

Research Article

A Fast Image Stitching Algorithm via Multiple-Constraint Corner Matching

Minchen Zhu,¹ Weizhi Wang,² Binghan Liu,¹ and Jingshan Huang³

¹ College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian 350108, China

² College of Civil Engineering, Fuzhou University, Fuzhou, Fujian 350108, China

³ School of Computing, University of South Alabama, Mobile, AL 36688, USA

Correspondence should be addressed to Binghan Liu; lbh@fzu.edu.cn

Received 8 September 2013; Accepted 30 September 2013

Academic Editor: Artde Donald Kin-Tak Lam

Copyright © 2013 Minchen Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Video panoramic image stitching is in general challenging because there is small overlapping between original images, and stitching processes are therefore extremely time consuming. We present a new algorithm in this paper. Our contribution can be summarized as a multiple-constraint corner matching process and the resultant faster image stitching. The traditional Random Sample Consensus (RANSAC) algorithm is inefficient, especially when stitching a large number of images and when these images have quite similar features. We first filter out many inappropriate corners according to their position information. An initial set of candidate matching-corner pairs is then generated based on grayscales of adjacent regions around each corner. Finally we apply multiple constraints, e.g., their midpoints, distances, and slopes, on every two candidate pairs to remove incorrectly matched pairs. Consequently, we are able to significantly reduce the number of iterations needed in RANSAC algorithm so that the panorama stitching can be performed in a much more efficient manner. Experimental results demonstrate that (i) our corner matching is three times faster than normalized cross-correlation function (NCC) rough match in traditional RANSAC algorithm and (ii) panoramas generated from our algorithm feature a smooth transition in overlapping image areas and satisfy human visual requirements.

1. Introduction

To stitch images and form a video panoramic image, the similarity of overlapping regions among adjacent images needs to be calculated in the first place. State-of-the-art algorithms for image registration (sometimes also referred to as “image alignment”) can be classified into intensity-based, frequency domain-based, and feature-based ones [1–7]. Intensity-based algorithms usually involve a large amount of computation and therefore are not appropriate for image alignment when there is image rotation and scaling. On the other hand, algorithms based on frequency-domain are in general faster and can handle well small translation, rotation, and scaling. Unfortunately, the performance of frequency domain-based algorithms will be degraded when dealing with scenarios where smaller overlapping regions exist. Feature-based algorithms utilize a small number of invariant points, lines, or edges to align images. One significant advantage of these algorithms is that the computational

complexity will be reduced due to less information that needs to be processed. Additionally, feature-based algorithms are robust to changes in image intensity. However, there is one serious issue identified for many existing algorithms. Most of these algorithms make use of an exhaustive search that is based on template matching. As a result, the computation, although already decreased to some extent, is still intensive, which does not meet the real-time requirement usually found in video panorama stitching.

We present in this paper a new algorithm to handle the aforementioned challenge. Our algorithm is motivated by the observation that adjacent images usually have small overlapping and small difference of translation, rotation, and scaling between each other. The proposed algorithm is based upon our innovative multiple-constraint corner matching. First, we filter out large numbers of candidate corners according to their position information. We then generate an initial set of matching-corner pairs based on grayscales of each corner's adjacent regions. Finally, multiple constraints, for example,

their midpoints, distances, and slopes, will be applied on every two candidate pairs to remove incorrectly matched pairs. Consequently, we are able to significantly reduce the number of iterations that are needed in conventional Random Sample Consensus (RANSAC) algorithm [8]. As a result, the video panoramic image stitching can be performed a lot more efficiently.

The rest of this paper is organized as follows. Section 2 introduces in detail our methodology; Section 3 describes experimental results; and Section 4 concludes with future research directions.

