PanoVi: A Multi-Camera Panoramic Movie System by Using Client-Side Image Mosaicking

MUNEHIRO DOI   TSUYOSHI YAMAMOTO
Graduate School of Engineering,
Hokkaido University,
Sapporo, Hokkaido, JAPAN
{mune,yamamoto}@nis-ei.eng.hokudai.ac.jp http://nis-ei.eng.hokudai.ac.jp

Abstract: - In this paper, we introduce a new panoramic movie system that can capture and play 360° viewing angle. The system captures surround movie by using four wide-angle video cameras, and then the video streams are encoded as a single stream of common video format. At the viewing side, geometric correction and image mosaicking are performed in real-time by using 3D graphics accelerators hardware. By mapping the mosaic on the inside wall of a cube in 3D space, the viewing system renders partial view of arbitrary direction as a virtual environment. This approach is suitable for real-time transmission of panoramic movie because this system uses standard video format in transmission layer. We developed a prototype system based on Windows and DirectX8 environments. We confirmed full-rate interactive panoramic movie playback on commodity Pentium4 based PCs.

Keywords: - Panorama movie, Image Mosaicking, Texture Mapping, DirectX, Virtual Reality

1 Introduction

Gains in computing power and network bandwidth have increased the popularity of technologies to record and transmit virtual presence. One aspect of virtual presence that is drawing much attention is panoramic movie. One successful system, still in its early stages, is Apple Computer's QuickTimeVR(QTVR)[1]. In QTVR, pictures are taken by a turntable-mounted camera before being mosaicked into one large image using a cylindrical coordinate system. This is performed as a content authoring process, and a viewer program cuts a portion from the panorama image depending one the viewing direction and corrects distortion caused by cylindrical encoding. QTVR's owes its commercial success to several strengths. First, it does not require special equipment for panorama scene capturing. It assumes that the pictures are taken by an ordinary camera, and that its deliverable content can be composed by personal computer. The second strength is the QTVR player's ability to run on less powerful PCs because it does only simple 2D image processing.

Following QTVR's success, more sophiscated panorama picture formats[2, 3, 5] have been proposed. Some of these afford more freedom in changing the viewing parameters, others provide improved accuracy of mosaicking; however, the impact from such sophiscated systems has not been as strong as that of QTVR despite those systems' existence for many years.

It is natural to imagine a movie version of a panorama picture system[4, 6, 10]. Up to now, several panoramic movie systems have been reported, some of which are commercially available. The BE HERE[10] system captures a 360° image using a spherical mirror attached to commercially available digital cameras. The system can capture a panoramic movie without the mosaicking process. Employing a spherical mirror or a fisheye lens is a straightforward technique for capturing panorama movies; however, this results in poor image quality. One reasons is inefficient use of video frame because these techniques map the subject area onto a small circle (most of video frame formats are rectangular); another is poor resolution from chambering the panoramic scene into

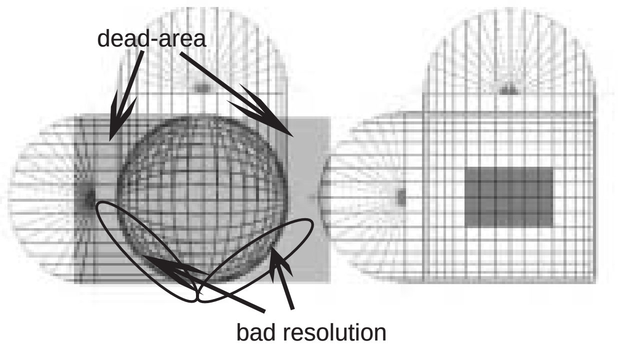

Fig.1 Frame efficiencies of fisheye lens(left) and wide-angle lens(right).

Fig.2 The PanoVi camera system.

a frame(Figure 1).

Multi-camera capturing is an alternative approach that improves the image quality of panorama movies (Figure.2). However, such an approach requires a specially designed optical configuration to reduce the parallax effect caused by misalignment of the camera focal points. Furthermore, greater bandwidth is required to record and transmit this multi-channel video signal. Consequently, the total cost of such systems ises and the range of applications is restricted.

