Panoramic Video Representation using Mosaic Image

Kam-sum LEE Yiu-fai FUNG Kin-hong WONG Siu-hang OR Tze-kin LAO
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong SAR, China

Abstract ***This paper presents a novel approach of video representation with panoramic techniques for low bit-rate video transmission. Using a background panorama as a prior information, foreground objects in a video sequence are separated from the background through a series of segmentation processes. These foreground segments are encoded by traditional compression technique and transmitted as a video stream, while the scene background is transmitted only once as a panorama. To reconstruct the original video frame, foreground objects are combined with the corresponding panoramic segment on-the-fly at the receiving side. Experiments show that our approach improves the compression performance, compared with MPEG-1 under the same quality factor. Our system can synthesize virtual environments without using blue-screen. The users can navigate throughout the scene or examine any particular details. Our system also provides an effective solution to scene-based video indexing.***

Keywords: panorama mosaic, video coding and compression, video indexing, image segmentation and registration

1 Introduction

There has been a growing interest in the use of mosaic images as a basis for efficient representation of video sequences rather than simply as a visualization device [7]. As successive frames in a video sequence usually overlap by a large amount, mosaic images often provide a significant reduction in the total amount of data needed to represent the scene. In block-based coding system, an image is divided into a 2D array of blocks. Among these blocks, the translational motion between successive frames is estimated. Representing the motion information by a block-wise description, data compression can be achieved by storing the limited amount of motion data. However, the moving objects usually do not fall within these blocks and the motion coherence thus extends beyond the blocks.

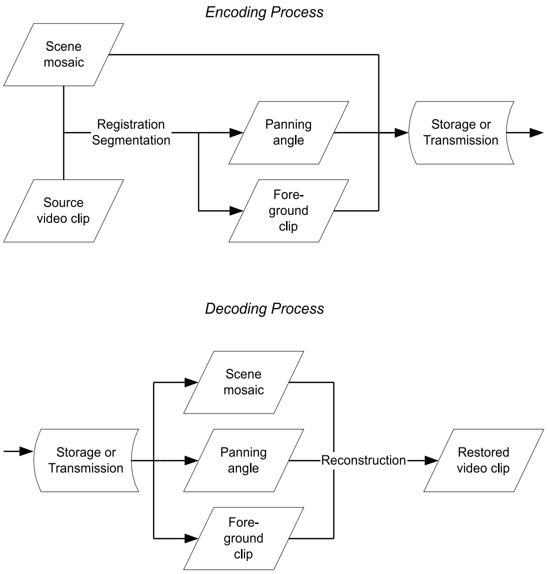

Figure 1: Panoramic video coding process

Hence, our focus is to reduce the redundancy by improving the determination of the coher-

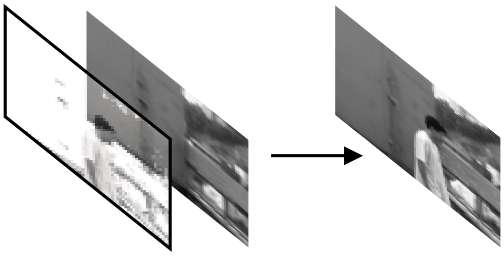

Figure 2: Layered representation

ent motion regions. These motion regions can be considered as some "moving" objects, relative to the background scene. Obviously, if we can have some prior knowledge of the background scene, it will be very useful in solving our problem. We describe a new coding scheme based on a layering concept as shown in Fig.2: a foreground layer with several moving objects on top of a stationary background panorama. A background scene mosaic is constructed first. For each frame, the foreground regions are segmented and registered. The two layers are handled separately during transmission or in storage until reconstruction at user-end. To simplify the problem, we assume that the camera position is fixed and its movement is limited to horizontal rotation (panning) only during a video stream.

Many researchers have been working on the use of mosaic images to represent the information contained in video sequences. Irani *et al.* [3] described two different types of mosaics, static and dynamic, that are suitable for storage and transmission applications respectively. Based on this categorization, they proposed a series of extensions to these basic mosaic models to provide representations at multiple spatial and temporal resolutions, and discussed a mosaic-based video compression technique. On the other hand, Hsu and Anandan [2] coined the term Mosaic-Based Compression (MBC), and described several kinds of hierarchical representations (temporal pyramids) suitable for MBC to reduce redundancy in video data.

This paper is organized as follows. Section 2 discusses the construction of background panorama mosaic from a set of camera images. The techniques and mathematical issues for foreground object segmentation and registration are described in section 3. In section 4, the reconstruction of video streams from foreground segments and background panorama is explained. Section 5 consists of experimental results together with discussions on algorithm improvements, and finally the conclusion and future directions are given in section 6. Fig.1 shows an overview of our entire system.