2. Methodology

2.1. Corner Selection. Harris algorithm [2] detects corners through the differential of the corner score and the autocorrelation matrix. Suppose that an image has an intensity of $I(x, y)$ and an image patch over the area $w(x, y)$ is shifted by (u, v) the intensity change, $E(u, v)$, of the pixel (x, y) can then be calculated by (1) where $M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} = w(x, y) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$ ($I(x)$ and $I(y)$ are partial derivatives of the pixel, respectively, and w is the Gaussian function to filter noises). One has

$$E(u, v) = [uv] M \begin{bmatrix} u \\ v \end{bmatrix}. \quad (1)$$

The corner response function is defined in (2) with k in the range of $[0.04, 0.06]$. Any pixels whose R value is greater than a threshold, λ , can be selected as candidate corners:

$$R = \det(M) - k(\text{trace}(M))^2. \quad (2)$$

Note that λ depends on characteristics of actual images, size and texture for example; Usually λ is determined indirectly: pixels are sorted in a descending order of their R values; then the first Sum pixels are selected as corners.

Harris detector only involves the first order difference and filtering operations of pixel grayscale, with low computational complexity. A large number of corners can be detected in regions with rich texture, whereas fewer corners will be selected in regions with less texture information. Therefore, selected corners are not evenly distributed; that is, corners tend to cluster around regions with richer texture. Zhao et al. proposed an algorithm in [9] where they fragmented the original image into several regions. A fixed number of corners with top R values were selected in each region as candidate corners; all such candidate corners (a total of Sum) were then sorted in their R values. Finally a scaling parameter, k , whose range is $(0, 1)$, was applied to finalize the corner selection, that is, generating a total of $k \times \text{Sum}$ corners. To assure that each region contains some finalized corners, this algorithm iteratively applied different k values in an ascending order and the iteration was terminated as soon as there was at least one finalized corner for each region. Because of its ability to select corners with relatively high quality, we adopt this algorithm when selecting Harris corners from adjacent images to be stitched.

2.2. Multiple-Constraint Corner Matching. The traditional RANSAC algorithm is inefficient, especially when stitching a large number of images and when these images have quite similar features. Thus, it does not meet the real-time requirement commonly found in video panorama stitching. Note that in the field of video panorama stitching, more often than not, adjacent images have highly similar features with each other, that is, small difference of translation, rotation, and scaling between each other. Based on this insight, we propose to apply multiple constraints on candidate matching-corner pairs to remove incorrectly matched pairs. As such, we can significantly reduce the number of iterations needed in RANSAC algorithm.

2.2.1. Create a Corner Similarity Matrix between Adjacent Images. Suppose that the image I has a resolution of $W \times H$, and the k th corner in I is denoted by I_k (with coordinates $I_k \cdot x$ and $I_k \cdot y$, and intensity $I(x_k, y_k)$). One corner from the left image (I_i^l) and another corner from the right image (I_j^r) can be matched with each other if the following conditions are satisfied: (i) the difference between y coordinates of these two corners is no greater than $H/3$; (ii) the x coordinate of the left corner is greater than or equal to that of the right corner; and (iii) there is a high intensity correlation between two corners. Accordingly, we utilize (3) to calculate pairwise corner similarity and create a similarity matrix between adjacent images I^l and I^r :

$$\text{sim}(i, j) = \begin{cases} |\text{NCC}(I_i^l, I_j^r)| & \text{if } |I_i^l \cdot y - I_j^r \cdot y| < \lambda_h, \\ & I_i^l \cdot x \geq I_j^r \cdot x, \\ 0 & \text{else.} \end{cases} \quad (3)$$

In (3), λ_h is the threshold of the difference between y coordinates of two corners, and normalized cross-correlation (NCC) function is the one described in [10]. Suppose that the similarity window size is $(2w + 1) \times (2w + 1)$; NCC is then calculated as

$$\begin{aligned} \text{NCC}(I_i^l, I_j^r) &= \frac{\sum_{u=-w}^w \sum_{v=-w}^w D_l(i, u, v) \cdot D_r(j, u, v)}{\sqrt{\sum_{u=-w}^w \sum_{v=-w}^w D_l(i, u, v)^2 \cdot \sum_{u=-w}^w \sum_{v=-w}^w D_r(j, u, v)^2}}, \end{aligned} \quad (4)$$

