Constraints-Based Graph Embedding Optimal Surveillance-Video Mosaicing

Feng Hu, *Student Member, IEEE*, Tuotuo Li, and Zheng Geng, *Senior Member, IEEE*
Institute of Automation, Chinese Academy of Sciences
{hoofeng09, ltt198612, zheng.geng.08}@gmail.com

***Abstract*—One single surveillance camera usually possesses a local view point, making it incapable of monitoring the whole scene. Even utilizing a pan, tilt, zoom (PTZ) platform, the captured frames are inconvenient for observation due to its discretion in presenting the global area. In this paper, we focus on the video mosaicing problem in surveillance environment, to provide a convenient and accurate panorama of the global site being monitored. Based on the specific requirements in surveillance environment, three constraints are proposed for video mosaicing, which include temporal constraint, spatial constraint and similarity constraint. Furthermore, we cast the problem of video mosaicing into the framework of graph embedding, and solve the optimum mosaic by finding the optimal path in the graph with Dijkstra method. Experiments from both outdoor and indoor applications demonstrate the advantage of our method over previous ones, and show that our method is very adaptable in surveillance applications.**

Keywords- video surveillance; video mosaicing; PTZ camera

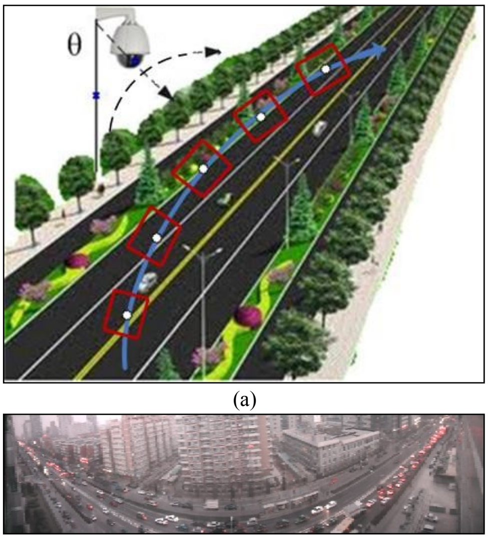
I. INTRODUCTION

Video surveillance systems have been extensively used in public places such as airports, railway stations, subways etc. The captured videos constitute valuable evidence for further analysis. However, current video surveillance systems have certain limits on representing the image of a large coverage of scene due to their narrow field of view (FOV). Therefore multiple cameras are required to monitor the scene together. Fisheye lens or optical attachment may be used to create wild FOV video [1], but the resulted images are distorted and thereby difficult to observe and analyze.

Recently, PTZ camera based video surveillance system has been proposed and successfully applied in a large amount of places. Despite its success, it is inconvenient that the captured frames are discrete representation of the whole scene. Therefore observers have to scan many frames to get a full understanding of the entire monitor area.

To break through this key limitation in PTZ camera surveillance applications, we propose a novel video mosaicing method to provide a complete image representation, which covers the whole scene.

Video mosaicing is the process of creating a panoramic image from video sequences captured from a wide scene [2]. Different from image mosaic problem, video sequences may contain thousands of frames, so the traditional image mosaic algorithms no longer work efficiently when applied directly to the video sequences due to their high computational cost. There are two main classes of methods to solve the video mosaicing problem: key-frame based method and manifold-based method.



(a)

(b)

Figure 1. (a) A simple illustration of an outdoor surveillance application, in which we label the PTZ camera and the changing scene images (red rectangels) captured as the camera rotates. (b) A sample video mosaicing result.

Key-frame based method is comprised of two steps. Firstly, key frames are identified and extracted from the video [3]. Then some image registration methods are applied in stitching these key frames together. Brown et al. [4] and Zitova et al. [5] gave excellent surveys in tackling image registration problem. Recently, there are mainly two kinds of image registration methods. One is feature-based method [6] [7] and the other is camera-parameters-based method [8] [9]. The main problem with feature-based method is that the feature selecting and image alignment process are usually complex, which involves a lot of computing and memory resources and fail to satisfy the real time requirement in surveillance environment. The main problem with camera-parameters based method is that, in PTZ camera systems, camera parameters (e.g. focal length) changes during monitoring process. In addition, these cameras are usually installed in an outdoor environment and hard to calibrate.