There are several commercial products that use this approach. The FlyCam[11] system uses up to four cameras to take a full view. The system focuses on streaming transmission via Internet, after which aberration correction and image mosaicking are performed by the transmission server. To save bandwidth, the four vido channels are multiplexed into one stream, which is transmitted to the server in NTSC format. Even though the resolution is intentionally dropped for Internet streaming, the average resolution is higher than the solutions using mirror and lens because four cameras capture full view images. The capturing unit is small, compact and capable of stand-up installation; however, the FlyCam system is not applicable for live outdoor use because mosaicking is performed on the transmission server side.

This paper proposes PanoVi, a panorama movie system featuring both high image quality and simple capture architecture [7, 8, 9]. PanoVi uses four miniature wide-angle video cameras to cover a 360° viewing angle as is also the case for the FlyCam system; however, mosaicking process is performed by the viewing software on the client-side. Also, PanoVi uses standard video formats such as NTSC or MPEG, so it may be applicable to existing transmission layer protocols. That gives capability for real-time network applications, as well as scalable extensibility. The main target of the PanoVi system is the mobile application. The capturing subsystem can very easily be made mobile by mounting it on a car or setting it on the ground. The system would be particularly suited to virtual touring and observation.

2 Architecture

Figure 3 overviews the PanoVi system, which consists of a subsystem for capturing, one for transmiting and another for viewing. The capturing subsystem takes a full view using four small general-purpose CCD cameras, and sends the video streams to a transmission line in quad movie format. The viewing system is designed to run on commercially available PC. The subsystem can play a movie in real time at full rate with image mosaicking. As each component is designed to minimize dependense on special hardware, the capturing and transmission methods can be arbitrary designed to fit the purpose. The following sections discuss the capturing and viewing subsystems.

2.1 The Capturing Subsystem

A full-view movie is captured by four tiny CCD cameras positioned on each edge of a square pillar(Figure 2). Each camera has more than 100° of viewing angle so as to overlap the next camera's viewing area. Too little overlap will cause difficulty to mosaicking the images. The camera viewing angle used for the prototype system is 114°. All the captured video streams are packed into one movie stream using a quad unit. The resolution of each stream is degraded to the half of original; however, this degradation can be improved by using higher-resolution video format such as HDTV. The PanoVi architecture favors standard formats for their long-term extensibility, rather than opting for the current supecial devices that afford high resolution or high performance.

The multiplexed video stream can be sent to the client subsystem via transmission line, and the format may be altered as required. Of course, if we choose a standard format such as NTSC, we can also use existing video device immediately. The prototype system uses NTSC, DV(also called IEEE1394, FireWire), MPEG1, MPEG2, MPEG4 in transmission or packet medium. These are all common formats for broadcasting and the Internet, so PanoVi is easily applicable to these contents delivery systems.

2.2 The Viewing Subsystem

At the viewing system end, the panorama content delivered through a digital network or packet medium may be decoded and mosaicked to obtain panoramic image. The decoded movie frame is sent to the texture memory storage on 3D graphics acceleration hardware. This movie texture is mapped onto the inside of a cube. The mapping correspondence between stream and wall is one-to-one, and texture coordinates are calculated as a combination of the result of image mosaicking and lens distortion. As there are four video streams per texture image, the four walls receive their textures as part of the quadruple texture image without needing to be unfolded.

