2009 Fifth International Conference on Image and Graphics

Foreground based Borderline Adjusting for Real Time Multi-Camera Video Stitching

Hongming Zhang, Wei Zeng
NEC Laboratories, China
11F, Bldg.A, Innovation Plaza, Tsinghua Science Park
Beijing, China, 10084
{zhanghongming,zengwei}@research.nec.com.cn

Xin Chen
Beijing Jiaotong University
No.3. Shangyuan Residence, Haidian District
Beijing, China, 100044
alben2008@hotmail.com

***Abstract*—In this paper, we propose a multi-camera video stitching approach, which conducts borderline adjusting based on foreground information. In real time applications of video stitching, one problem is to deal with dynamic scenes, where foreground objects often cause broken objects like artifacts in panoramic video. To address this problem, we propose a foreground based borderline adjusting method to achieve smooth video stitching. In this method, foreground objects are extracted from different viewpoints, and the stitching borderlines are adjusted according to the foreground content in dynamic scenes. Guided by the adjusted borderlines, foreground objects are smoothly synthesized into panoramic video. Experiment results show that this approach obtains more than 81% correct rate for dynamic scenes, compared with 25%~50% correct rate on the condition of not utilizing foreground information, and achieves processing speed of 10~13 frames per second.**

Keywords-Foreground Objects; Borderline Adjusting; Video Stitching; Multi-Camera

I. INTRODUCTION

Given some source image sequences with limited overlapped regions, video stitching is the process to merge the input image sequences into one high resolution panoramic video with wide view field. Video Stitching has a wide range of applications, such as visual surveillance, human machine interaction, video processing and editing, and remote video conference.

Video stitching methods can be categorized from two standpoints according to the application cases of video stitching. The first standpoint is to apply video stitching approaches by manipulating source image sequence from single camera or multiple cameras, and the second standpoint is to utilize video stitching systems for offline processing or online processing.

Single-camera video stitching technologies have been developed for various applications. Szeliski [1] develops an image-based video stitching method that can be used for tele-presence applications. Sato et al. [2] propose a method that can create high resolution video mosaics from a mobile camera. Sand and Teller [3] present an approach that finds best matching frames in different videos captured by a moving camera, and use this approach to create wide field-of-view video. The work in [4] creates dynamic panoramas with events simultaneously occurred for videos taken by a static camera. Compared with the single-camera based video stitching methods, the topic of multi-camera video stitching recently has gained attention. An example is Stanford Multi-Camera Array project [5]. The system uses more than 100 inexpensive cameras to obtain videos of large view filed. Haenselmann et al. [6] present a multi-camera video stitching method, and investigate the side effects of perspectives for moving objects.

For offline applications, various kinds of image stitching methods [7] can be used in video stitching tasks. Brown et al. [8] employ SIFT features based image alignment to merge images into one mosaic. Uyttendaele et al. [9] propose methods to eliminate ghost and exposure artifacts in mosaics. Jia et al. [10] describe an apporach using structure deformation to achieve seamless image mosaic. In real applications, the video stitching task usually requires online processing speed. In [11], Pan et al. present a method based on pre-calculated geometry to speed up the video stitching algorithms. This algorithm is suitable to scene with unchanged depth. Zheng et al. [12] use two webcams to create wide field-of-view videos in real time, by using a nonlinear blending method.

The target of this paper is online multi-camera video stitching for dynamic environments. In the real applications of creating panorama video in real time, a major problem is that broken object like artifacts usually happen in video stitching results for dynamic scenes where moving objects appear. To address this issue, we propose a foreground based borderline adjusting method for multi-camera video stitching. The basic idea is that we adjust stitching borderlines according to the foreground content change in scenes, and use the optimized borderlines to smoothly synthesize foreground objects into panoramic video.