The other class of method for video mosaicing is manifold-based method. Peleg et al. [10] create the panorama with video frames by manifold projection, which is comprised of three major steps: image alignment, image cut-and-paste, and image

This Research is supported by the National High technology Research and Development Program (863 program) of China (No. 2007AA01Z338).

978-1-4577-0121-4/11/$26.00 ©2011 IEEE

blending. Correlation operation is applied in their method for every two adjacent frames and patches are selected from the registered frames. This method needs a large amount of calculations and it is not suitable in surveillance situations where real time requirement is important. Wexler et al. [11] considers the whole video as a space-time volume, and the video mosaic problem is then converted to finding a manifold through this volume. Despite of time-consuming, distortion often appears when dealing with the PTZ camera videos, for their general method neglect some specific requirements in the surveillance applications.

In this paper, we propose a novel and effective video mosaicing framework in handing surveillance-video mosaicing problem. Based on characteristics of the surveillance applications, three major constraints, i.e. temporal constraint, spatial constraint and similarity constraint, are for the first time proposed for the process of picking up mosaic patches. Then we construct a graph consisting of all frame strips, with connection controlled by the above three constraints. Finally optimization method is applied to create the panorama by finding the shortest-path in the graph. We test our algorithms on real systems and also compare it with previous works. The results in different surveillance environments show the superior performance of our method.

The rest of paper is organized as follows. In Sect. 2, our video mosaicing algorithm is described in detail. Then experimental results and discussions are presented in Sect. 3. Finally, the conclusion is made in Sect. 4.

II. The Algorithm

In this section, we firstly introduce the surveillance application environment and how the PTZ platform is utilized to capture the source frames. Then we propose our video mosaic algorithm, including both the formulation and the solution procedures.

A. Surveillance application and PTZ camera frames capture

The application we focus on is specifically for surveillance, where the scene images are captured by a PTZ camera system. The environment applications can be simulated as in Fig.1, where we show a surveillance application using a PTZ camera rotating from right to left to cover the whole scene.

From Fig.1, several representative characteristics need to be noticed, which are unique and specific in the surveillance applications: (1) the PTZ camera is set high to observe the scene in a looking-down manner; (2) during one rotation, the camera obeys a fixed direction, e.g. from left to right or otherwise; take "from right to left" case for instance, this specific rotation means that the later captured frames should be shown at a left position in the output mosaic; (3) usually such an application demands that algorithms should achieve real-time effect. For simplicity, in the following sections we always assume the camera rotates from right to left.

B. Scene Constraints

Let $\mathrm{V} = \{\mathrm{v_f}\}, \mathrm{f} = 1{:}\,\mathrm{M}$ be the video sequence captured as in Sect. A, where M is the total number of frames. For each frame, we equally split the frame into K strips horizontally, and therefore, each frame v is a set of K strips, i.e. $\mathrm{v_f} = \{\mathrm{v_f^s}\}, \mathrm{s} = 1{:}\,\mathrm{K}$. Here we set all strips have exactly one pixel width, which makes each strip the frame column. Based on that, wider strips can be formed naturally. Suppose that we have assigned the first and last strip for the final mosaic as ψ_{beg} and ψ_{end}, then our goal is to select a subset of strips $\psi_1, \ldots, \psi_N$, which makes the final mosaic. Suppose the corresponding frames in the original video of these strips are $\mathrm{v}_{\psi_1}, \ldots, \mathrm{v}_{\psi_N}$.

Based on the above requirements in Sect. A and the nature of mosaicing, we propose the following constraints on these selected strips:

1) Temporal Constraint. The temporal relationship of original strips in each frame should be preserved in the final mosaic.