where

$$\begin{aligned} D_l(i, u, v) &= I^l(x_i + u, y_i + v) - \bar{I}_i^l, \\ D_r(j, u, v) &= I^r(x_j + u, y_j + v) - \bar{I}_j^r, \end{aligned} \quad (5)$$

and \bar{I}_i^l and \bar{I}_j^r are the mean intensity of windows around corners I_i^l and I_j^r , respectively. In addition, we further filter out corner pairs with low similarity using (6), where λ_n is the similarity threshold (a real number that is greater than 0.5):

$$\text{sim}(i, j) = \begin{cases} \text{sim}(i, j) & \text{if } \text{sim}(i, j) > \lambda_n, \\ 0 & \text{else.} \end{cases} \quad (6)$$

In brief, we use (3) and (6) to calculate pairwise corner similarity, $\text{sim}(i, j)$, resulting in a similarity matrix between two adjacent images.

2.2.2. Generate an Initial Set of Matching-Corner Pairs. A set of indexes of matching-corner pairs is generated by the following procedure: in each row of the similarity matrix obtained previously, we find the column index so that the corresponding cell in the matrix has the maximum value for that row, and the pair of (*row index*, *column index*) is added into the set. After we process all rows in the matrix, we will obtain a set of index pairs, L^l . Such a procedure is formally described in (7), where Sum_l is the predefined total number of corners in the left image I^l :

$$L^l = \{(i, j) \mid \forall i \in [1, \text{Sum}_l], \quad (7)$$

$$s(i, j) = \max(s(i, :)), s(i, j) \neq 0\}.$$

Similarly, we can obtain another set of index pairs, L^r , by searching the maximum row index for each column. Equation (8) is a formal description of this procedure, where Sum_r is the predefined total number of corners in the right image I^r :

$$L^r = \{(i, j) \mid \forall j \in [1, \text{Sum}_r], \quad (8)$$

$$s(i, j) = \max(s(:, j)), s(i, j) \neq 0\}.$$

In general, Sum_l in (7) and Sum_r in (8) can take different values. In our algorithm we use the same value for these two parameters. Now we compare two sets, L^l and L^r . If a row index and a column index happen to have each other as the other component in a pair, their similarity will be adjusted to 1. That is, if two corners mutually find their ‘‘best’’ match as each other, such a pair will have an updated similarity value of 1. Equation (9) formalizes this procedure of similarity adjustment:

$$\text{sim}_{\text{updated}}(i, j) = \begin{cases} 1 & \text{if } (i, j) \in L^l \cap L^r, \\ \text{sim}(i, j) & \text{else.} \end{cases} \quad (9)$$

Finally we generate an initial set of matching-corner pairs, L , by a union of L^l and L^r , shown in (10). Note that this initial set of pairs is already reduced in size compared with NCC rough match in traditional RANSAC algorithm because as shown in (3) we have already filtered out some inappropriate corners according to their positions in respective regions (i.e., their coordinate values). One has

$$L = L^l \cup L^r = \{(l, r) \mid (l, r) \in L^l \text{ or } (l, r) \in L^r\}. \quad (10)$$

2.2.3. Apply Multiple Constraints on Matching-Corner Pairs. Consider two initial matching-corner pairs in Figure 1, (l_m, r_m) and (l_n, r_n) , along with their respective midpoints, that is, l_{mn} between l_m and l_n and r_{mn} between r_m and r_n . Let δ_m and \bar{m} be the slope and length of the segment formed by l_m and r_m , respectively, and let δ_n and \bar{n} be the slope and length of the segment formed by l_n and r_n , respectively. We design

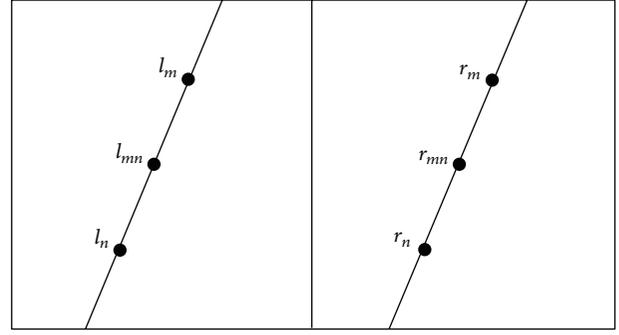


FIGURE 1: Two initially matching-corner pairs.