Unlike image-based video stitching methods and video stitching technology based pre-calculated geometry relationship of cameras, our approach has the capability to reduce artifacts since it keep consistency among foreground objects and background in dynamic scenes. Furthermore, our approach focus on minimizing quality depredation of foreground objects, because the quality of foreground objects is critical in succeeding applications such as object tracking and face recognition. Extensive experiments on indoor and outdoor scenarios are conducted to evaluate the effectiveness of the proposed multi-camera video stitching method.

Our method performs calibration for multiple cameras initially, and conducts video stitching in real time. The rest of this paper is organized as follows. Section 2 outlines the proposed approach. Section 3 describes the calibration method. In Section 4, methods of foreground based borderline adjusting are presented. Section 5 gives experiments results, and Section 6 concludes this paper.

978-0-7695-3883-9/09 $26.00 © 2009 IEEE
DOI 10.1109/ICIG.2009.42

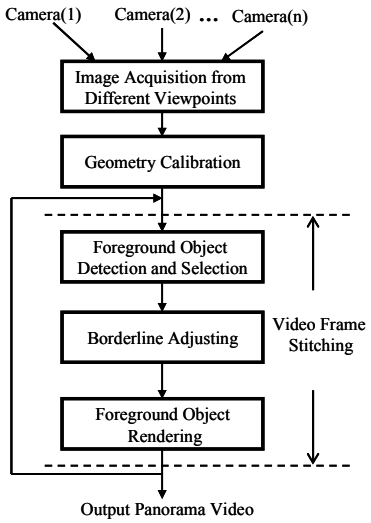


Figure 1. Process of the proposed multi-camera video stitching approach

II. Overview of the Proposed Approach

The proposed approach is designed to merge videos from different viewpoints into one high resolution video in real time, and to achieve its robustness to foreground objects in dynamic scenes. As illustrated in Fig.1, the system contains three parts: image acquisition, geometry calibration, and video frame stitching. The purpose of geometry calibration part is to obtain the image transforms and stitching borderlines among input videos. Video frame stitching part is the core component of our method. In this part, panorama video is created in a frame-by-frame mode. For each frame, foreground objects are extracted and employed to adjust borderlines, consequently foregrounds are synthesized with background to render video frame.

III. Geometry Calibration

In the geometry calibration part, image transforms and borderlines among cameras are computed. Image transforms play an elemental role to describe the geometry relationship of adjacent viewpoints. Stitching borderlines are separating lines between adjacent input videos in panoramic video, and initially are determined by image transforms in the calibration stage.

There are many image registration approaches to obtain image transforms [7, 13]. We adopt one SIFT features based method [8] to compute image transforms. Given any pair of images I_1 and I_2 from two viewpoints, which are grabbed from two cameras with adjacent viewpoints, the image transform of them can be written as:

$$X_2 = HX_1 \quad (1)$$

where X_1 and X_2 are the homogenous coordinates of points in image plane I_1 and I_2, and H is a 3×3 matrix. After computing the image transforms information among multiple cameras, we employ a distance-based method [13] to determine initial stitching borderlines. Fig.2 shows examples of obtained image transform and borderline for stitching.

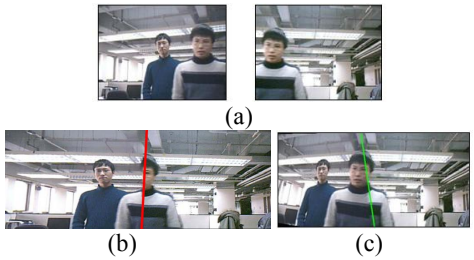


Figure 2. Example of video stitching for dynamic scenes. (a) Input images containing objects, (b) and (c) are stitched images with borderlines obtained using background information and foreground respectively

In the real time stitching stage, it is straightforward to use the image transforms and borderlines to combine input image sequence to one panorama video. However, this kind of method has one drawback that artifacts occur due to content changes in dynamic scenes. Video contents of a scene are divided as two types: foreground objects and background. Foregrounds refer to moving contents appear in dynamic scenes, such as persons walking in office rooms, cars running in streets. Background region is the area of the scene behind foreground objects. As shown in the examples of Fig.2, artifacts often occur when foreground objects appear. This phenomenon is related to the parallax problem resulted from difference between scene depths of background and foreground.

