

# Automatic Panorama Generation from a Video with Dynamic Background

Kumara W. G. C. W.<sup>#1</sup>, Shih-Ming Chang<sup>\*2</sup>, Timothy K. Shih<sup>#3</sup>

<sup>#</sup>Department of Computer Science and Information Engineering, National Central University  
No.300, Zhongda Rd., Zhongli City, Taoyuan County 32001, Taiwan (R.O.C.)

<sup>\*</sup>Department of Computer Science and Information Engineering, Tamkang University  
No. 151, Yingzhuān Rd, Danshui District, New Taipei City, 25137, Taiwan (R.O.C.)

<sup>1</sup>chinthakawk@gmail.com, <sup>2</sup>rest306@hotmail.com, <sup>3</sup>timothykshih@gmail.com

**Abstract**— Panorama photos are very useful when capturing large scenic backgrounds like famous constructions or eye catching landscapes. Users normally prefer capturing such backgrounds with them or friends as the foreground, but the moving people in such popular locations always obstruct the iconic structure in the background. In this paper a solution for automatic panorama generation is presented which is capable of removing moving objects in the background. First user captures a short video starting from the centre of the focus area and then following a clockwise circular path. The proposed application generates the panorama based on input video following the steps: segmentation of human object, removal of moving human objects, and generation of panorama. Experimental output panorama photos show that the proposed system is very usable and results are satisfactory.

**Keywords**— Automatic Panorama Creation, Image Inpainting

## I. INTRODUCTION

Generation of the panorama from set of photos is being a useful and attractive research topic within the researches in the domain for several years. Kekre, H. *et al.* [1] presented a panorama generation approach to nullify effect of rotation of partial images on process of vista creation. Their method is capable of resolving the missing region in the vista caused due to the rotation of partial image parts used during the vista creation. Image inpainting has been used during the process to fill the missing region. That missing view regeneration method was also able to overcome the problem of missing view in vista due to cropping, irregular boundaries of partial image parts and errors in digitization. Meng and Wang [2] presented an approach to create a single view point full view panorama photograph from a set of image sequence. Individual ordered frames extracted from a panning video sequence have used as the input making it simple for both shooting and stitching.

Going forward another step of panorama generation Wagner, Daniel *et al.* [3] presented a method for the real-time creation and tracking of panoramic maps on mobile phones. Specially, the maps generated are accurate and allow drift-free rotation tracking. Song, Baosen *et al.* [4] then presented another panorama generation based research to enlarge the horizontal and vertical angles of view for an image. Camera calibration, image feature extraction, image registration, bundle adjustment, photometric optimization, image fusion and output panorama projection has been used during the design. They presented that their application can stitch together multi-row images automatically into a panorama. A smoothly varying affine stitching field which is flexible enough to handle parallax while retaining the good extrapolation and occlusion handling properties of parametric transforms was presented by Lin, Wen-Yan *et al.* in [5]. Their algorithm which jointly estimates

both the stitching field and correspondence permits the stitching of general motion source images, provided the scenes do not contain abrupt protrusions.

However, all the recent researches on panorama generation focused only on the generation of the panorama images using the set of input images. But users' real needs on panorama generation are more complex. Generally panorama images are very useful to capture scenic locations or large buildings, where naturally crowded with moving people and other objects. Even though users like taking panorama images while they are in the foreground and such scenes as background, those moving objects obstruct the panorama. Hence, in this paper we present a novel method of panorama generation which is capable of avoiding the moving objects from the panorama.

First, user captures a short video of the intended focus area starting from the centre of the expected panorama image, then following a clockwise circular path as shown in Fig. 1.

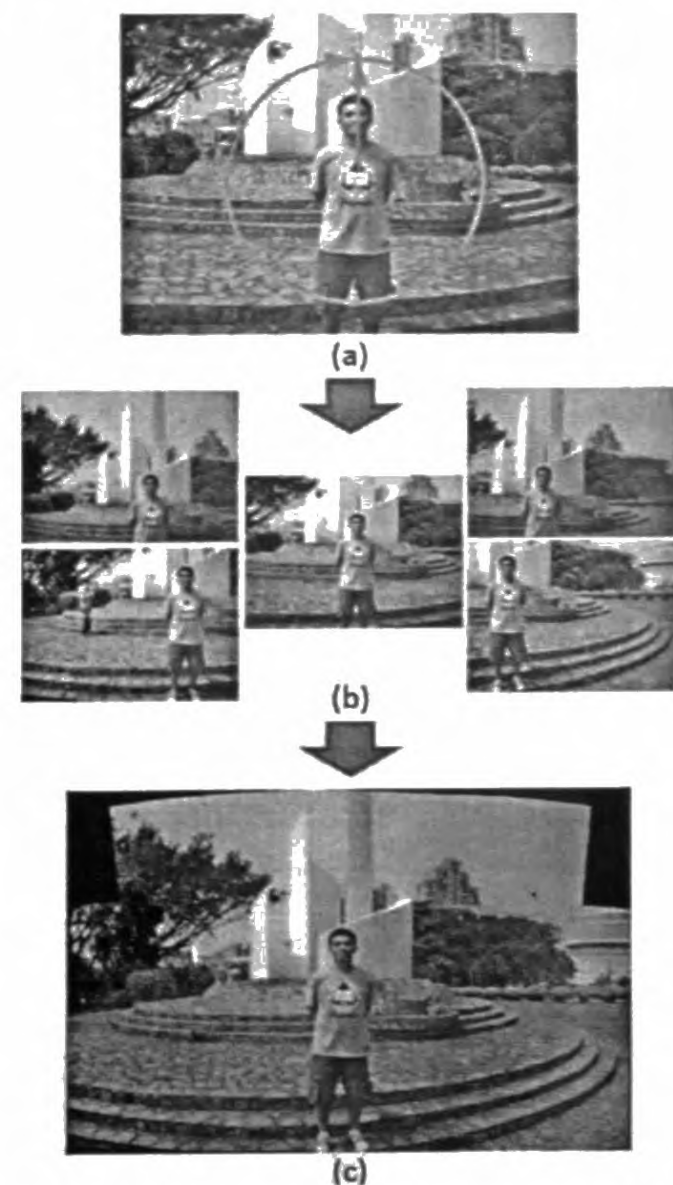


Fig. 1 Flowchart of the system use (a) video direction (b) selected frame set (c) generated panorama

Internal operational steps of the proposed solution are shown in Fig. 2. First, a set of five frames is selected from the source video with equal inter frame distance. Then, there are three main phases as segmentation of human object, removal of moving human objects, and generation of panorama.