three constraints to be applied to these two matching-corner pairs, as follows:

$$\begin{aligned} \text{constraint 1: } & |\delta_m - \delta_n| < \lambda_\delta, \\ \text{constraint 2: } & |\bar{m} - \bar{n}| < \lambda_d, \\ \text{constraint 3: } & |\text{NCC}(I_{l_{mn}}^l, I_{r_{mn}}^r)| > \lambda_n. \end{aligned} \quad (11)$$

The intuition of (11) is that, between two matching pairs, not only the intensity of their respective midpoints (constraint 3) should be correlated, but also the slope (constraint 1) and length (constraint 2) of the segments formed between these two pairs should be similar with each other as well. According to multiple constraints specified in (11), we calculate pairwise similarity between every two initial matching pairs using (12) and generate a matrix D of size $K \times K$, with K being the cardinality of L generated in (10). One has

$$D(m, n) = \begin{cases} \frac{\text{sim}_{\text{updated}}(l_m, r_m) + \text{sim}_{\text{updated}}(l_n, r_n) + |\text{NCC}(I_{l_{mn}}^l, I_{r_{mn}}^r)|}{3} & \text{if satisfying all 3 constraints} \\ 0 & \text{else.} \end{cases} \quad (12)$$

2.2.4. Generate the Final, Reduced Set of Matching-Corner Pairs. Among a total of K initial matching-corner pairs, according to (13), we search for a special pair, t , which has the strongest correlation with all other pairs:

$$t = \underset{i \in [1, K]}{\text{argmax}} \left(\sum_{j=1}^K D(i, j) \right). \quad (13)$$

Then we refer back to the matrix D generated previously and find out all initial matching pairs that have some correlation with the aforementioned special pair, t ; that is, an initial matching-corner pair will be output to the final, further reduced set as long as the cell in D corresponding to this pair and the special pair t has a nonzero value. Equation (14) formally specifies this final selection step, and the resultant set T is the finalized, reduced set of matching-corner pairs. Note that the size of T is further reduced from that of L , and we explained earlier that L is already reduced in size