2 Mosaic Construction

2.1 Panorama Mosaic

In recent years, a number of techniques and software systems have been developed for capturing panoramic images of real-world scenes. In particular, Chen [1] has developed a less hardware-intensive method with only regular photographic frames over the whole viewing space. As discussed in [8], the first step in building a full view panorama is to map 3D world coordinates (x, y, z) onto 2D panoramic screen coordinates (θ, v) with cylindrical projection:

$$\theta = tan^{-1}(x/z), v = y/\sqrt{x^2 + z^2} \qquad (1)$$

where θ is the panning angle and v is the scanline. Once we have wrapped all the frames in a scene sequence, constructing mosaic images becomes a pure frame alignment problem, with minor compensations for vertical jitter and optical twist. Various 2D or 3D parametric motion transformations [9] have been suggested to cancel out the effect of camera motion and combined component frames into complete panoramic images. In our current implementation, Live Picture PhotoVista^{TM} was used to generate cylindrical mosaic images from 2D environment snapshots. Only the information about horizontal translation t_x and vertical translation t_y for each input image were fed into the "stitching" algorithm, so that it would estimate the incremental translational $\delta\mathbf{t} = (\delta t_x, \delta t_y)$ by minimizing the intensity error $E(\delta\mathbf{t})$ between two images. Fig.3

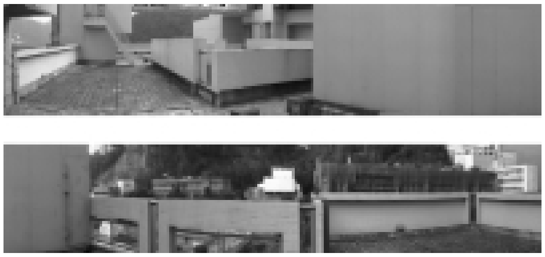

Figure 3: Panorama mosaic sections

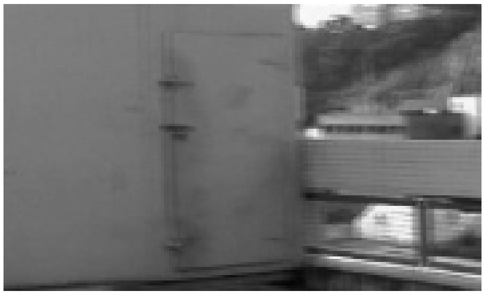

Figure 4: Background from panorama

shows mosaic segments constructed in our experiment.

2.2 Cylindrical Projection

Once the construction of the mosaic image is completed, it can be displayed with a special purpose viewer like QuickTime VR^{TM} [1]. The mosaic image is actually wrapped onto a sphere or cylinder surface using texture-mapping. Every time a user looks through the panoramic viewer, not the whole panoramic image is visible on the image plane and only a portion of it is displayed. The bounding rectangle of this sub-texture is called texture window. Under full-perspective projection model and with the knowledge of current viewing parameters, we can find the exact coordinates of the current texture window by projecting several points in the image plane onto the cylindrical surface and bounding the projected shape with a rectangle. The viewing parameters include the view vector, field of view, aspect ratio, size of panoramic cylinder, etc.

3 Foreground Segmentation and Registration

To perform segmentation and registration of foreground objects, we have to estimate the camera rotation throughout the video stream, i.e. the incremental changes in panning view angle of the panorama with respect to each frame. A video frame can be considered as a mixture of background scene and foreground objects. As foreground objects are absent in the panorama, global image processing techniques like difference map cannot be applied directly. Instead, we use some small block templates on the background region to perform local processing over the entire frame. As a first step, we adjust the horizontal and vertical panoramic view angles to suit the size of the frame. We use $I_{frame}(i)$ to denote the i^{th} frame of the source video stream and $I_{pano}(\phi)$ to denote the viewing window of the panorama at panning view angle ϕ. Fig.4 shows a scene from the background panorama used in our experiment.

For the first frame in a sequence, some small block regions on scene background, denoted as template region $TR(I_{frame}(0))$, are selected through user interaction. Depending on the frame resolution, at least one block with size variable from 5×5 to 10×10 should be selected, while more blocks would provide better results at the expense of longer execution time. $TR(I_{frame}(0))$ should include distinct edges or corners on the background scene, and must not be occluded by any foreground objects. During the processing of video stream, these template regions should be monitored to prevent occlusion. A new set of $TR(I_{frame}(0))$ should be reselected in case of occlusion. Taking the first frame to have a panning view angle ϕ_0 of 0, we have a minimization problem of E_i in the HSI color space:

$$E_i(\phi_i) = [TR(I_{frame}(i)) - TR(I_{pano}(\phi_i))]^2 \tag{2}$$

where ϕ_i is the new panning view angle of the panorama at the i^{th} frame following a small