For instance, suppose v_{ψ_i} and v_{ψ_j} are two selected frames from the video, where v_{ψ_j} is captured after v_{ψ_i}. Then in the final mosaic, ψ_j should come after ψ_i.

Furthermore, if ψ_i and ψ_j are two adjacent strips in the final mosaic, v_{ψ_i} and v_{ψ_j} should not differ too many frames in original videos.

2) Spatial Constraint. For two adjacent strips ψ_i *and* ψ_j, their original frames in video should not differ too much spatially.

For instance, if ψ_i and ψ_j are adjacent in the final mosaic, the columns which v_{ψ_i} and v_{ψ_j} represent should not differ too much spatially.

Furthermore, if ψ_j is after ψ_i , the column of v_{ψ_j} should be no right than that of v_{ψ_i} , when the rotation speed of the PTZ camera is not too fast.

3) Similarity Constraint. The adjacent strips in mosaic should be similar; otherwise, distortions will be introduced.

For instance, if ψ_i and ψ_j are adjacent, their similarity should be high. So the final panorama can transit smoothly as from one side to the other.

Based on the above constraints, we now can formulate the proposed video mosaicing algorithm.

C. Algorithm Formulation

We formulate the video mosaicing problem as the following optimization problem:

$$[\psi_1, \ldots, \psi_N] = \operatorname{argmin} \sum_{i=1}^{N-1} \left|\left|\psi_i - \psi_{i+1}\right|\right|^2 + ||\psi_{\mathrm{beg}} - \psi_1||^2 + ||\psi_N - \psi_{\mathrm{end}}||^2 \qquad (1)$$

where $[\psi_1, \ldots, \psi_N]$ are the final strips in mosaic, and $\psi_{\mathrm{beg}}, \psi_{\mathrm{end}}$ are the beginning and ending strips respectively, which are assigned by the user.

The objective in optimization function (1) forces the similarity smoothness in the final strips, which obeys the similarity constraint in Sect. B. To further make the mosaic spatial and temporal constrained, two more constraints are added as follows:

$$0 \le v_{\psi_i} - v_{\psi_{i+1}} \le F_c$$

$$0 \le C_{\psi_i} - C_{\psi_{i+1}} \le C_c \qquad (2)$$

where v_{ψ_i} indicates which frame ψ_i belongs to (regardless of column position), and C_{ψ_i} indicates which column ψ_i belongs to (regardless of frame position). By setting the upper bound F_c and C_c of adjacent strips, the two constraints force the spatial and temporal relationship preserved among the mosaic strips.

We then cast the video mosaicing problem into a graph embedding framework for solution. Let $G = (V, E)$ be a graph representing the video frames. The set of vertices V is consisted of all strips in all frames within the video, namely $V = \{v_f^s\}, f = 1{:}M, s = 1{:}K$. The edge E represents the connection between the strips. We construct the edges in the following way, obeying the above constraints:

$$E_{i,j} = \begin{cases} \text{score} & \text{if } i, j \text{ satisfy the constraints} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

The above equation connects node strip i and j only if the corresponding strips satisfy the temporal and spatial constraint as defined above, where we set $F_c = 3$ and $C_C = 2$ in this paper. In the above construction of E, the score means the similarity distance of the node i and j, which serves as the edge weight in graph G. We define the similarity between strips X and Y using the Pearson correlation coefficient.