IV. Foreground Based Borderline Adjusting

In order to eliminate artifacts caused by foreground objects in synthesized panorama video frames, our method emphasizes on incorporating foreground information during the stage of multi-camera video stitching. As mentioned above, our method first extracts foreground objects from the input videos of different viewpoints, then performs borderline adjusting according to the extracted foreground objects, and finally renders foreground objects into the panorama video by using the adjusted borderlines.

A. Foreground Object Detection

A lot of methods can be used to detect foreground objects. In this paper, we adopt one motion detection method [14] for its stability and computational inexpensiveness. It performs background subtraction to extract moving objects as foreground. A foreground detection example is shown in Fig.3. For each object region, its shape is denoted as one polygonal rectangle $R(x_{lt}, y_{lt}, x_{rb}, y_{rb})$, where its left top point is (x_{lt}, y_{lt}), and (x_{rb}, y_{rb}) is its right bottom point.

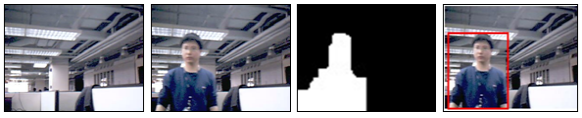


Figure 3. Example of foreground detection. (a) One scene image, (b) One image with moving object, (c) Moving detection result of(b),(d) Detected foreground region

B. *Foreground Object Selection*

After obtaining the foreground objects in input images, the next step is to determine which foreground object is used to adjust borderlines. The key issue of foreground object selection is to determine the relationship among foreground object regions. In our approach, we employ spatial information and texture information to describe relationship among foreground object regions, and use the relationship information to select suitable foreground objects.

Given any pair of images L and R with overlapped coverage, each has foreground object sets respectively: $L_{Set} = \{L_1, L_2, ..., L_m\}$ and $R_{set} = \{R_1, R_2, ..., R_n\}$, where m and n are the numbers of objects in each set. The selection of foreground is described as follow:

(1) **Initialization**

$$L_{Select} = \varnothing, R_{Select} = \varnothing, OverlapSet = \varnothing.$$

(2) **Region transform**

The coordinates of all foreground object regions are transformed into the panorama video plane, according to the Equation (1).

(3) **Spatial relation decision**

(a) For any region $L_i \in L_{Set}$, find its overlapped region from R_{set}.

- If the overlapped region is R_j, (L_i, R_j) is added into $OverlapSet$: $OverlapSet = OverlapSet \cup \{(L_i, R_j)\}$.
- If there is no overlapped region for L_i, L_i is added into L_{Select}, $L_{Select} = L_{Select} \cup \{L_i\}$.

(b) For any region $R_i \in R_{Set}$, similar process is performed to update R_{Select} and $OverlapSet$.

(4) **Texture similarity checking**

For each region pair (L_i, R_j), compute the similarity $H(L_i, R_j)$ between L_i and R_j using texture similarity measurement. If $H(L_i, R_j) \le T$, (L_i, R_j) is deleted from the $OverlapSet$:

$$OverlapSet = OverlapSet \setminus \{(L_i, R_j)\}.$$

Here, T is a predefined threshold to measure the texture similarity of two foreground regions.

(5) **Overlapped region selection**

For any overlapped region pair (L_i, R_j) in the $OverlapSet$, foreground selection is performed. If the area of L_i is greater than the area of R_j, L_i is added into L_{Select}, $L_{Select} = L_{Select} \cup \{L_i\}$; otherwise R_j is selected, $R_{Select} = R_{Select} \cup \{R_j\}$.

The above foreground object selection procedure can be summarized as two rules. One rule is that if one foreground object in one input image is not overlapped with any foreground object in other input images, it is selected as the foreground object to be embedded in panorama video. Another rule is that if one overlapped foreground object pair has high texture similarity score, the object with greater area is selected to be embedded.