The rest of the paper is organized as follows. Section II presents the system architecture details while section III is focused on the experimental results. Finally the presented work is concluded in the section IV with the possible future research directions on the topic.

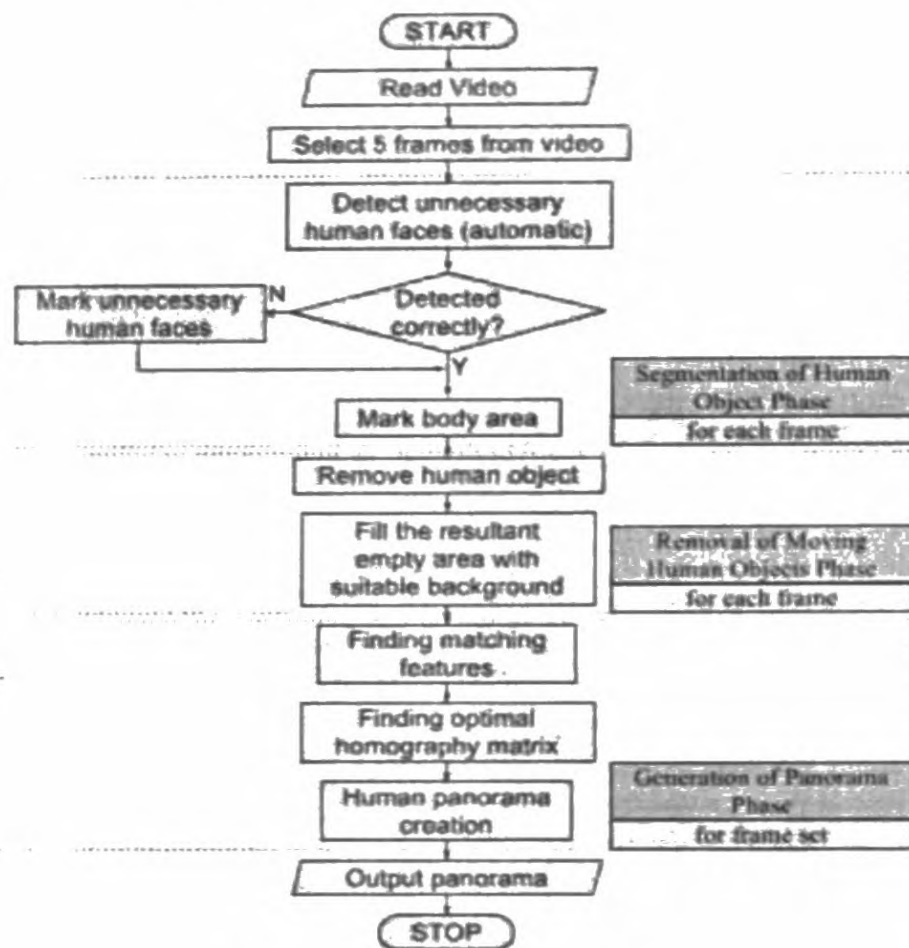


Fig. 2 Operational steps of panorama generation

## II. SYSTEM ARCHITECTURE

This section presents the internal architectural details of the proposed system and the operational steps segmentation of human objects phase, removal of moving human objects phase and generation of panorama phase.

### A. Segmentation of Human Objects Phase

Detection of the faces of the humans appear in the image set of the video is carried out here. Human faces are automatically detected based on the *Haar-Features* [11, 23] and skin color mechanisms of face detection. In the case of failure in detecting faces automatically user can manually mark the face area. Fig. 3 shows one example of face detection of the proposed system.



Fig. 3 Face detection example

Then based on the detected face region human body area is marked except for the main human object. Proposed solution is still limited only to standing adults. Static ratios are used during the marking of body area as,  $height_{face:body} = 6.0$  and  $width_{face:body} = 2.5$ . An example result is shown in Fig. 4.



Fig. 4 An example of marked human body area

Next from that marked human body area, only the human object data is extracted to be used in the next phases. A graph-cut based algorithm named Automatic GrabCut (Auto-GrabCut) is used here for the segmentation of the objects [10, 20, 21, 26]. Algorithm 1 presents the steps of segmentation of human object using Auto-GrabCut in detail. One example of the segmented human object is shown in Fig. 5.

### Algorithm 1. Segmentation of human object

Input: Each frame of the source frame set

Output: Detected human body

begin

1. Convert RGB to grayscale format
2. Find edges using Sobel operator
3. Change the color of the corresponding pixels in the original RGB image (conspicuous points) based on the Sobel values in the grayscale image as follows,  
If (Sobel value of the pixel < 180)  
    New color = black  
Else  
    New color = Red
4. Tag conspicuous points as foreground pixels based on the new pixel values  
Equally divide the definition area (the box around the human body) into two vertically and use *GrabCut* algorithm in each part  
If (result area of *GrabCut*) < (40% of definition area)  
    Set a sample model based on human body structure in the definition area. Sizes of each human body part are calculated based on the size of face area as  $width_{face:hand:leg} = 1:2.2:5$  and  $height_{face:hand:leg} = 1:2.5:3.5$ .  
    Cluster color in each part of definition area into two parts which covered with sample model using mean-shift algorithm and mark the small color cluster as white (background) and large color cluster as red (foreground).  
    Use *GrabCut* algorithm again with the new background and foreground values  
Else  
    Go to step 5
5. Use the erosion and dilation method in the *GrabCut* result.