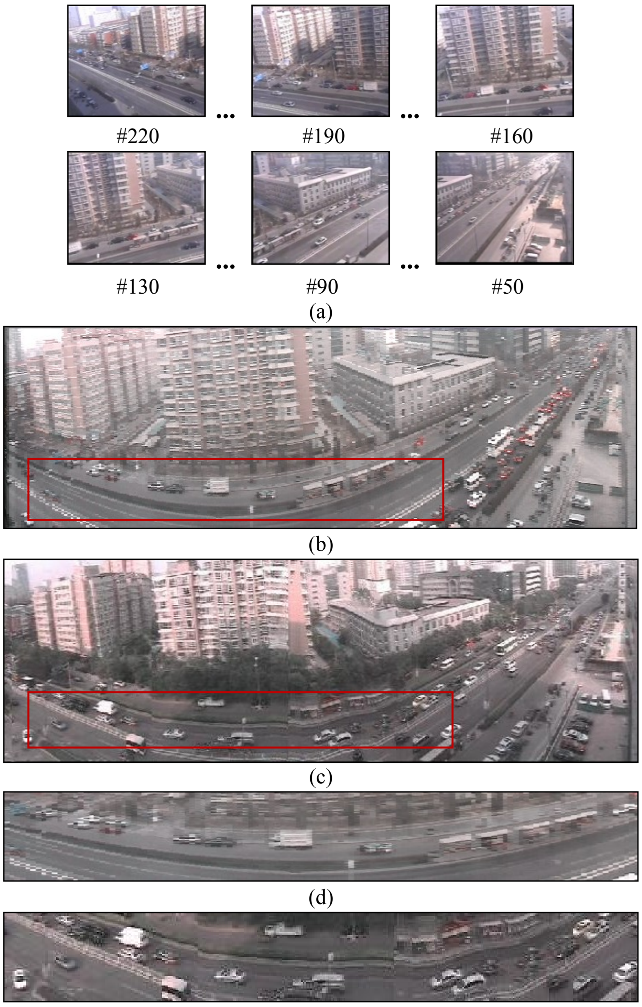
$$\text{score} = \frac{1}{T-1}\sum_{i=1}^{T}\left(\frac{X_i-\bar{X}}{s_X}\right)\left(\frac{Y_i-\bar{Y}}{s_Y}\right) \qquad (4)$$

T is the number of rows in one strip, and X_i, Y_i are the intensity value of row i. $\bar{X}$ and $\bar{Y}$ are the mean values of these two strips, and s_X, s_Y are the standard deviations of X and Y respectively.

After the above construction, we now have a graph with its nodes indicating the frame strips, and edges indicating the similarity between nodes. We assign the beginning node as the first strip in the first frame, and the ending node as the last strip in the last frame. Then the mosaic optimization objective now becomes the problem of finding the shortest-path from the beginning node to the ending node in the graph. Along that shortest-path, the nodes, or strips will form the final mosaic, which obeys all the spatial, temporal and similarity constraints as we defined above. And the final shortest-path guarantees a global minimum among all possible paths, which makes the objective function achieve its optimum. In this paper we utilize the Dijkstra method [13] to get an efficient solution.

III. EXPERIMENS

Our algorithm is implemented in C++ using MFC with Visual Studio 2010, and has been embedded into the real surveillance system, Smart PTZ. Our hardware configuration is a common PC with 2GHz CPU and 2G memories.



(a)

(b)

(c)

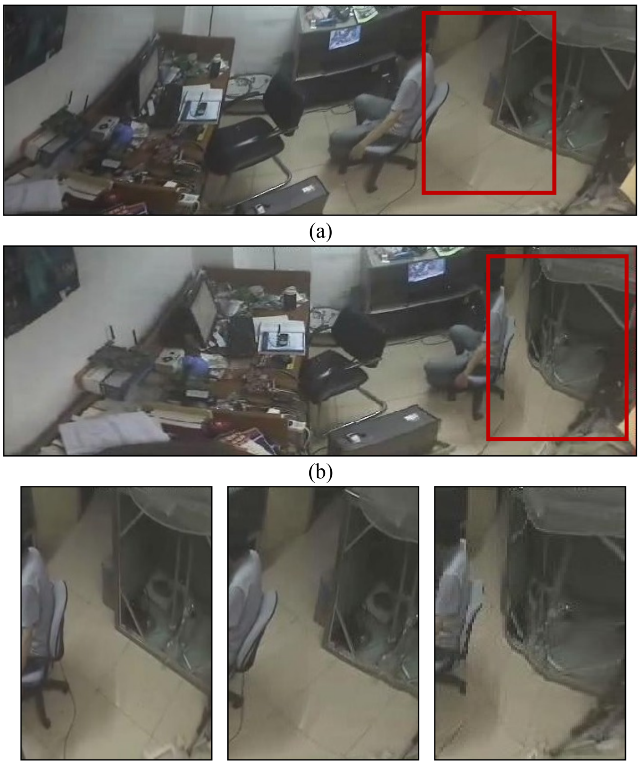
(d)

