VideoVR: A Real-Time System for Automatically Constructing Panoramic Images from Video Clips

Ding-Yun Chen, Murphy Chien-Chang Ho, Ming Ouhyoung

Communications and Multimedia Lab.
Dept. of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, 106, R.O.C.

Abstract. An authoring system is proposed to construct panoramic images of real-world scenes from video clips automatically. Instead of using special hardware such as fish-eye lens, our method is less hardware-intensive and more flexible to capture real-world scenes without loss of efficiency. Unlike current panoramic stitching methods, where users need to select a set of images before constructing a panoramic image, our system will choose essential frames and stitch them together automatically in 16 seconds on a Pentium-II PC. In addition to popular image-based VR data formats, we also output the panoramic images in VRML97 format.

1 Introduction

In recent years, panoramic images have been widely used to build virtual environments from real-world scenes[1,2,8,10,11,12]. Hybrid geometry- and image-based approach for the rendering of architectures is also propose [13]. Besides using special hardware to capture panoramic images of real-world scenes[5,6,9], a number of techniques based on "stitching" have been developed[10,11,12]. However, most of the techniques of this class are of high computational complexity, need multiple steps to build a single panoramic image, and users are required to adjust parameters interactively.

In this paper, we introduce a real-time system to capture panoramic images of real-world scenes automatically[3] by simply panning a hand-held camcorder. Our implementation of constructing a panoramic image from 6 seconds long CIF format video clip takes 16 seconds on a Pentium II-233 PC.

2 System Overview

Video clips provide more information about recorded scenes than that of photos. Due to the characteristic of high similarity and low difference between successive video frames, correctness rate of stitching is higher. In general, for a 320x240 true-color video clip, the uncompressed data rate is about 6.6 Megabytes per second. To efficiently construct panoramic images, we use the folowing multi-stage approach:

Stage1: Capture and choose necessary frames
Stage2: Calculate accurate translation of selected frames, and stitch them together
Stage3: Cylindrical stitching
Stage4: Output in VRML97 format or other formats

In general, stitching and constructing a 360° panoramic image needs 15 to 30 images in average. It is unnecessary to use all frames in video clips to construct a single panoramic image. So, capturing and choosing necessary frames in entire video clips is done at the first stage. Then, the next stage is to stitch together selected frames. The algorithm to capture and stitch frames is described in section 3.

After the completion of stitching all selected frames, seeking and stitching the beginning and the ending parts is needed to construct a 360° panoramic image. Section 4 describes this algorithm.

3. Capturing and Stitching Frames

The method we use to stitch two frames together is by minimizing the gradient error between two frames, and uses the following equation,

$$E(\Delta x, \Delta y) = \sum_{x,y} [G_1(x'+\Delta x, y'+\Delta y) - G_0(x, y)]^2 \quad (1)$$

where G is gradient operator, (Δx, Δy) is the translation which is the same for all pixels, (x'+Δx,y'+Δy) and (x,y) are corresponding points in two images, and E is error term.

In the first stage, choosing a frame depends on the distance of the translation between previously selected frames. If the distance is smaller than a threshold (we use one-sixteenth of frame size), the frame is dropped until there are enough distance between previously selected frames. In this stage, all we want to know is the approximate distance to decide whether to choose a frame or not. Therefore, we use decimated image of the full frame to calculate the translation efficiently using equation (1). For example, we first sub-sample an 64x48 image from 320x240 image. Next, prediction is used to decrease the search space, which allows this algorithm to be more efficient. In our implementation, this stage could process the input video clips in real-time.

To accurately stitch together selected frames, the error that happens during calculating the translation can not be ignored. So, we make a local search to find the minimal gradient error in stage 2. After obtaining accurate translation, the selected frames are painted on a canvas according to their translation. To reduce


(a)


(b)

Fig. 1. Capturing and stitching frames: (a) six frames in video clips; (b) stitched frame in a panoramic image

discontinuities in intensity and color between frames, we weight the pixels in each frame proportionally to their distance away from its border. Fig 1 shows the selected frames from video clips and the stitched image.

4. Cylindrical Stitching

To construct a 360° panoramic image, seeking and stitching the beginning and the end of a canvas which is constructed in stage2 (Fig 2a, 3a) is needed. The way to seek is also depending on equation (1). We get a block from the right side of the canvas, and match it in the left region. Once the correct location is found, the redundant region will be clipped out. However, due to accumulated errors during stitching frames, the vertical location of the beginning of the canvas can be different from the vertical location of the end. To solve this problem, the canvas (Fig 2a, 3a) is sheared smoothly to compensate for vertical difference (Fig 2b, 3b).