### B. Removal of Moving Human Objects Phase

In the step A the human objects to be removed were identified. Now those detected human objects should be removed from the image. The vacant region caused by the removal of the human object should be properly covered. The concept called video inpainting [16, 24] can be used to fill the empty area with suitable background. Video inpainting works well for the regular videos, because the background

information required appears in the previous or next frames of the original video. But in here video was captured following a circular path which makes it difficult to find the relevant background information in the same location of the previous or next frame as the background changes when camera moves. Cross Diamond Hexagonal Search (CDHS) algorithm from Cheung *et al.* [9] is used to find the similar structural properties of two consecutive frames, then the relevant background to be used can be found. Fig. 6 shows the schematic diagram of filling the empty area caused by removal of moving human object. A basic box shape is used to demonstrate the concept, even though the human object in the real application is a much complex shape. Algorithm 2 presents the steps of removing the moving human object and filling the background area. A completed result is shown in Fig. 7.

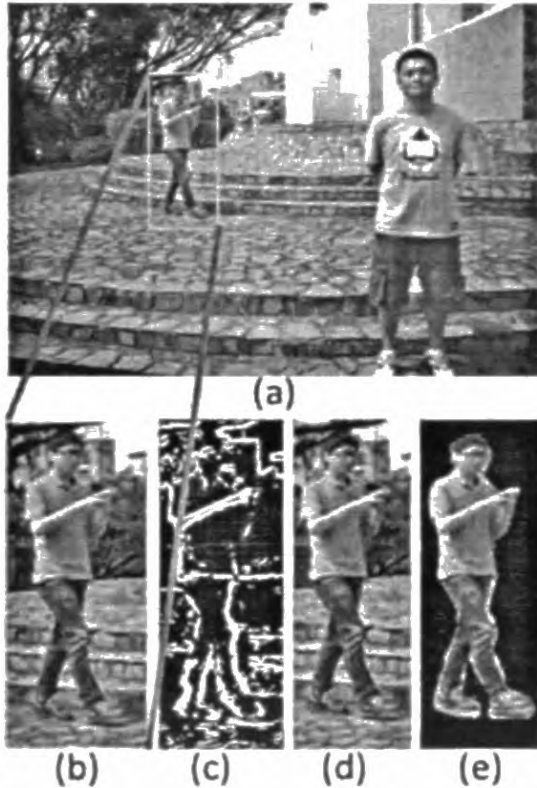


Fig. 5 An example of human object segmentation (a) human body area marked original frame (b) human body area (c) detected edges with Sobel (d) conspicuous points marked (e) Auto-GrabCut result

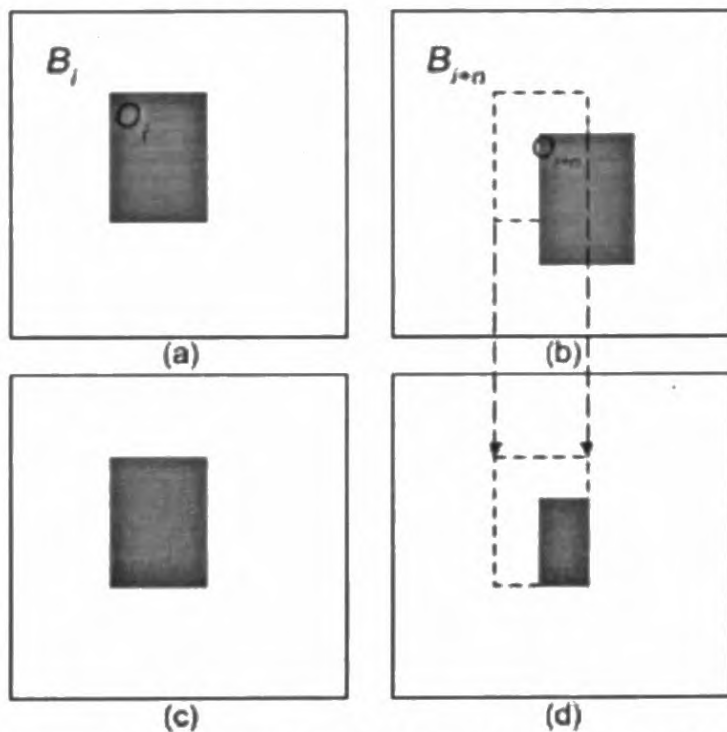


Fig. 6 Background filling procedure (a) original frame, (b) original frame<sub>i+n</sub> (c) after human object removal (d) after background fill

### C. Generation of Panorama Phase

The final panorama generation is discussed in this section. In the section II-B moving human objects have been removed from the selected frames. In this section the procedure of generation of the panorama using those five frames are

discussed. The most important thing here is matching the same features from the frames and merging them together to generate the panorama without any recognizable defects.

#### Algorithm 2: Removal of Moving Human Objects and refilling the background

Data: Detected human body area information and source frame set  
Result: Background refilled image

begin

1. Find the covering region by the object  $O_i$  in frame<sub>i</sub> and frame<sub>i+n</sub> using *Auto-GrabCut*.
2. Find the movement direction of boundary of  $O_i$  using CDHS algorithm. The used block search area is  $15 \times 15$  pixels. The block matching of direction detection in CDHS algorithm is uses the Sum of Squared Differences (SSD) method.

$$SSD \text{ value} = \sum_{i=0}^n (R - R')^2 + (G - G')^2 + (B - B')^2 \quad (1)$$

where,  $R, G, B, R', G'$  and  $B'$  are Red, Blue, Green values in frame<sub>i</sub> and frame<sub>i+n</sub> respectively.

3. Create the mark map to tag the state of background  $B_i$  and  $O_i$  in consecutive frames. The relative position can be extracted with the block matching in step 2.

$$\text{mark map} = \begin{cases} 1, & \text{background } B_i \\ 2, & \text{covering region } O_i \text{ and } O_{i+n} \end{cases} \quad (2)$$

4. If the mark value is changed due to movement of  $O_i$ 
  - a) Change the value of mark map accordingly.
  - b) Calculate the similarity between  $O_i$  and  $B_i$  with Sum of Absolute Difference (SAD) method [19]. Repair  $O_i$  with the background information in  $B_i$ .

$$SAD \text{ value} = \sum_{i=0}^n (R - R') + (G - G') + (B - B') \quad (3)$$

Repeat for all the frames

- c) Load the next frame and repeat from step 1.