(e)

Figure 2. (a) Some sample images among the 250 frames captured by an outdoor PTZ camera. (b) Result of our video mosaicing algorithm. (c) Result of [11]'s algorithm. Some details marked with red rectangles in (b) and (c) are respectively zoomed in and shown in (d) and (e). (d) is our result and (e) is [11]'s.

In our first experiment, an outdoor PTZ camera monitoring a road and its surrounding is used to capture video sequences. Pan direction is from right to left. The resolution of original frame is 704*576 pixels. 250 frames are collected when the camera covers about 180 degree FOV in horizontal. Fig.2 (a) shows some sample frames. Frames are stored while the PTZ camera pans across to cover the whole scene. Our system takes 62 ms to form a desired panorama, which is fast enough to guarantee a real-time use.

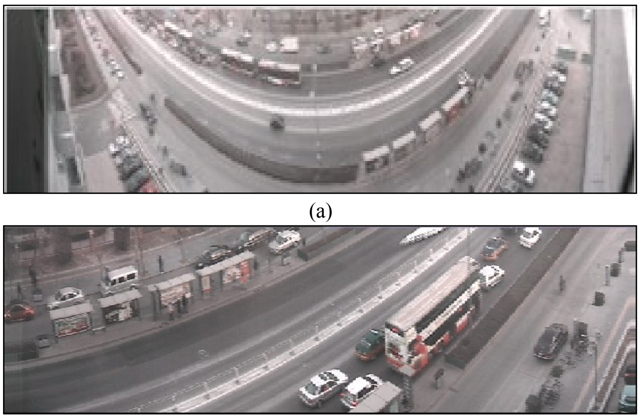
We have compared our work with [11]'s, whose source code can be easily downloaded at: http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeSceneManifolds/. Fig.2 (c) shows their result. We zoom in the details contained in red rectangle in Fig. 2 (b) and Fig. 2 (c) and present them in Fig.2 (d) and Fig.2 (e) respectively. Due to the lack of constraints existed uniquely in surveillance environment and effective similarity measure method, our result performs well and has less disorders and distortion than others', as shown in Fig.2 (d) and Fig.2 (e).


(a)

(b)

(c) (d) (e)

Figure 3. (a) Our video mosaicing result with indoor environment. (b) Result of [11]'s. (c) Ground truth image of the red rectangle region.. (d) Our result of this region. (e) [11]'s result of this region.


(a)

(b)

Figure 4. (a) A video mosaicing result with middle size zoom scale. (b) A vidoe mosaicing result with large zoom scale.

In the second experiment, we employ our algorithm in indoor surveillance circumstance. Fig. 3 (a) shows our video mosaicing result and Fig. 3 (b) shows that of previous work's [11].The partial images contained in red rectangles are zoomed in and displayed in Fig. 3 (d) and (e) respectively. Fig. 3 (c) is the ground truth image.

Our algorithm is not only suitable when zoom value is small, but also capable of handling other zoom value settings of the same scene in surveillance environment. Fig. 4 (a) and (b) illustrate our video mosaicing results with various zoom values.