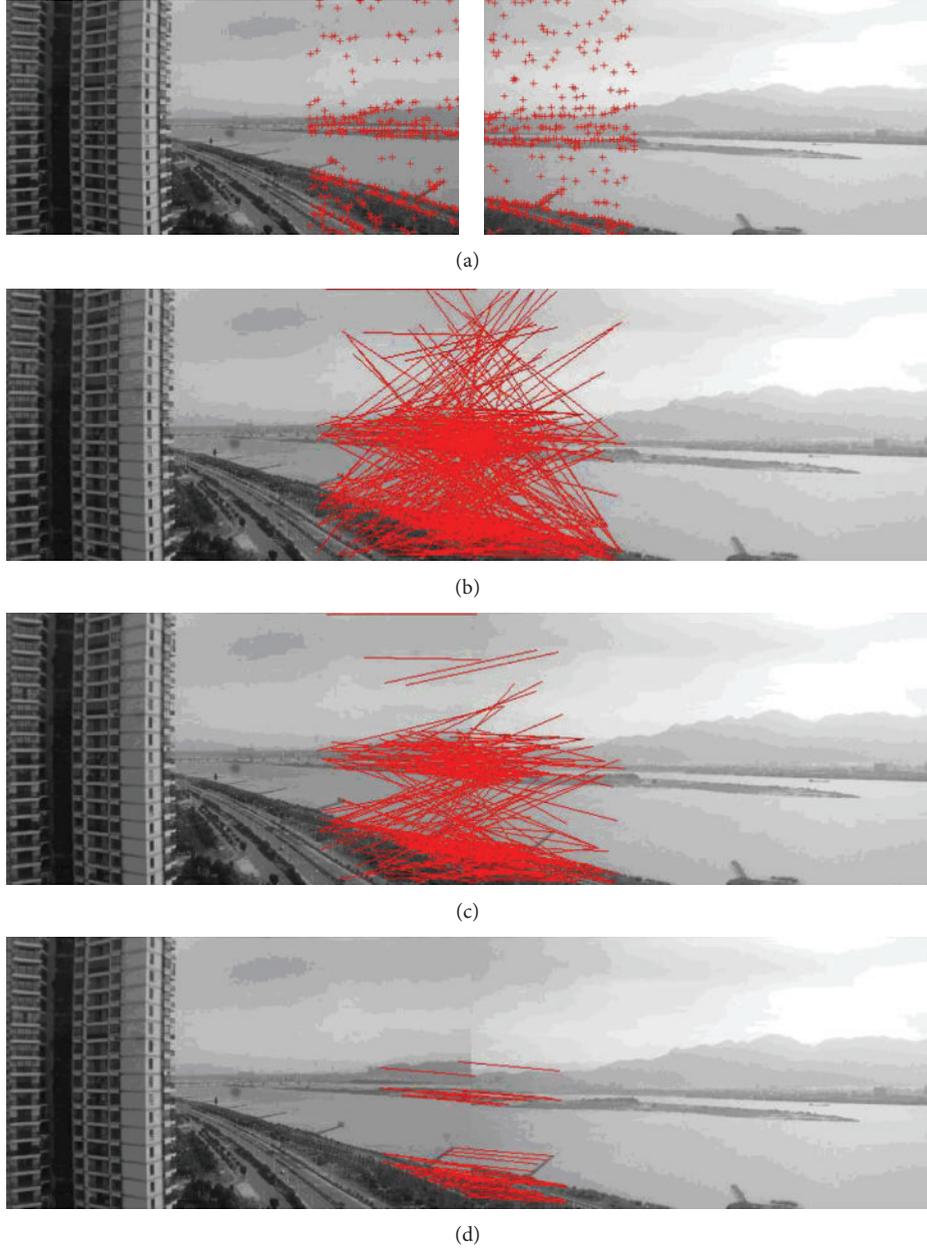


FIGURE 2: Experiments on corner matching.

compared with NCC rough match in traditional RANSAC algorithm:

$$T = \{l \mid \forall l \in L, D(t, l) \neq 0\}. \quad (14)$$

2.3. Image Stitching. After we obtain a reduced set of matching-corner pairs between two original images to be stitched, we select one as the reference image and calculate the affine transformation parameters using RANSAC algorithm. Based on these parameters we map pixel coordinates in the other image into the coordinate system of the reference image. The light conditions may vary among different cameras; therefore, the panorama to be generated may be inconsistent in terms of its intensity. To obtain a smooth

transition in overlapping areas among images to be stitched, we utilize the weighted-sum method introduced in [10] to perform a gradual fading-in and fading-out image stitching process to generate the final video panoramic image.

3. Experimental Results and Analysis

3.1. Experimental Environment and Parameter Setup. Experimental Environment and Parameter Setup are as follows: PC: CPU E2200 + 2.2 GHz, 4 GB memory, Matlab 7.0; image resolution: 1280×720 .

Various parameters described earlier in Section 2 were set as follows. Note that the setting of these experimental