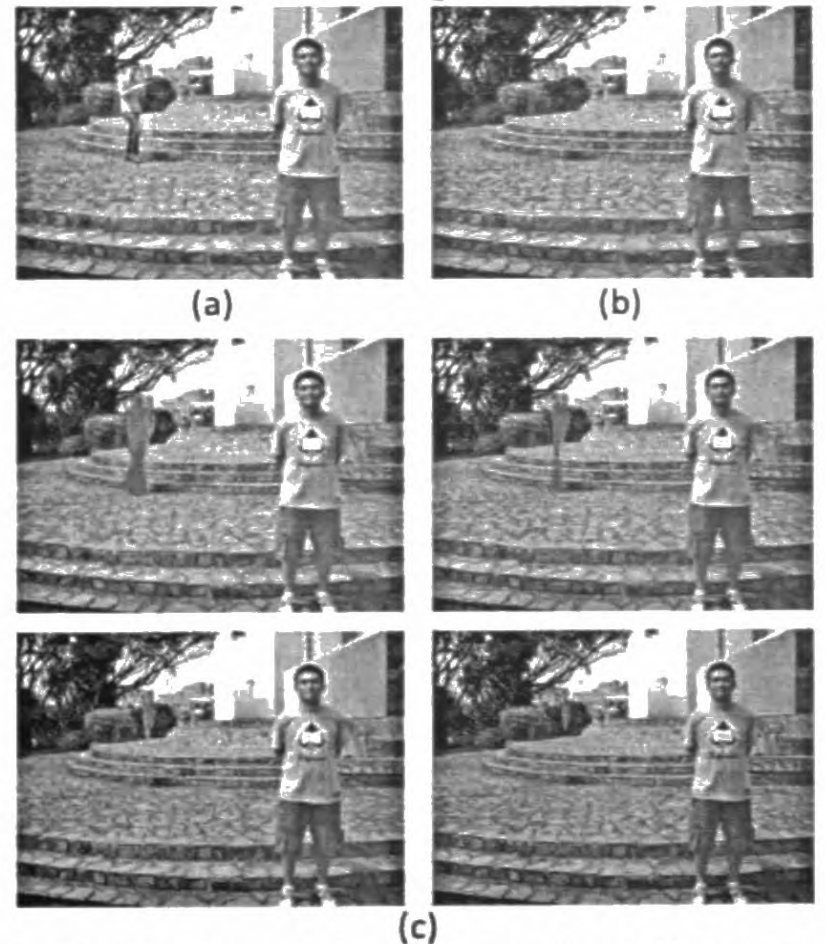


Fig. 7 An example of removal of moving human objects (a) before removal of moving human object (b) after removal of moving human object (c) several selected intermediate steps

Generation of panorama is carried out in three steps as given in Algorithm 3. In first step it is required to find out matching features of the five selected frames in order to merge them together to generate the panorama. Affine-SHIFT (ASHIFT) proposed by Morel [12] is used as the tool to find the matching features from two frames taken in different shooting angles and positions. As the comparing two frames are captured in only

two different angles all the matching features should follow the same angle between them. This concept is used to filter out any false matches by the ASHIFT. An example of matched features using the SHIFT [15, 17, 22, 25] and ASHIFT algorithms using four parameters [12, 13] is shown in Fig. 8.

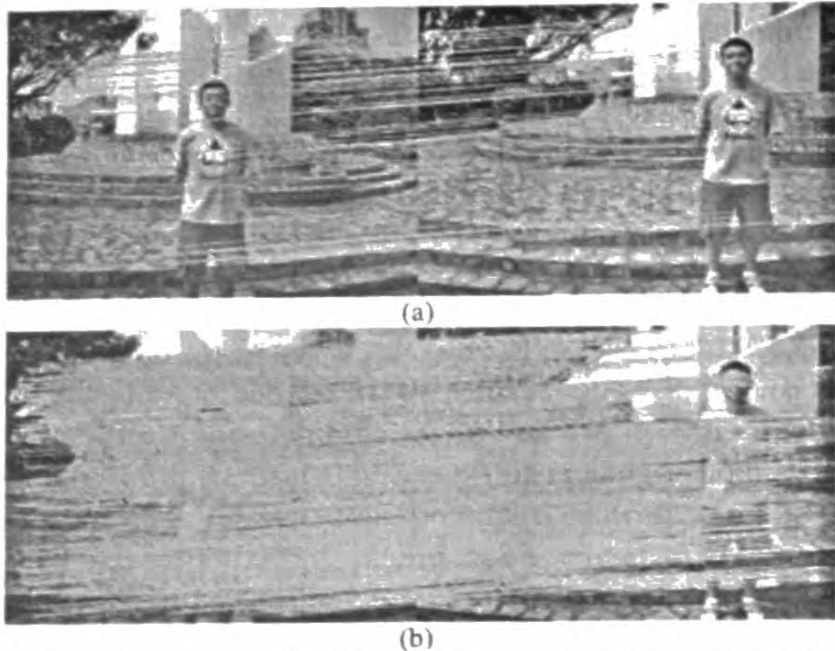


Fig. 8 An example of matched features (a) using SHIFT algorithm (b) using ASHIFT algorithm

Then the source frame set can be merged together to generate the panorama based on the matching feature set. Traditionally average color value of the pixels were used as the new color value in the overlapping regions, which then resulted a ghost effect in the overlapping regions. In this paper seam carving [18] based image stitching method [15] is used to generate the panorama from the source frame set. Seam carving is used to divide the overlapping area of the panorama in to two parts, then left and right of it is filled by using the relevant patches from the originals as shown in Fig. 9.