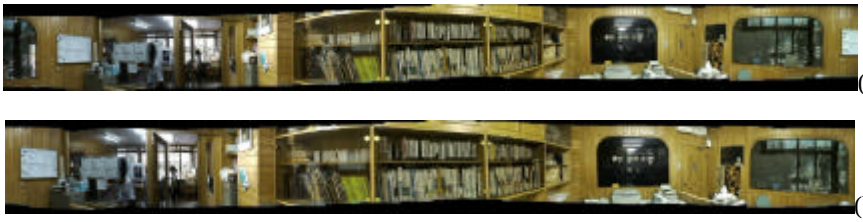

Fig. 2. Indoor scene (a) capturing and stitching frames; (b) cylindrical stitching

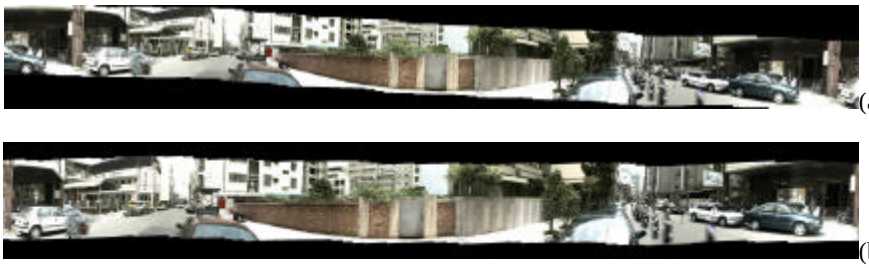

Fig. 3. Outdoor scene (a) capturing and stitching frames; (b) cylindrical stitching

5. Conclusion

In this paper, we have developed a real-time system to capture panoramic images of real-world scenes automatically from video clips. We take advantage of using video clips because it provides a lot of information. We also avoid redundant video frames by choosing essential frames and then stitching them together. There is a fundamental assumption that focal length and aperture change slowly between successive frames and do not change in tilting or rotation. Therefore, instead of warping the input frames and estimating the focal length [1, 11] to stitch frames in cylindrical coordinate, input frames were stitched together directly to construct panoramic images. In short, this is not intended for challenging cases of forward motion and of zoom, but for quick

construction in “rotation” only cases. Experiment results are given in Table 1. After the panoramic image is constructed, it can be converted to other formats and we can use some image-based browsers such as LivePicture[8] and PhotoVR[14] to navigate it in Web pages. In our implementation, we can also output the panoramic images in VRML97 format. This allows us to use standard VRML browsers to navigate. For further information, such as executable programs and demos, please refer to http://www.cmlab.csie.ntu.edu.tw/cml/g/VideoVR.

Stage No. / Test Video	Stage1		Stage2				Stage3		Total time used
	Number of frames	Time used	Number of selected frames	Time used	Size after stitching frames	Time used	Size after cylindrical stitching	Time used	
Fig2	196	4.67s	88	5.82s	3018x363	2.03s	2667x363	3.20s	15.72s
Fig3	166	3.90s	84	5.54s	2942x293	1.82s	2705x293	1.43s	12.69s

Table 1. Panoramic image generation using a Pentium II-233 64MB RAM PC, and video clips are in CIF format

References

[1] R. Szeliski and H.Y. Shum, “Creating Full View Panoramic Image Mosaics and Environment Maps”. Proc. of ACM SIGGRAPH’97 (Los Angeles), pp.251-258, August 1997.

[2] S. E. Chen, “QuickTime VR – an Image-based Approach to Virtual Environment Navigation”. Proc. of ACM SIGGRAPH’95, pp. 29-38, August 1995.

[3] Video Brush, http://www.videobrush.com

[4] QuickTime VR, http://qtvr.quicktime.apple.com

[5] IPIX, http://www.ipix.com

[6] Be Here, http://www.behere.com

[7] Smooth Movie, http://www.smoothmove.com

[8] Live Picture, http://www.livepicture.com

[9] OmniCam,http://www.cs.columbia.edu/CAVE/ omnicam

[10] R. Szeliski, “Video Mosaics for Virtual Environments”, IEEE Computer Graphics and Applications , pp. 22-30, March 1996.

[11] B. Rousso, S. Peleg, I. Finci and A. Rav-Acha, “Universal Mosaicing Using Pipe Projection”, International Conference on Computer Vision , pp. 945-952, 1998.

[12] S. Peleg and J. Herman, “Panoramic Mosaics with VideoBrush”. In IUW-97, New Orleans, Louisiana, May 1997. Morgan Kaufmann, pp. 261-264.

[13] P. E. Debzvec, C. J.Taylor, J. Malik, “Modeling and Rendering Architecture from Photographs: A Hybrid Geometry and Image-based Approach”, pp.11-20, Proc. of ACM SIGGRAPH’96, New Orleans, USA, 1996.

[14] J.J. Su, Z.Y. Zhuang, S.D. Lee, J.R. Wu, and M. Ouhyoung, "Photo VR: An Image-Based Panoramic View Environment Walk-Through System", Proc. of IEEE International Conference on Consumer Electronic (ICCE’97), pp. 224-225, Chicago, USA 1997.