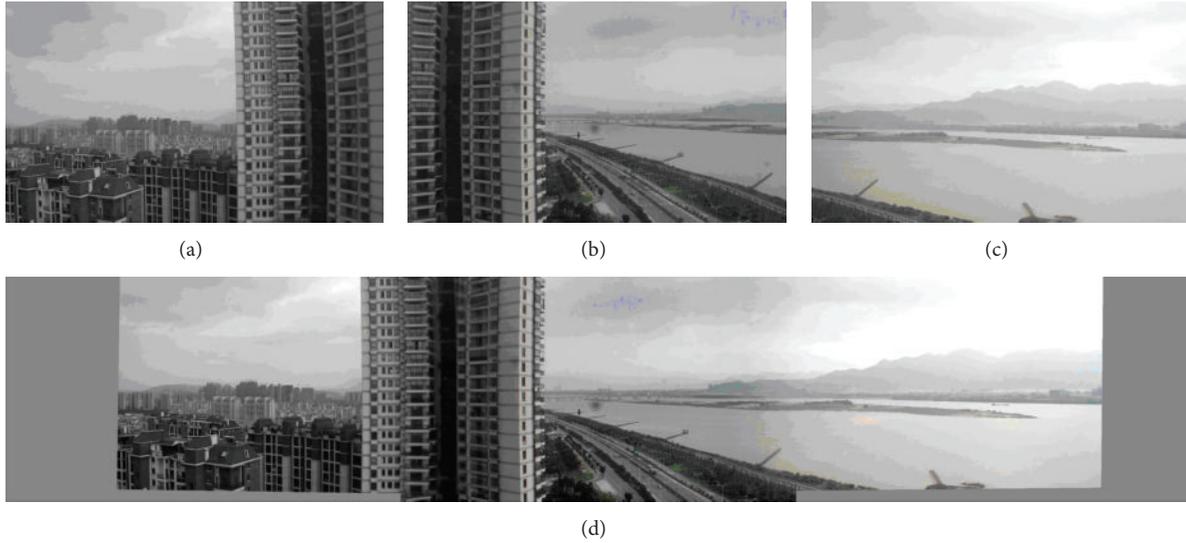


FIGURE 3: Experiments on image stitching.

parameters was based upon our previous experience from numerous experiments.

- (i) The difference of y coordinates of adjacent cameras was not greater than $H/3$; that is, λ_h in (3) was set to $H/3$;
- (ii) The horizontal overlapping was not great than $W/3$;
- (iii) The original image was segmented into regions of size 80×80 , and the number of corners for each region was set to six;
- (iv) The similarity threshold, λ_n , in (6) was set to 0.75, and the similarity window size in (4) was set to 7×7 ; that is, w was set to three.

3.2. Evaluation on Corner Matching. The experimental results are demonstrated in Figure 2. Two original images with corners selected using algorithm in [9] are exhibited in Figure 2(a). We chose the right one-third region of the left image and the left one-third region of the right image as two regions to perform corner matching. So we had a total of 45 ($= (H/80) \times ((W/3)/80)$) segmented regions, and the total number of corners is 270 ($= 6 \times 45$). The total number of matching-corner pairs from NCC rough match in traditional RANSAC algorithm was 388 (Figure 2(b)), whereas the total numbers of initial and finalized matching-corner pairs from our algorithm were 332 (Figure 2(c)) and 35 (Figure 2(d)), respectively. This result verified our earlier discussion in Section 2.2; that is, the initial set of pairs is already reduced in size compared with NCC rough match in traditional RANSAC algorithm because as shown in (3) we have already filtered out some corners according to their positions in respective regions (i.e., their coordinate values). Note that most of the 35 matching pairs in Figure 2(d) were correct ones. In addition, as demonstrated in Table 1, our multiple-constraint corner matching was three times faster than NCC rough match in traditional RANSAC algorithm. The reason

TABLE 1: Efficiency comparison between traditional RANSAC and our algorithm.

	NCC rough match in traditional RANSAC	Our initial matching	Multiple- constraint matching
Number of matching pairs	388	332	35
Time spent (s)	8.74	2.30	2.94

for us to obtain a much shorter matching process is that traditional RANSAC algorithm needs to calculate the NCC function, which is very time consuming, for *all pairwise combinations* of corners, whereas in our algorithm only a *small number of combinations* need to be considered. To be more specific, (3) ignores all corners that do not meet the position requirement, and we further avoid NCC calculation if two initial pairs do not satisfy the first two constraints specified in (11). More experimental results can be found at the following project Web link: <http://www.soc.southalabama.edu/~huang/ImageStitching/ExperimentResults.rar>.