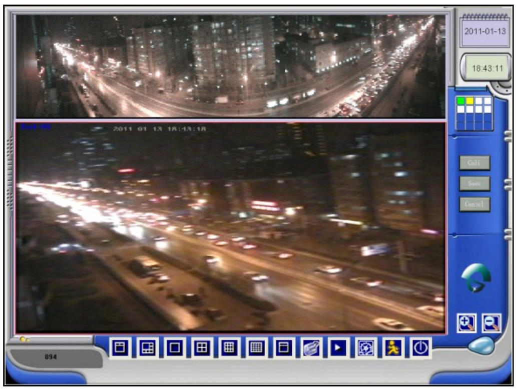


Figure 5. A video mosaicing result at night in our System. The upper window shows a panorama and the lower window shows the real-time frame captured by the PTZ camera.

Our algorithm is also insensitive to illumination change and can robustly work in both day and night. Fig.5 illustrates one result when handling night surveillance video. The upper window shows a final panorama, and the low window shows real time PTZ camera video.

Other wide FOV scene video mosaicing results are as well as shown in Fig.6. In these scenes, cameras are rotated from right to left and frames are captured during these processes. Despite of various lighting conditions: back lighting, front lighting or side lighting, our algorithm works robustly and accurately.

IV. CONCLUSION

In this paper, a novel and efficient video mosaic framework is proposed to handle surveillance videos. Based on characteristics of the surveillance applications, we propose three major constraints, which include temporal constraint, spatial constraint and similarity constraint. Then we construct a graph consisting of all frame strips, with connection controlled by the above three constraints. Finally an optimization method is applied to create the panorama by finding the shortest-path in the graph. The results in different surveillance environment demonstrate the advantages of our algorithm.

In the future, we plan to detect and track moving objects using panoramas as backgrounds and analyze objects' behaviors. Video synopsis with a moving PTZ camera is also challenging and interesting.

V. ACKNOWLEDGEMENT

The authors want to thank Zhang Zhiwei, Bai Yancheng, Wang Xiaoguo, Zhang hongyan and others for their help in this project. The authors also want to thank the anonymous reviews for their insightful comments and suggestions.

This project is also supported by National Natural Science Foundation of China (Grant Numbers: NNSFC 60921061, NNSFC 90920305).

REFERENCES

[1] G. Krishnan, S. K. Nayar, "Cata-Fisheye camera for panoramic imaging", Proceedings of the 2008 IEEE Workshop on Applications of Computer Vision, Copper Mountain, CO , Jan. 2008, pp. 1-8.

[2] M. Irani, P. Anandan, S. Hsu, “Mosaic based representations of video sequences and their applications”, Proceedings of the Fifth International Conference on Computer Vision, June 20-23 , 1995, pp. 605.

[3] D. Steedly, C. Pal, R. Szeliski, “Efficiently registering video into panoramic mosaics”, Tenth IEEE International Conference on Computer Vision (ICCV'05) , Beijing , 2005, Volume 2.

[4] L. G. Brown, "A survey of image registration techniques", ACM Computing Surveys, vol. 24, no. 4, 1992, pp.325 - 376.

[5] B. Zitova, J. Flusser, "Image registration methods: A survey," Image and Vision Computing, vol. 21, October 2003, pp. 997-1000.

[6] M. Brown, D. G. Lowe, “Automatic panoramic image stitching using invariant features”, International Journal of Computer Vision, vol. 74, 2007, pp. 59-73

[7] M. Brown and D. Lowe, “Recognising panoramas,” Proc. Ninth Int'l Conf. Computer Vision, 2003, pp. 1218-1227.