Many studies have concidered panoramaic image-making using cylindrical or other mapping[1, 2]. The least-squares

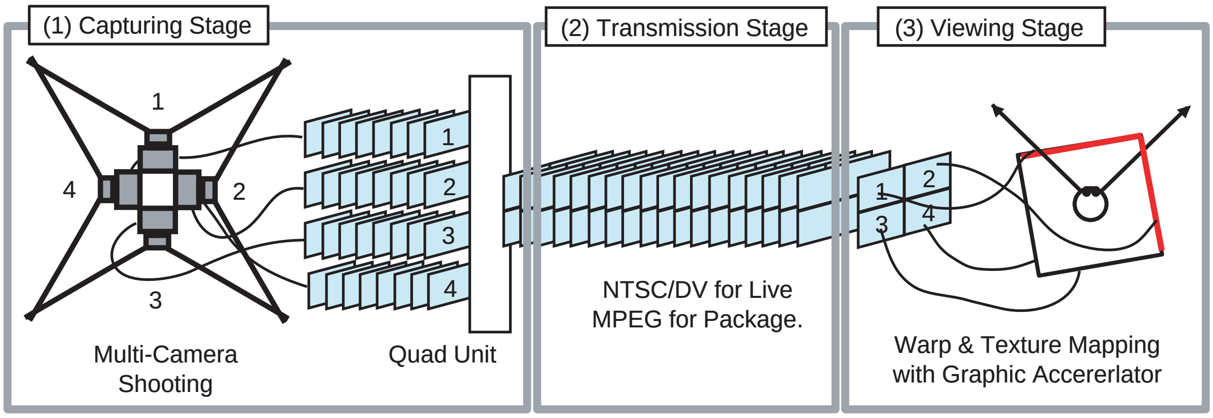

Fig.3 Overview of PanoVi.

method and optical-flow estimation are often used in image mosaicking, but these are likely to be time-consuming. These studies focus on making a high-resolution image by mosaicking several still pictures taken by a handheld camera; in contrast, we focused on real-time capturing and viewing, or panoramic images taken by fixed cameras. Previous methods cannot be applied unmodified to movie panorama mosaicking playing on 30fps.

The main problem in real-time mosaicking is the processing time of dynamic mosaicking in each frame. However, if these parameters can be assumed to be constant over short periods of time, then the processing can be assumed to be simplified. The PanoVi system uses this assumption to achieve a full-rate panoramic movie.

Our method is based on Szeliski's[2]. For a camera centered at the origin, the relationship between a 3D point $\mathbf{p} = (X, Y, Z)$ and its image coordinates $\mathbf{x} = (x, y, 1)$ can be described by

$$\mathbf{x} \sim \mathbf{VRp}, \tag{1}$$

where

$$\mathbf{V} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{and } \mathbf{R} = \left[r_{ij} \right], \tag{2}$$

are focal length scaling, and 3D rotation matrices. For simplicity, we assume the focal length l is known and is the same for all images, i.e, $\mathbf{V_k} = \mathbf{V}$.

The mapping between two images k and l is given by

$$\mathbf{M} \sim \mathbf{V R_k R_l^{-1} V^{-1}}. \tag{3}$$

To correct the mapping, we perform an incremental update to $\mathbf{R_k}$ based on the angular velocity until the error between two images is minimized. The error is estimated by calculating cross-correlation in corresponding area between two images,

$$E(\mathbf{M}) = \sum_i \frac{(I'_k(\mathbf{x}_i) - \bar{I}'_k)(I'_l(\mathbf{x}_i) - \bar{I}'_l)}{\sigma_k \sigma_l} \tag{4}$$

where I' is the corresponding area between two images after projection onto common coordinate, $\bar{I}'$ is the intensity average of I' and σ is the standard deviation of I'.

The texture coordinates are calculated by the combination of the translation matrices $\mathbf{M}$ and the lens distortion formula,

$$\begin{cases} x'_i = x_i + \gamma_x (x_i - x_{ic}) r_i^2 \\ y'_i = y_i + \gamma_y (y_i - y_{ic}) r_i^2 \end{cases}, \tag{5}$$

where (x_i, y_i) and (x'_i, y'_i) are the coordinates before and after correction, (x_{ic}, y_{ic}) is the image center, r_i is the distance from the image center to (x_i, y_i), and γ is the correction intensity. After calculating the texture coordinates, these are set on the decoded frame(Figure.4).