But as the properties of the images like exposure may slightly differ from each other as in Fig. 10, it is first required to calibre the image set with each other using a homography matrix [8]. In the step 2 homography matrix is found and finally in the third step panorama is generated [14, 15]. In camera calibration the homography [6] is also used to ensure all source images can be projected in the same plane. Therefore it is required to calculate the homography matrix via features that can be extracted by ASIFT algorithm. Hence from [6],

$$s * m' = H * m \quad (4)$$

where,  $s$  is the scale matrix,  $H$  is the homography matrix,  $m = (x, y, 1)$  and  $m' = (x', y', 1)$  is a pair of corresponding points matrix in the original image and in panorama plane.

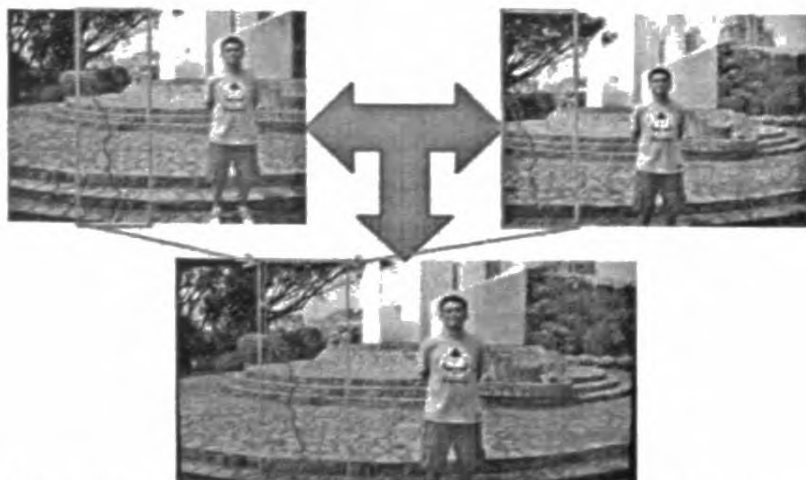


Fig. 9 Generation of panorama using seam carving (optimum seam is marked in red) top: original images, bottom: combined image with overlapping region

is marked in green

Algorithm 3: Generation of panorama

#### Step 1: Finding matching features

Data: Source frame set

Result: Coordinates of the matching feature points

begin

1. Find the matching feature points between two frames with ASIFT algorithm.
2. Calculate the gradient  $M_F$  of matching feature points in frames A and B as,

$$M_F = \frac{\Delta Y}{\Delta X} = \frac{Y_B - Y_A}{X_B - X_A} \quad (5)$$

where  $(X_A, Y_A)$  and  $(X_B, Y_B)$  are the coordinates of the matching points.

3. If  $M_F \neq 0$ , find the number of high similarity feature matching using SAD method in a small range bounded by  $3 \times 3$  pixels.
4. Find the maximum number of  $M_F$  and compute calibration parameters matrix using coordinates of matching features.
5. Repeat for all frames.

#### Step 2: Finding optimal homography matrix

Data: Source frame set and coordinates of matching feature points

Result: Homography matrix

begin

1. Cluster matching feature points according to color features using the mean-shift algorithm
    - Convert the color space into CIELuv
    - Copy the  $L$  and  $U$  dimension parameters of CIELuv into a 2D array  $arrayLU$
    - According to the  $arrayLU$  perform the clustering process
    - Eliminate small regions by merging with neighbor regions
  2. For each group calculate the homography matrix by using the matching feature points within the group. Solve at least four pair of corresponding points. If there is no sufficient number of points the group is neglected (see Appendix A for more details).
  3. The homography matrix of each group is fed into Eq. 6 to calculate the value of  $(H * m)$  and compare the deviation  $d$  between the actual  $m'$  and the calculated  $(H * m)$  where  $n$  is the number of matching feature pairs.
- $$d = \frac{\sum_{i=0}^n (m'_i - H * m_i)}{n} \quad (6)$$
4. The optimal homography matrix is the one with the minimum deviation that is the one with the smallest  $d$ .

#### Step 3: Human panorama creation

Data: Homography matrix and the source frame set without moving human objects

Result: Final panorama

begin

1. Registration: Image registration involves matching features in a set of images or using direct alignment methods to search for image alignments that minimize the SAD method between overlapping region because the overlapping region is relatively similar than the other regions between two image. In some methods of image stitching the registration is done manually because this may need less processing time and let the stages become faster.
2. Calibration: Image calibration [7] aims to minimize differences between an ideal lens models and the camera-lens combination that was used optical defects such as distortions. If feature detection methods were used to register images and absolute positions of the features were recorded and saved, that data can be used for geometric optimization of the images. In the proposed system the feature detection methods use ASIFT algorithm to find feature matching in the set of source image.
3. Blending: Image blending involves executing the adjustments of source images resulted in the calibration stage and combine with remapping of the images to an output projection. Colors are adjusted between images to compensate for exposure differences. In this part color obtained also can use the averaging method and interpolation to compute the value of pixel.

III. EXPERIMENTAL RESULTS

Here the results of the experiment are discussed. Without using any supporting device for the camera (like tripod) input videos were captured to simulate a regular user who uses a regular camera. Selected set of frames and the generated output panorama for three videos are shown in Fig. 10.

A PC with 1.8 GHz CPU and 2 GB RAM is used for the processing of the results. Time taken in each phase of the process for the three videos was measured and displayed in Table I and Fig. 11. As can be seen in the Table 1 and Fig. 11 time taken in the segmentation of human objects and removal of moving human objects phases is almost similar for all three source videos. However time taken in the generation of panorama phase for three source videos is largely different. The reason for this probably is the color complexity and structural complexity of the input frame set. As an example Fig. 10-(c) video which was captured indoor required a comparatively less processing time due to the fact that it is less complex in color and structure comparing to Fig. 10-(b) and (c).