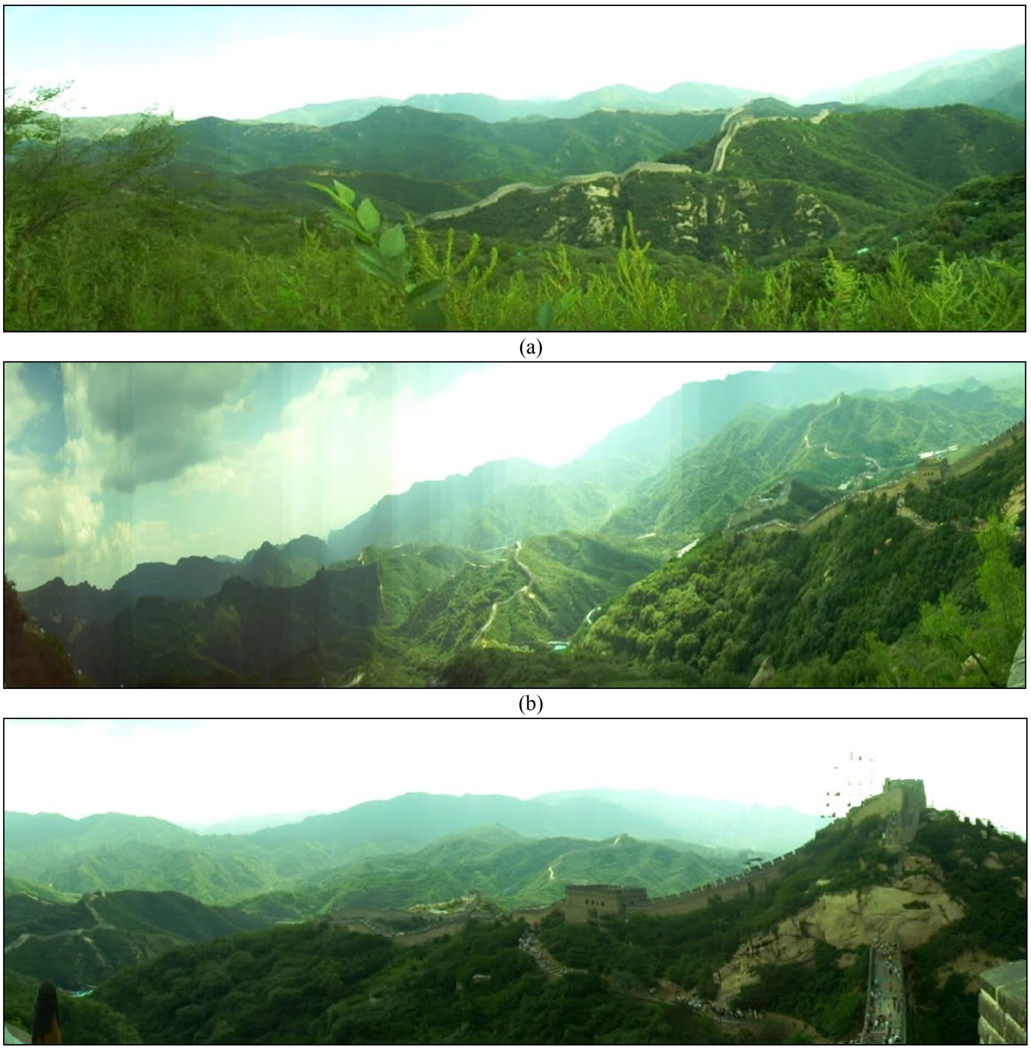
[8] R. Szeliski, H. Y. Shum, “Creating full view panoramic image mosaics and environment maps”, Proceedings of the 24th annual conference on Computer graphics and interactive techniques, August 1997, pp.251-258.

[9] H.Y. Shum, R. Szeliski. “Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment”, International Journal of Computer Vision, 2000, pp.101–130.

[10] S. Peleg, B. Rousso, and A. Rav-Acha, A. Zomet, “Mosaicing on adaptive manifolds”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, pp.1144–1154.

[11] Y. Wexler, D. Simakov, “Space-time scene manifolds”, In Tenth IEEE International Conference on Computer Vision, volume 1, 2005, pp. 858–863.

[12] E. W. Dijkstra, “A note on two problems in connection with graphs.” Numerische Math. Vol 1, 1959, pp. 269-271.



(a)

(b)

(c)

Figure 6. More examples of video mosaicing results with different lighting conditions: (a) front lighting, (b) back lighting, (c) side lighting using our method.