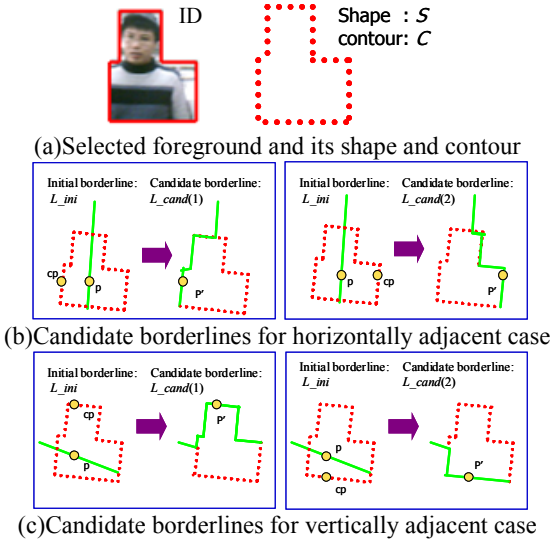


Figure 4. Candidate borderlines generation

C. *Borderline Adjusting Method*

Given foreground objects selected from different viewpoints, the key issue is to utilize foreground information to adjust borderlines for the purpose of keeping the integrity of foregrounds and the consistency between foregrounds and background contents. In our method, the shape and contour information of foreground objects is employed for borderline adjusting, which can be obtained by refining the motion detection results [14].

The borderline adjusting method is performed in two steps. The first step is to generate candidate borderlines according to the shape and contour information of foreground. The second step is to select one borderline from the candidate borderlines based on fusion image quality measurement.

In the first step, candidate borderlines are generated according to the principal direction of the initial borderline, as shown in Fig.4. Suppose that for two overlapped images I_1 and I_2, the selected foreground object is ID and its shape and contour information is denoted as (S, C). Shape is the area that is covered by contour C. They are wrapped into panorama video. The initial borderline is L_ini, and the candidate borderline is represented as L_cand. For any point $p(x, y)$ in L_ini, the corresponding point in L_cand is $p'(x, y)$. If I_1 and I_2 are horizontally neighboring and the principal direction of L_ini is vertical, there are two candidate borderlines $L\_cand(1)$ and $L\_cand(2)$, and the position of $p'(x, y)$ is determined as below:

(1) $L\_cand(1)$:

$$\begin{cases} \text{Y position}: p'(y) = p(y) \\ \text{X position}: p'(x) = \begin{cases} cp(x) & \text{if } p(x,y) \in S \\ p(x) & \text{otherwise} \end{cases} \end{cases}, \quad (2)$$

where $cp \in C$, and $cp(y) = p(y)$, $cp(x) < p(x)$.

(2) $L\_cand(2)$:

$$\begin{cases} \text{Y position}: p'(y) = p(y) \\ \text{X position}: p'(x) = \begin{cases} cp(x) & \text{if } p(x,y) \in S \\ p(x) & \text{otherwise} \end{cases} \end{cases}, \quad (3)$$

where $cp \in C$, and $cp(y) = p(y)$, $cp(x) > p(x)$. If the images I_1 and I_2 are vertically neighboring and the principal direction of L_ini is horizontal, similar operation is performed to obtain two candidate borderlines $L\_cand(1)$ and $L\_cand(2)$.

In the second step, the candidate borderlines are evaluated by using the image quality index [15, 16]. Given two gray images a and b, the image quality index [16] is computed as

$$Q_0(a,b) = \frac{4\sigma_{ab}\bar{a}\bar{b}}{(\bar{a}^2 + \bar{b}^2)(\sigma_a^2 + \sigma_b^2)}, \quad (4)$$

where $\bar{a}$ and $\bar{b}$ are the mean of a and b, σ_a^2 and σ_b^2 are the variance of a and b, σ_{ab} is the covariance of a and b. In fact, the value of Q_0 is a similarity measure of images a and b [16]. It achieves the maximum value $Q_0 = 1$ when a and b are identical. For one candidate borderline L_cand for I_1 and I_2, its fusion image measurement is computed by sliding one sub window along each point in L_cand,

$$Q_{L\_cand} = \frac{1}{N} \sum_{P \in L\_cand} Q_0(I_1, I_2 \mid w(P)), \quad (5)$$

