optimized design of automatic panoramic images mosaic," *Infrar. Laser Eng.*, vol. 43, no. 3, pp. 985–990, Mar. 2014.

[5] C. Harris and M. J. Stephens, "A combined corner and edge detector," in *Proc. IEEE Alvey Vis. Conf.*, Jan. 1988, pp. 147–151.

[6] M. Gan, Y. Cheng, Y. Wang, and J. Chen, "Hierarchical particle filter tracking algorithm based on multi-feature fusion," *J. Syst. Eng. Electron.*, vol. 27, no. 1, pp. 51–62, Mar. 2016.

[7] L. Chen, W. Lu, J. Ni, W. Sun, and J. Huang, "Region duplication detection based on Harris corner points and step sector statistics," *J. Vis. Commun. Image Represent.*, vol. 24, no. 3, pp. 244–254, 2013.

[8] H. Li, J. Qin, X. Xiang, L. Pan, W. Ma, and N. N. Xiong, "An efficient image matching algorithm based on adaptive threshold and RANSAC," *IEEE Access*, vol. 6, pp. 66963–66971, 2018.

[9] J. Cui, J. Xie, T. Liu, X. Guo, and Z. Chen, "Corners detection on finger vein images using the improved Harris algorithm," *Optik*, vol. 125, no. 17, pp. 4668–4671, Sep. 2014.

[10] Z. Wang, R. Li, Z. Shao, M. Ma, J. Liang, W. Liu, J. Wang, and Y. Liu, "Adaptive Harris corner detection algorithm based on iterative threshold," *Mod. Phys. Lett. B*, vol. 31, no. 15, May 2017, Art. no. 1750181.

[11] S. Shen, X. Zhang, and W. Heng, "Auto-adaptive harris corner detection algorithm based on block processing," in *Proc. Int. Symp. Signals, Syst. Electron.*, Sep. 2010, pp. 1–4.

[12] K. S. Changan and P. G. Chilveri, "Stereo image feature matching using Harris corner detection algorithm," in *Proc. Int. Conf. Autom. Control Dyn. Optim. Techn.*, Sep. 2016, pp. 691–694.

[13] A. Banharnsakun, "Feature point matching based on ABC-NCC algorithm," *Evolving Syst.*, vol. 9, no. 1, pp. 71–80, Mar. 2018.

[14] J. Sun, Y. Liu, Y. Ding, X. Zhu, and J. Xi, "NCC feature matching optimized algorithm based on constraint fusion," in *Proc. IEEE 3rd Int. Conf. Image, Vis. Comput.*, Jun. 2018, pp. 336–341.

[15] J. Banks, M. Bennamoun, and P. Corke, "Non-parametric techniques for fast and robust stereo matching," in *Proc. IEEE Region Annu. Conf., Speech Image Technol. Comput. Telecommun.*, Dec. 1997, pp. 365–368.

[16] V. Borisagar and M. Zaveri, "Census and segmentation-based disparity estimation algorithm using region merging," *J. Signal Inf. Process.*, vol. 6, no. 3, pp. 191–202, Aug. 2015.

[17] K. Ambrosch, W. Kubinger, M. Humenberger, and A. Steininger, "Hardware implementation of an SAD based stereo vision algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–6.

[18] D. Min, J. Lu, and M. N. Do, "Joint histogram-based cost aggregation for stereo matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2539–2545, Oct. 2013.

[19] Y. Quan, Y. Huang, and H. Ji, "Dynamic texture recognition via orthogonal tensor dictionary learning," in *Proc. IEEE int. Conf. Comput. Vis.*, Dec. 2015, pp. 73–81.

[20] Y. Quan, Y. Xu, Y. Sun, and Y. Luo, "Lacunarity analysis on image patterns for texture classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 160–167.

[21] Y. Xu, Y. Quan, H. Ling, and H. Ji, "Dynamic texture classification using dynamic fractal analysis," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1219–1226.

[22] Y. Quan, Y. Xu, and Y. Sun, "A distinct and compact texture descriptor," *Image Vis. Comput.*, vol. 32, no. 4, pp. 250–259, Apr. 2014.

[23] Y. Chen, Y. Zhao, and S. Wang, "Fast image stitching method based on SIFT with adaptive local image feature," *Chin. Opt.*, vol. 9, no. 4, pp. 415–422, Aug. 2016.

[24] J. Brandt, "Transform coding for fast approximate nearest neighbor search in high dimensions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Aug. 2010, pp. 1815–1822.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

ZETIAN TANG received the B.S. degree in communication engineering from the Lanzhou University of Technology, China, in 2017. He is currently pursuing the M.S. degree in electronics and communication engineering with the College of Big Data and Information Engineering, Guizhou University, China. His research interests include image processing and computer vision.

ZHAO DING received the Ph.D. degree in physics from the University of Arkansas, Fayetteville, AR, USA, in 2002. Since 2003, he has been with Guizhou University, where he is currently a Professor with the Key Laboratory of Micro-Nano-Electronics. His research interests include optoelectronic devices, low-dimensional semiconductor material and devices, atomic force microscope, and scanning tunneling microscope image processing.

RUIMIN ZENG received the B.S. degree in electronic information engineering from the Chengdu University of Information Technology, in 2017. She is currently pursuing the M.S. degree in electronic science and technology with the College of Big Data and Information Engineering, Guizhou University. Her research interests include image processing and photoelectric.

YANG WANG received the B.S. degree in electronic science and technology from Guizhou University, China, in 2018, where he is currently pursuing the M.S. degree in electronic science and technology with the College of Big Data and Information Engineering. His research interests include image processing and computer vision.

JUN WEN received the B.S. and Ph.D. degrees in electronic engineering from Xidian University, Xi'an, China, in 2006 and 2011, respectively, where she was a Lecturer with the School of Electronic Engineering, from 2011 to 2013. Since 2014, she has been an Associate Professor with the School of Computer and Electrical Information, Guangxi University, Nanning, China. Her research interests include space-time adaptive processing and SAR ground moving target indication.

LIFENG BIAN received the B.S. degree in applied physics and microelectronics technology from the Hefei University of Technology, Hefei, China, in 1998, the master's degree from the Institute of Intelligent machine, Chinese Academy of Sciences (CAS), in 2001, and the Ph.D. degree from the State Key Laboratory for Superlattices and Microstructures, Institute of Semiconductors, CAS, in 2004. Until 2006, she was a Postdoctoral Researcher with the Paul-Drude-Institute for Solid-State Electronics, Berlin, Germany.

Since 2006, she has been an Associate Professor with the Suzhou Institute of Nano-Tech and Nano-Bionics, CAS, Suzhou, China, where she was a Professor, in 2013. Her research interest includes semiconductor optical-electronic devices, such as laser and the LED of GaAs- and GaN-based materials and device.

CHEN YANG received the Ph.D. degree in microelectronics and solid electronics from the Institute of Semiconductor, Chinese Academy of Sciences, Beijing, China, in 2010. From 2010 to 2012, he was an Engineer with Synopsys Company Ltd., for device library. From 2014 to 2015, he was a Postdoctoral Researcher with the KTH Royal Institute of Technology, Stockholm, Sweden. He is currently an Associate Professor with Guizhou University. His research interests include photovoltaic device testing systems, computer vision, and its application in semiconductor device testing.