TABLE I  
TIME TAKEN TO GENERATE THE PANORAMA

Experiment video	Segmentation of human object (s)	Removal of moving human objects (s)	Generation of panorama (s)	Total (s)
(a)	27.31	27.84	485.27	540.42
(b)	25.52	24.17	402.80	452.48
(c)	25.41	25.76	100.81	151.98

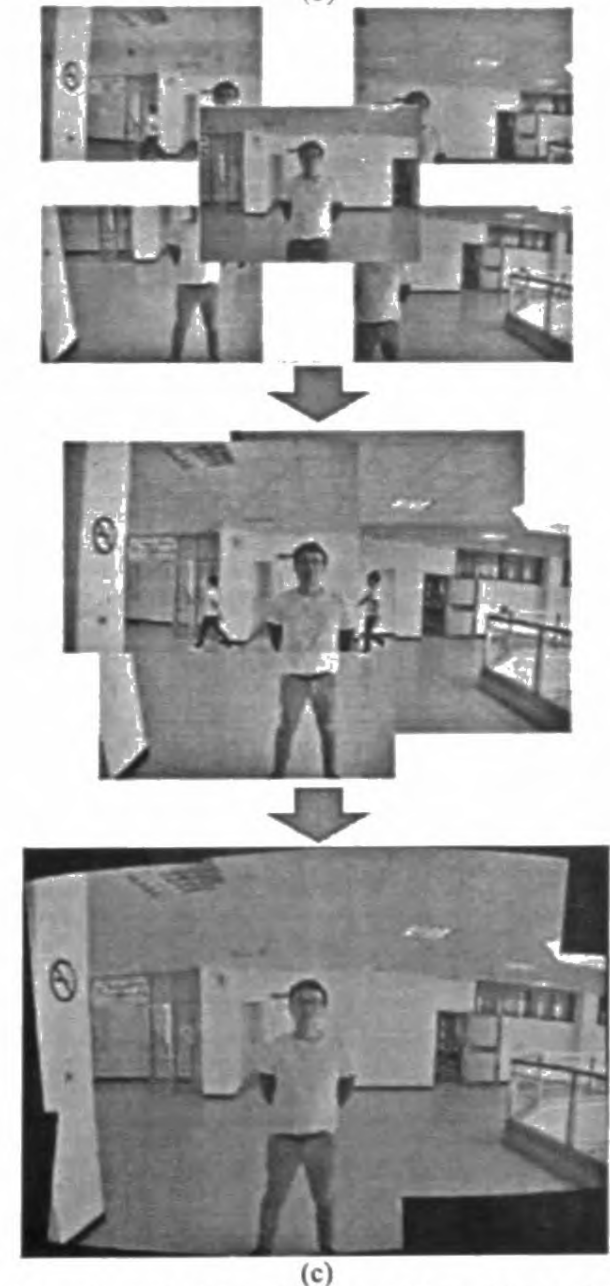
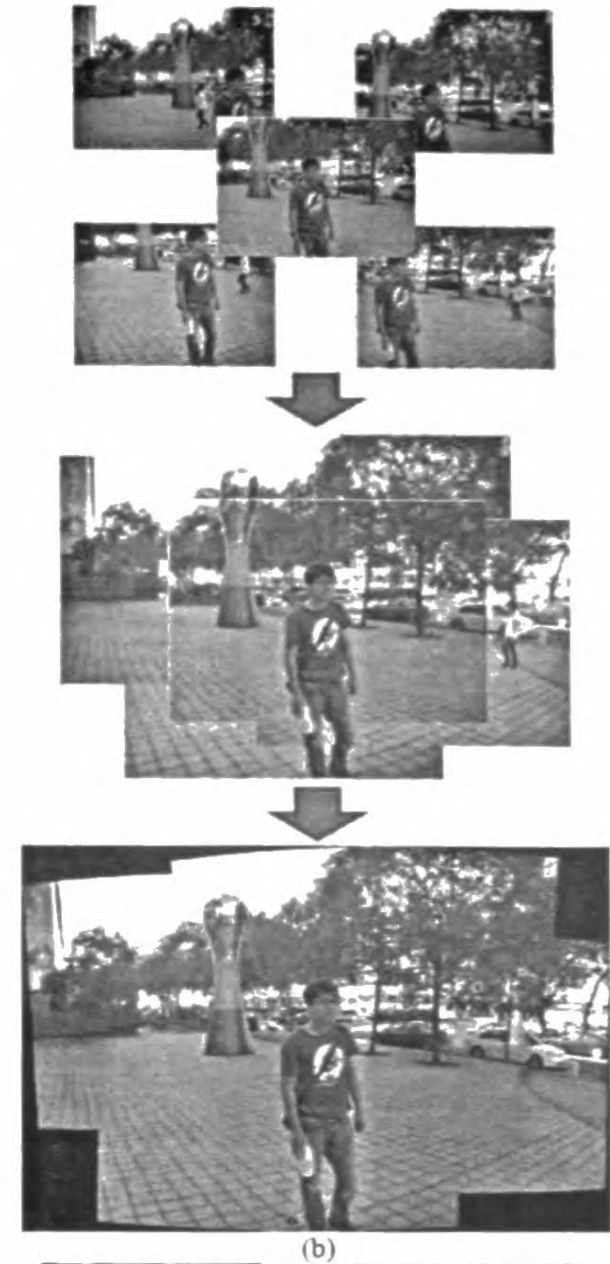
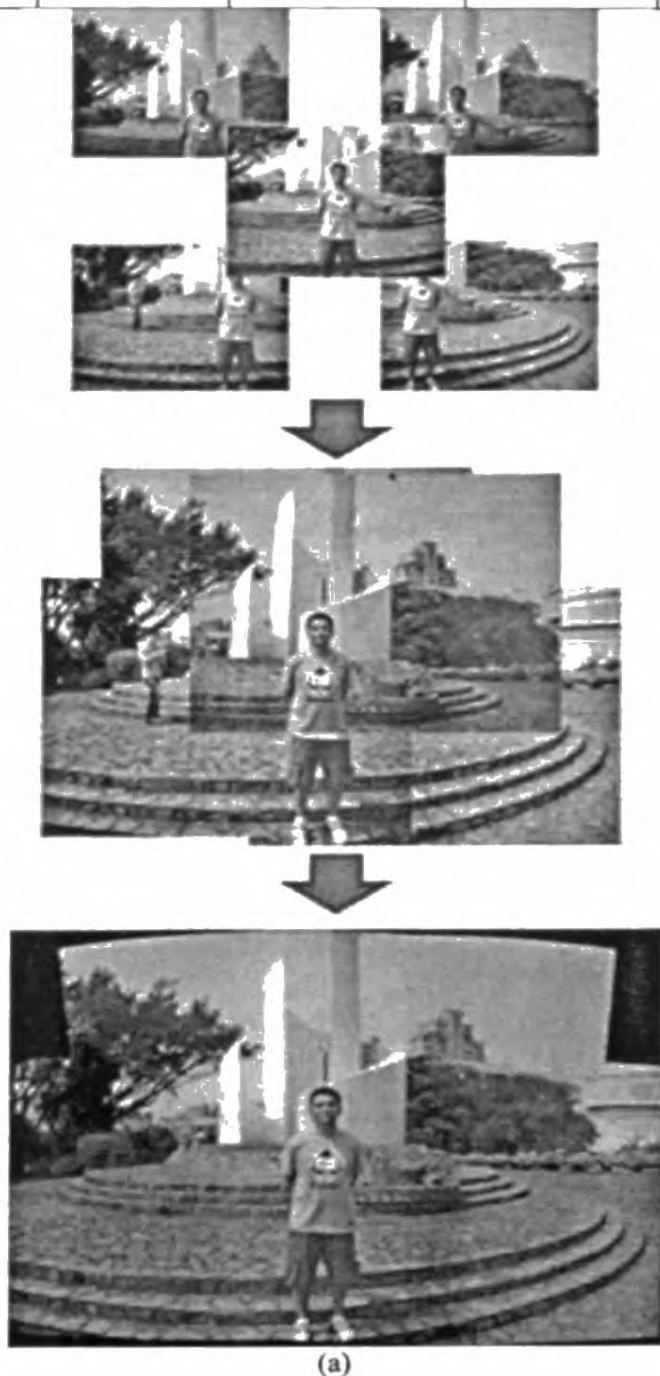