The calculated parameters are used to be initializing parameters of next frame. The parameters of the very first frame is manually preset based on the assumption that the each cameras are positioned on a different side of a square and has a viewing angle width of about 100°. The mosaicking process runs in parallel with frame decoding and rendering by threading technique, so as to avoid dropping frames while mosaicking is begin completed. It is also possible that the mosaicking parameters might not be updated if the mosaicking process could not completed in time, but this may not be a significant problem because sequencial frames are so close.

3 Implementation

We implemented prototypes of both the capturing and the viewing subsystems of PanoVi. The capturing subsystem uses four miniature, wide-angle cameras each mounted on a different side of a square measuring 2.5c per side. The viewing system is designed to run on standard PC with PentiumIII or faster CPU and mid-range 3D graphics accelerator. We developed the system for Microsoft Windows 2000 or 98 operating systems. The frame rate depends on the format

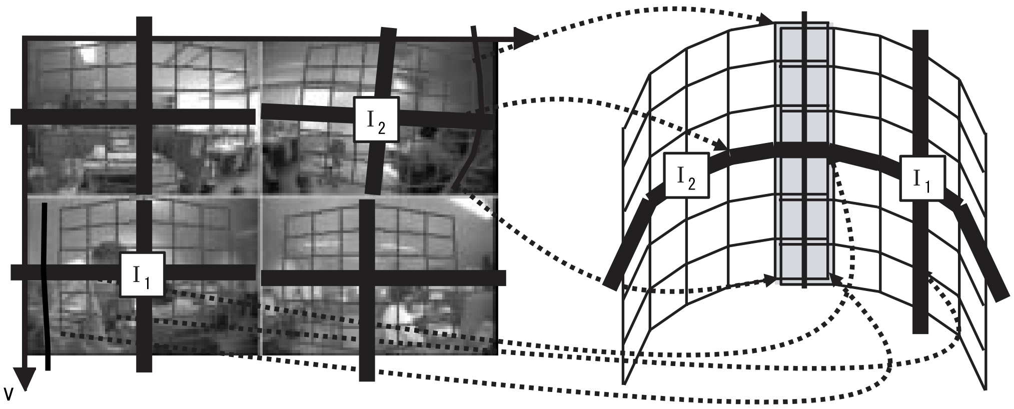

Fig.4 Movie texture with texture coordinates.

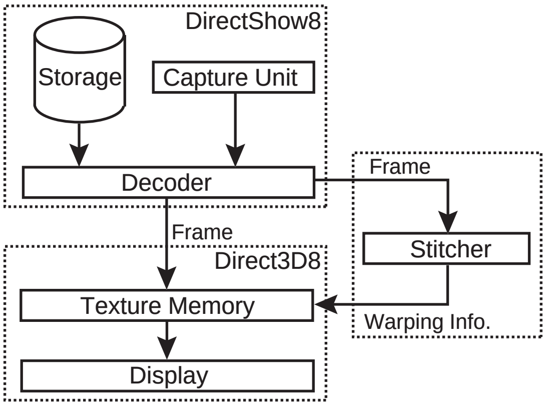

Fig.5 Block chart of PanoVi viewer client.

used for transmission. The frame rate may decrease if high-resolution formats or exotic compression formats are used, because such formats draw heavily on CPU resources. We tested the three implementation cases described below.

Case 1: For live telecast of panoramic movies to PC clients, the system uses DV, a video encoding format that is readily available. The DV format has 720×480 resolution and plays at 29.97 fps. It is decoded sufficiently fast using PC hardware, but the image size may be so great to occupy much of the memory transfer bandwidth. In this case, a high performance CPU and wide bandwidth memory architecture such as Pentium4 with RDRAM system faciliate full-speed operation.

Case 2: For playback from a packet medium, the system uses compressed movie formats such as MPEG. MPEG4 is used to achieve higher compression; however, it requires very complex processing to decode. Because in MPEG formats the frame size can be selected, we can reduce frame size to 320×240 pixels, to obtain a frame rate high enough to achieve quality that is expected to be equivalent to NTSC. In this configuration, the viewer will work in satisfactory performance on a PC equiapped with processors of the PentiumIII, 1.0GHz class.