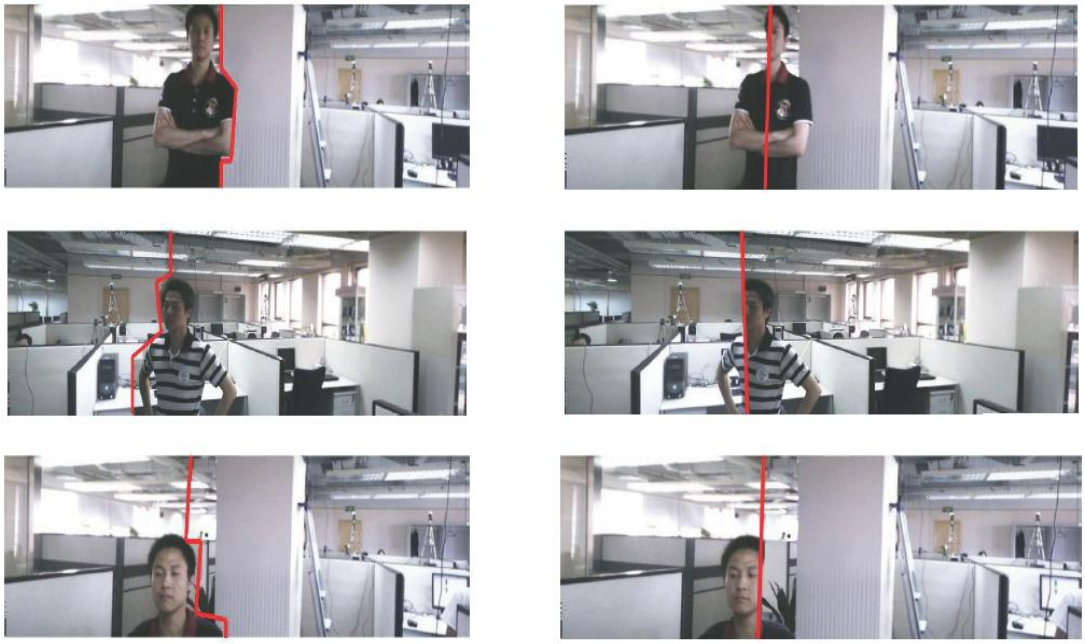
where $w(P)$ is the sliding window centered at point P, and N is the number of points in L_cand. The adjusted borderline L_adj is selected from the candidate borderline as the one which has maximum fusion image measurement value, i.e.

$$L\_adj = \begin{cases} L\_cand(1) & \text{if } Q_{L\_cand(1)} > Q_{L\_cand(2)} \\ L\_cand(2) & \text{otherwise} \end{cases}. \quad (6)$$

Using the adjusted borderlines, the foreground objects are smoothly synthesized into panorama video frames.