Fig. 10 Experimental results (a) and (b) are outdoor, (c) is indoor, for each set upper frame set is the selected input frame set from the input video, second is merged results without camera calibrations, and below photo is the generated panorama. Note that moving human objects are removed from the panorama

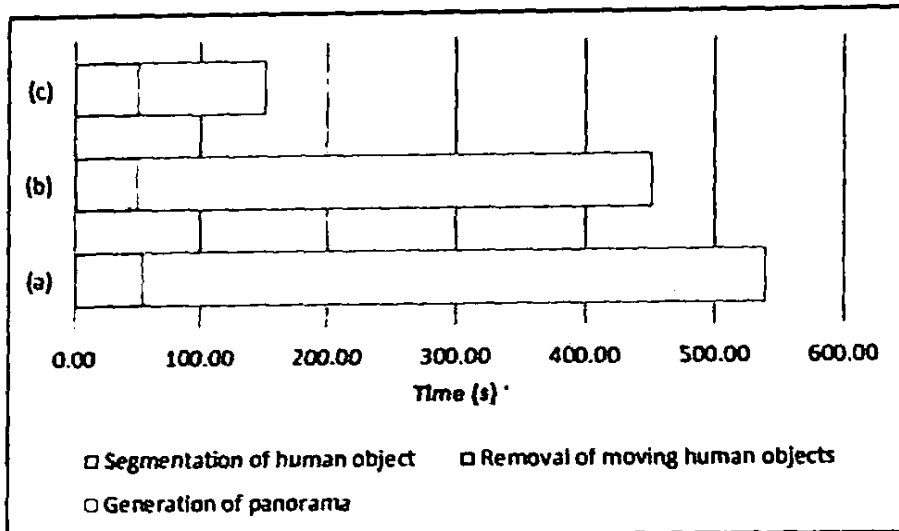


Fig. 11 Time taken to generate the panorama in each phase for the experimental results given in Fig. 10

#### IV. CONCLUSIONS

A novel method for generation of panorama image from a video captured from a simple digital camera by a novice user is presented in this paper. Users prefer panorama photos of famous buildings or large structures with main human objects as the foreground. But normally such locations are crowded with moving human objects in the background. Most important contribution of the presented method is that it is capable of removing such moving human objects from the background. User first captures a short video starting from the center of the main human object and follows a clockwise circular path covering the required background area. After the video is fed to the system, it generates the panorama image following three main phases as segmentation of human objects, removal of moving human objects, and generation of panorama. Presented results show that application generates good quality panorama images.

Authors are further investigating the ways to improve the application. Target future directions are implementation of the application for mobile platforms and automatically frame the generated panorama without current empty regions around.

#### ACKNOWLEDGMENT

This work was supported by the National Science Council, Taiwan with the project Development of Intelligent Image Inpainting System (Project No.: 101-2622-E-008-004-CC2). Authors thank to all colleagues who supported during the work.

#### REFERENCES

- [1] Kekre, H. B., and Sudeep D. Thepade. "Rotation Invariant Fusion of Partial Image Parts in Vista Creation using Missing View Regeneration." *WASET International Journal of Electrical Computer and Systems Engineering (IJECSSE)* 2, no. 2 (2008): 94-101.
- [2] Wang, Meng. "Panorama Painting: With a Bare Digital Camera." In *Image and Graphics, 2009. ICIG'09. Fifth International Conference on*, pp. 82-86. IEEE, 2009.
- [3] Wagner, Daniel, Alessandro Mulloni, Tobias Langlotz, and Dieter Schmalstieg. "Real-time panoramic mapping and tracking on mobile phones." In *Virtual Reality Conference (VR), 2010 IEEE*, pp. 211-218. IEEE, 2010.
- [4] Song, Baosen, Yongqing Fu, and Jinlin Wang. "Automatic panorama creation using multi-row images." *Information Technology Journal* 10, no. 10 (2011): 1977-1982.
- [5] Lin, Wen-Yan, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong. "Smoothly varying affine stitching." In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 345-352. IEEE, 2011.
- [6] A. Criminisi, I Reid, A Zisserman (1999) A plane measuring device. *Image and Vision Computing*. 17(8):625-634.
- [7] B. Heigl, R. Koch, M. Pollefeys, J. Denzler, L. Van Gool (1999) Plenoptic modeling and rendering from image sequences taken by hand-held camera. In *Mustererkennung*. pp 94-101.
- [8] Bruno Postle (2011) Panorama Tools. Open Source Software for Immersive Imaging. <http://panotools.sourceforge.net/>. Accessed 22 February 2011.