Decoding and rendering of the viewing software is done using Microsoft DirectX8 API(Figure 5). The system decodes video streams using DirectShow8 API, and copies the decoded frame image to texture memory on the graphics accelerator chip. Then, the system renders the virtual environment of a cube model with movie texture using Direct3D8 API. Since PanoVi decodes video stream using CODECs supported by Windows, we can utilize a new CODEC immediately once Windows supports it.

Figure 6 shows snapshots of the overall the PanoVi prototype system and a camera variation. Figure 7 shows snapshots of the viewer windows playing a movie. A user can change the viewing angle arbitrarily by mouse operation and can zoom in and out by keyboard operation.

4 Conclusions

This paper introduced the PanoVi, a new panoramic movie system that can capture and play a 360° view in real time. Because PanoVi uses standard video format in the transmission layer, it is easily applied to existing techniques, such as network, digital/analog broadcast and compressed movie package. The capturing and the viewing subsystems are constructed using only standard hardware, so PanoVi can be extended as hardware evolves.

5 Future Works

When playing a movie that has large depth of field, the mosaicking may become inaccurate. To remedy this problem, we can adjust the parameters of the mosaicking process adaptively, because it becomes possible to estimate depth of subject by estimating optical-flow between two sequential frames. PanoVi uses very-wide-angle lens that causes heavy geometric and optical distortions, whose compensation is a serious problem to be solved. Although the main applications of panorama movies are entertainment, the technology has the potential for more practical applications.

Fig.6 The prototype system and a camera variation.

Fig.7 Snapshots of the viewer windows playing a movie.

Acknowledgements: The authors wish to express their thanks to the Telecommunications Advancement Organization of Japan(TAO) for its generous support.

Reference:

[1] S.E.Chen, "QuickTime VR - An Image-Based Approach to Virtual Environment Navigation," In *Proceedings of SIGGRAPH '95*, pp.29-38, 1995, ACM.

[2] R.Szeliski and Heung-Yeung Shum, "Creating Full View Panoramic Image Mosaics and Environment Maps," In *Proceedings of SIGGRAPH '97*, pp.251-257, 1997, ACM.

[3] N.Chiba, H.Kano, M.Ninoh and M.Yasuda, "Feature-Based Image Mosaicing," In *The Transactions of IEICE*, Vol.J82-D-II No.10, pp.1581-1589, 1999.

[4] X.Qin, K.Tadamura, T.Ishibashi and E.Nakamae, "Composite Panned Landscape Video Sequences with Computer Generated Still Images," In *The Transactions of IEICE*, Vol. J83-D-II No.4, pp.1140-1152, 2000.

[5] J.Hoshino and I.Masuda, "A Robust Technique for Creating Panoramic Images against Local Intensity Variations," In *The Transactions of IEICE*, Vol. J82-D-II No.2, pp.222-229, 1999.

[6] J.Hoshino and I.Masuda, "Creating Panoramic Images by Minimizing Errors between Observed and Estimated Images," In *The Transactions of IEICE*, Vol. J81-D-II No.11, pp.2565-2573, 1998.

[7] T.Yamamoto and M.Doi, "PanoVi:PANORAMIC MOVIE SYSTEM FOR REAL-TIME NETWORK TRANSMISSION," In *IEEE Fourth Workshop on Multimedia Signal Processing*, pp.389-394, 2001.

[8] M.Doi and T.Yamamoto, "A Development of Movie Browser for Seamlessly Connected Panoramic Movies Taken by Multi-camera Configuration," In *Proceedings of the 2001 IEICE general Conference*, D-11-89, 2001.

[9] M.Doi and T.Yamamoto, "A Development of Real-Time Transmission and Browsing Systems Using Standard Video Format for Seamlessly Connected Panoramic Movies," In *Proceedings of the 2001 Information and Systems Society Conference of IEICE*, D-11-50, 2001.

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

[11] Fuji Xerox Co. Ltd., http://www.ubiquitous-media.com/flycam.html