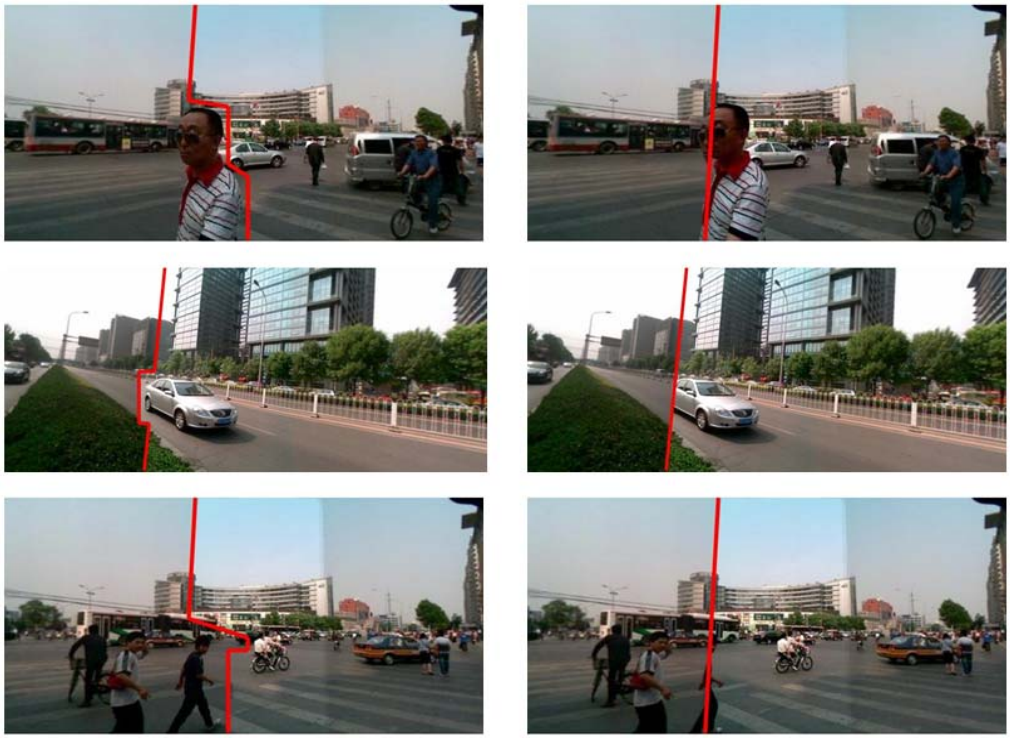
V. EXPERIMENTS

We conduct experiments on real environments including indoor scenes and outdoor scenes to evaluate the proposed approach. In Fig.5 and Fig.6, some video stitching results are given. The part (a) demonstrates video stitching result of our borderline adjusting approach, and part (b) shows that of not using borderline adjusting. As illustrated in part (a), dominant foreground objects are covered by the adjusted borderlines. Our approach can eliminate artifacts caused by foreground and can handle foreground objects along different depths of field.



(a)Using borderline adjusting (b)Not using borderline adjusting

Figure 5. Stitching result examples of indoor scenes



(a)Using borderline adjusting (b)Not using borderline adjusting

Figure 6. Stitching result examples of outdoor scenes

We have collected a video dataset, which contains two video samples for indoor and outdoor scenarios respectively. The indoor video sample is collected by using three cameras to grab one office room scene where one person is moving, and it contains 710 frames for each viewpoint. The outdoor video sample records one street where multiple cars and pedestrians are moving. It contains 492 frames for each viewpoint.

For each synthesized panorama frame, we propose Correct Frame Index (CFI) to measure rendering quality of foreground objects, and CFI is determined as

$$CFI = \begin{cases} 1 & \text{if the frame contains no broken objects} \\ 0 & \text{otherwise} \end{cases} . \quad (7)$$

We regard CFI as indication of correctly synthesizing, and compute Correct Rate (CR) to measure video stitching results, i.e.

$$CR = \frac{1}{N}\sum_{t=1}^{N} CFI(t) , \quad (8)$$

where N is the number of frames in panorama video.

On the collected dataset, our system correctly synthesizes 580 panorama video frames and 443 panorama video frames for the indoor and outdoor video samples respectively. The testing result is listed in Table I and Table II. The correct rate of using borderline adjusting method is higher than that of not using borderline adjusting. These experiment results prove that the proposed approach is effective to reduce artifacts in multi-camera video stitching process.

We implement the proposed method on one PC with Intel Core 2 CPU of 1.86GHZ and 2 GB memory. On the condition of 320x240 source videos from three cameras, our method proceeds at 10 frames per second. On the condition of operating on 320x240 input image sequences from two cameras, the running speed is 13 frames per second. These results show our method is efficient for real time processing.

In Fig.7, some error video stitching results are given. Some foreground objects with large sizes are broken in source images, so they are not correctly synthesized. In some examples, rendering errors are caused by incorrect moving detection, since borderline adjusting process is guided by moving detection result. How to overcome these limitations is our future work.