- [9] C. H. Cheung, L. M. Po (2005) Novel cross-diamond-hexagonal search algorithms for fast block motion estimation. *IEEE Transactions on Multimedia*. 7(1):16-22.
- [10] C. Rother, V. Kolmogorov, A. Blake (2004) "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23:309-314.
- [11] Jian Yao, Jean-Marc Odobez (2008) Fast human detection from videos using covariance features. Presented at: The Eighth International Workshop on Visual Surveillance (VS2008). <http://hal.inria.fr/inria-00325628/>. Accessed 29 September 2008.
- [12] J. M. Morel, G. Yu (2009) ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences*. 2(2):438-469.
- [13] L.-M. Po, W. C. Ma (1996) A Novel Four-Step Search Algorithm for Fast Block Motion Estimation. *IEEE Trans. Circuits Syst. Video Technol.* 6(3):313-317.
- [14] M. Brown, D. G. Lowe (2003) Recognising Panoramas. In *Proceedings of the 9th International Conference on Computer Vision (ICCV2003)*. pp 1218-1225.
- [15] M. Brown, D. G. Lowe (2007) Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision*. 74(1):59-73.
- [16] Patwardhan KA, Sapiro G, Bertalmio M. (2007) Video Inpainting Under Constrained Camera Motion. *IEEE Transactions on Image Processing*. pp 545-553.
- [17] Saeid Fazli, Hamed Moradi Pour, Hamed Bouzari, (2009) Particle Filter based Object Tracking with Sift and Color Feature. *International Conference on Machine Vision*. pp 89-93.
- [18] Shai Avidan, Ariel Shamir (2007) Seam Carving for Content-Aware Image Resizing. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*. 26(3): Article 10. doi:10.1145/1275808.1276390
- [19] S. Vassiliadis, E. A. Hakkennes, J. S. S. M. Wong, G. G. Pecharnek (1998) The Sum-Absolute-Difference Motion Estimation Accelerator. In the 24th. *EUROMICRO Conference*. pp 559-566.
- [20] S. Vicente; V. Kolmogorov, C. Rother (2008) Graph cut based image segmentation with connectivity priors. In *IEEE Conference on Computer Vision and Pattern Recognition*. pp 23-28.
- [21] V. Vezhnevets, V. Konouchine (2005) Grow-Cut - Interactive Multi-Label N-D Image Segmentation. *Proc. Graphicon*. pp 150-156.
- [22] Wan-Lei Zhao, Chong-Wah Ngo (2009) Scale-Rotation Invariant Pattern Entropy for Keypoint-Based Near-Duplicate Detection. *Image Processing of IEEE Transactions*. 18(2):412-423.
- [23] Yu-Ting Chen, Chu-Song Chen (2008) Fast Human Detection Using a Novel Boosted Cascading Structure With Meta Stages. *IEEE Transactions on Image Processing*. 17(8):1452-1464.
- [24] Y. Wexler, E. Shechtman, M. Irani (2007) Space-Time Completion of Video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 29:463-476.
- [25] Yifan Lu, Lei Wang, Hartley, R., Hongdong Li, Chunhua Shen (2008) Multi-view Human Motion Capture with an Improved Deformation Skin Model. *Computing: Techniques and Applications (DICTA) Digital Image*. pp 420-427.
- [26] Yuri Y. Boykov, Marie-Pierre Jolly (2001) Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. *Eighth International Conference on Computer Vision (ICCV'01)*. 1:105-112.

#### APPENDIX A

Eq. 1 can be expanded as,

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (7)$$

where,  $h_{ij}$  represents each element of the homography matrix.

Eq. 7 can be further simplified as follows,

$$\begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix} = \begin{bmatrix} h_{11}x + h_{12}y + h_{13} \\ h_{21}x + h_{22}y + h_{23} \\ h_{31}x + h_{32}y + h_{33} \end{bmatrix} \quad (8)$$

Then value of s can be found as follows,

$$s = h_{31}x + h_{32}y + h_{33} \quad (9)$$

By applying s in first two rows of Eq. 8,

$$\begin{aligned}(h_{31}x + h_{32}y + h_{33})x' &= h_{11}x + h_{12}y + h_{13} \\ (h_{31}x + h_{32}y + h_{33})y' &= h_{21}x + h_{22}y + h_{23}\end{aligned}\quad (10)$$

Scale of H is a variable and  $h_{33}$  is usually normalized to 1 [6]. Therefore to solve the eight parameters at least four pairs of corresponding points are required. As an example for point pair  $(x_1, y_1)$  and  $(x'_1, y'_1)$  equation can be rewritten as,

$$\begin{aligned}(h_{31}x_1 + h_{32}y_1 + 1)x'_1 &= h_{11}x_1 + h_{12}y_1 + h_{13} \\ (h_{31}x_1 + h_{32}y_1 + 1)y'_1 &= h_{21}x_1 + h_{22}y_1 + h_{23}\end{aligned}\quad (11)$$

By taking  $x'_1$  and  $y'_1$  to the right of equation,

$$\begin{aligned}h_{11}x_1 + h_{12}y_1 + h_{13} - h_{31}x_1 - h_{32}y_1 &= x'_1 \\ h_{21}x_1 + h_{22}y_1 + h_{23} - h_{31}x_1 - h_{32}y_1 &= y'_1\end{aligned}\quad (12)$$

Repeat the same for all four point pairs and the following equation can be obtained. Then by solving this homography matrix can be found.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2 x_2 & -x'_2 y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2 x_2 & -y'_2 y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3 x_3 & -x'_3 y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3 x_3 & -y'_3 y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4 x_4 & -x'_4 y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4 x_4 & -y'_4 y_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix}\quad (13)$$