3.3. Evaluation on Image Stitching. The experimental results are demonstrated in Figure 3. We performed both the regional Harris corner selection and multiple-constraint corner matching between Figures 3(a) and 3(b) and between Figures 3(b) and 3(c), respectively. After we obtained two finalized sets of matching-corner pairs, we selected Figure 3(b) as the reference image and calculated the affine transformation parameters as discussed earlier in Section 2.3. We then mapped pixel coordinates in Figures 3(a) and 3(c) into the coordinate system of Figure 3(b), respectively. Finally we performed a gradual fading-in and fading-out image stitching process. The final result in Figure 3(d) clearly demonstrated that (i) our corner matching was accurate; (ii) we obtained a smooth transition in

overlapping areas among images to be stitched; and (iii) the panorama generated satisfied human visual requirements. Similarly, more experimental results can be found at the following link: <http://www.soc.southalabama.edu/~huang/ImageStitching/ExperimentResults.rar>.

4. Conclusions

We presented an innovative algorithm to handle challenges in video panoramic image stitching, for example, small overlapping regions and extremely time-consuming stitching processes. Our contribution can be summarized as (i) a multiple-constraint corner matching and (ii) a more efficient image stitching process. To overcome the inefficient corner matching in traditional RANSAC algorithm, we first filtered out a large number of corners according to their position information. We then generated an initial set of matching-corner pairs based on grayscales of adjacent regions around each corner. Finally we applied multiple constraints on every two candidate pairs to remove incorrectly matched pairs. We were able to significantly reduce the number of iterations needed in RANSAC algorithm, resulting in a much more efficient panorama stitching process. Experimental results (both those that were detailed in this paper itself and those additional ones in the Web link provided) demonstrated that (i) our corner matching is three times faster than traditional RANSAC matching and (ii) panoramas generated from our algorithm feature a smooth transition in overlapping image areas and satisfy human visual requirements.

One possible future research direction is to investigate on automatically determining the total number of corners according to the image texture information. Another interesting future work is to handle the motion ghost challenge during image stitching.

Acknowledgment

This research was supported by the Project of Fujian Province under Grant nos. 2011Y0040, 2012J01263, and 2013J01186.

References

- [1] E. De Castro and C. Morandi, "Registration of translated and rotated images using finite Fourier transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 700–703, 1987.
- [2] C. G. Harris and M. Stephen, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, Manchester, UK, 1998.
- [3] Z. Li, "A quick image stitching algorithm for images with overlapping borders," *Computer Engineering*, vol. 26, no. 5, pp. 37–38, 2000.
- [4] A. Chalechale, G. Naghdy, and A. Mertins, "Sketch-based image matching using angular partitioning," *IEEE Transactions on Systems, Man, and Cybernetics Part A*, vol. 35, no. 1, pp. 28–41, 2005.
- [5] Q. Zhu, B. Wu, and Z.-X. Xu, "Seed point selection method for triangle constrained image matching propagation," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 2, pp. 207–211, 2006.
- [6] Y. Zhang, G. Gao, and K. Jia, "A fast algorithm for cylindrical panoramic image based on feature points matching," *Journal of Image and Graphics*, vol. 14, no. 6, pp. 1188–1193, 2009.
- [7] S. Cao, J. Jiang, G. Zhang, and Y. Yuan, "Multi-scale image mosaic using features from edge," *Computer Research and Development*, vol. 48, no. 9, pp. 1788–1793, 2011.
- [8] H. Yu and W. Jin, "Evolution of research on digital image mosaics methods," *Infrared Technology*, vol. 31, no. 6, pp. 348–353, 2009.
- [9] W. Zhao, S. Gong, C. Liu, and X. Shen, "A self-adaptive Harris corner detection algorithm," *Computer Engineering*, vol. 34, no. 10, pp. 212–214, 2008.
- [10] J. Wang, J. Shi, and X. Wu, "Survey of image mosaics techniques," *Application Research of Computers*, vol. 25, no. 7, pp. 1940–1943, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