TABLE I. TESTING RESULT (INDOOR, 710 FRAMES)

	Correct Frames	Correct Rate
Using borderline adjusting	580	81.6%
Not using borderline adjusting	173	24.3%

TABLE II. TESTING RESULT (OUTDOOR, 492 FRAMES)

	Correct Frames	Correct Rate
Using borderline adjusting	443	90.0%
Not using borderline adjusting	242	49.2%

Figure 7. Some error examples of stitched frames

VI. CONCLUSIONS

In this paper, we present a foreground based borderline adjusting approach for multi-camera video stitching in dynamic scenes. This method extracts foreground object regions and performs borderline adjusting to keep foreground objects and background contents to be consistent in the video stitching result. Experiment results show that the proposed approach is effective in eliminating artifacts caused by foreground objects, and is efficient for real time processing.

REFERENCES

[1] R.Szeliski. Image Mosaicing for Tele-Reality Applications. IEEE workshop on Applications of Computer Vision, 230-236, 1994.

[2] T.Sato, S.Ikeda, M. Kanbara, A.Iketani, N.Nakajima, N. Yokoya, K.Yamada. High-resolution Video Mosaicing for Documents and Photos by Estimating Camera Motion. Proceedings of SPIE, vol.5299, 246-253,2004.

[3] P.Sand,S.Teller. Video Matching. ACM Transactions on Graphics, 23(3), 592-599.2004.

[4] A.R.Acha, Y.Pritch, D.Lischinski, S.Peleg. Dynamosaicing: Mosaicing of Dynamic Scenes. IEEE Trans.on PAMI, 29(10), 1789-1801.2007.

[5] B.Wilburn,N.Joshi,V.Vaish,E.Antunez,A.Barth,A.Adams, M.Horowitz, M.Levoy. High Performance Imaging Using Large Camera Arrays.In ACM Transactions on Graphics, 24(3), 765-776,2005.

[6] T.Haenselmann, M.Busse, S.Kopf, T.King, W. Effelsberg. Multi-Camera Video Stitching. In Proceeding of Content generation and coding for 3D-television, High Tech Campus, Eindhoven, The Netherlands, June 2006.

[7] R.Szelisk, Image Alignment and Stitching: A Tutorial, Foundations and Trends in Computer Graphics and Vision, 2(1), 1-104, 2006.

[8] M.Brown, D.G. Lowe. Automatic Panoramic Image Stitching using Invariant Features. IJCV, 74(1), 59-73, 2007.

[9] M.Uyttendaele,A.Eden,R.Szeliski. Eliminating Ghosting and Exposure Artifacts in Image Mosaics. CVPR2001, vol.2, 509-516, 2001.

[10] J.Jia,C.K.Tang.Image Stitching Using Structure Deformation. IEEE Trans.on. PAMI, 30(4), 617-631.2008.

[11] P. Pan, T.Mitsushita, C.Lin, B.Kuo. Optimized Video Stitching Method. US Patent Application Publication, No. US 2007/0211934A1, Sep. 2007.

[12] M.Zheng,X.Chen,L.Guo. Stitching Video from Webcams. Proceedings of the 4th International Symposium on Advances in Visual Computing, Vol(2),420-429,2008

[13] A. Ardeshir Goshtasby. 2-D and 3-D image registration: for medical, remote sensing, and industrial applications. JohmWiley & Sons, Inc.2005

[14] I. Haritaoglu, D. Harwood, L.S. Davis. W4: Real-time surveillance of people and their activities. IEEE Trans.on PAMI , 22(8), 809-830, 2000.

[15] Z.Wang, Bovik, A.C. A Universal Image Quality Index. IEEE Signal Processing Letters, 9(3), 81-84, 2002

[16] G. Piella, H. J. A. M. Heijmans: A new quality metric for image fusion. ICIP 2003, Vol. 3, 173-176, 2003.