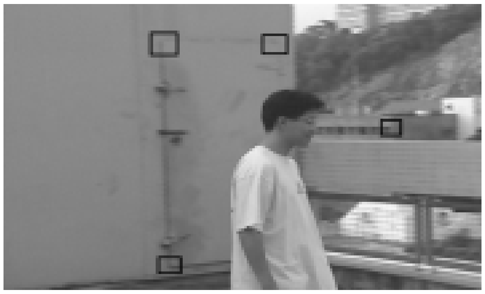

Figure 5: Video frame with template blocks

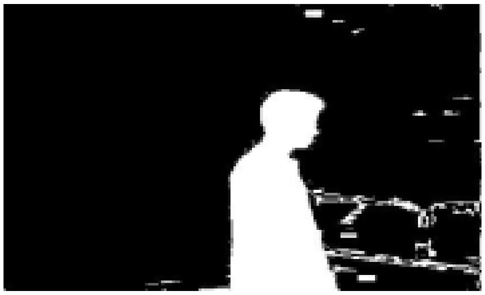

Figure 6: Alpha map

update $\delta\phi_{i-1,i}$:

$$\phi_i = \phi_{i-1} + \delta\phi_{i-1,i} \tag{3}$$

At an optimal $\delta\phi_{i-1,i}$, the difference between video frame and panoramic view would be minimized. Fig.5 shows a video frame with template blocks indicated by rectangles. With the normal frame rate of 20-30 fps in typical video sequences, the motion between two consecutive frames would be very small under practical panning speed. For example, the average difference in panning angle between two consecutive frames is only 0.5 degree for an angular velocity of 15 degrees per second and frame rate at 30 fps. With this simplification, we can apply a linear search algorithm to find an estimate of $\delta\phi_{i-1,i}$, which is assumed to fall within -1.0 to +1.0 degree.

To segment foreground object information from current frame $I_{frame}(i)$ and panorama $I_{pano}(\phi_i)$, we define a binary alpha map α_i in which elements may be 0 (black) or 1 (white):

$$\alpha_i = I_{pano}(\phi_i) \oplus I_{frame}(i) \tag{4}$$

Fig.6 shows the alpha map obtained from Fig.5. Elements in black denote the matching areas between the video frame and panorama view, while white areas represent moving objects in foreground that should be encoded separately from the background scene. Owing to the inherent noise in real images, there will inevitably be some isolated small spots (both in black or white) in the alpha map. Since they do not carry much information for further processing, they will be removed by size-filtering before segmentation. The foreground objects $I_{fore}(i)$ are thus extracted by:

$$I_{fore}(i) = \alpha_i \odot I_{frame}(i) \tag{5}$$

where $\odot$ is the element-wise multiplication. The resulting $I_{fore}(i)$ contains foreground object regions and all other areas that are white in the alpha map, and will be used to register the changes in the corresponding panoramic panning view angle $\delta\phi_{i-1,i}$. Fig.7 shows the extracted foreground regions from Fig.6 and Fig.5. However, instead of storing every pair of $\delta\phi_{i-1,i}$ and $I_{fore}(i)$, we only record the subtotal change in panning view angle $\delta\phi_{i\rightarrow i+n-1}$ for every n frames to save storage space:

$$\delta\phi_{i\rightarrow i+n-1} = \sum_{j=i}^{i+n-1} (\delta\phi_{j-1,j}) \tag{6}$$

The value of n depends on the panorama panning speed. For a fast changing video section with large values of $\delta\phi_{i-1,i}$, then n should be smaller. An upper bound on $\delta\phi_{i\rightarrow i+n-1}$ is imposed to prevent over-smoothing during the reconstruction of video streams. The frame sequence of foreground segments will be compressed by MPEG-1, and have a much smaller size than the original sequence under the same compression. Further details will be discussed in section 5.

4 Video Reconstruction

Now we have three separated objects as a result of scene decomposition for every

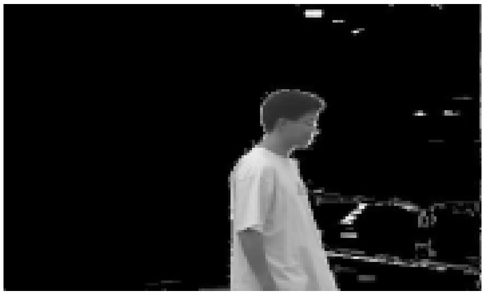

Figure 7: Extracted foreground regions

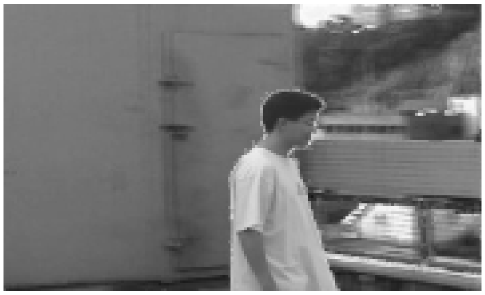

Figure 8: Reconstructed video frame

n frames: background panorama $I_{pano}(\phi_i)$, frames of foreground object segments $I_{fore}(i)$, and changes in panning view angle $\phi_{i\to i+n-1}$. Taking them as input, a special viewer is used to decode the video stream and reconstruct the original frame sequence. Let's consider the reconstruction of background scene first. Given the subtotal change in panning view angle $\delta\phi_{i\to i+n-1}$ for every n frames, the viewer select an appropriate background scene $I_{pano}(\phi_i)$ for each frame by performing a linear interpolation on $\widetilde{\delta\phi_i}$ to generate smooth viewpoint transition:

$$\widetilde{\delta\phi_i} = \frac{1}{n}\delta\phi_{i\to i+n-1} \qquad (7)$$

After that the viewer can simply decode and render the foreground object segments $I_{fore}(i)$ over the background scene from panorama to reconstruct an approximated original frame. Fig.8 shows the resulting video frame of the reconstruction from Fig.7 and Fig.3.

Our system provides a simple and effective solution for video indexing. In traditional coding methods, the search of a certain frame or video clip can be done only sequentially using the time or frame as index. In our system, since every frame is registered by the relative panning angle with respect to the background mosaic, a user can access a specific frame by providing the scene information, i.e., indexing through various panning angle ϕ. This approach is a complement to the content-based (color and texture) indexing method but easier and more efficient to implement.

5 Experiments and Discussion

A digital video camera with resolution 720 × 480 in pixels was used to capture outdoor images for the construction of a background panorama and a testing video sequence. Fig.10 shows another resulting frame of our system. The use of panorama mosaic and extraction of foreground regions provide a higher compression performance. In our system, only the foreground regions are stored in the frame sequence. They are considered to be the coherent motion regions. During the MPEG-1 compression, a frame of foreground regions can be compressed with a higher ratio than the original frame. In the inter-picture coding of 'P/B' frames, since the background regions are removed and only the motion information (temporal information) of the foreground regions is involved, the run-length (RLC) / variable-length (VLC) encoded and quantized DCT coefficients will be smaller than those of the original complete frames.

In the intra-picture coding of 'I' frames, since the background regions are removed, the frames of foreground regions will contain less spatial information, thus also have a smaller set of RLC / VLC encoded and quantized DCT coefficients. The relative compression gain in this process will be higher with a smaller size of foreground regions and a higher complexity of background regions. As an example in Fig.9, we compare three JPEG compressed pictures with different background complexity.

First, as shown in the Table 1, we can easily observe that the foreground frames of the

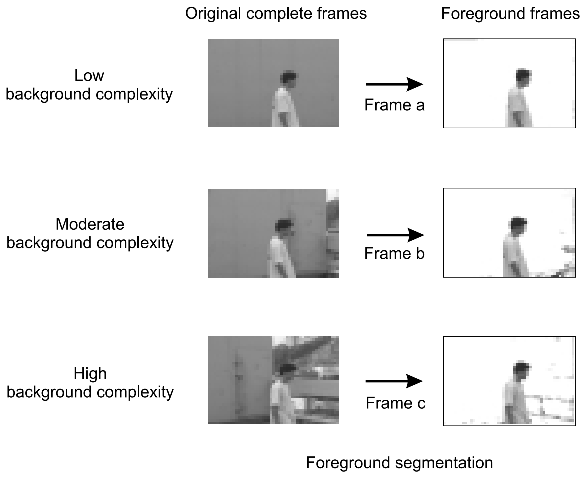

Figure 9: Different frames under JPEG

Table 1: Intra-picture coding performance

Size kb	$\lvert I_{ori}\rvert$	$\lvert I_{fore}\rvert$	$\lvert I_{ori}\rvert : \lvert I_{fore}\rvert$
Frame a	11.5	5.1	1 : 0.44
Frame b	17.8	7.4	1 : 0.42
Frame c	23.6	8.8	1 : 0.37

Table 2: Storage size

Items	Size kb
Original source $\lvert V_u\rvert$	993
MPEG-coded source $\lvert V_{MPEG}\rvert$	210
Mosaic image $\lvert I_{pano}\rvert$	22
MPEG-coded fore-clip $\lvert V_f\rvert$	85

three pictures are compressed with higher ratio than their original complete frames. Moreover, it is obvious that the picture with more complicated background regions is having a higher compression gain under foreground extraction than the others. This shows that our system will perform better in those video clips with more complicated background scenes.

Table 2 shows the resulting storage sizes of different components involved in our system. The real video clip contains 60 frames in two seconds. A partial mosaic image of the background scene is used in the experiment. The total storage size needed in our approach is the sum of the size of the mosaic image and the size of the MPEG-1 coded foreground clip:

$$|V_{pano}| = |V_f| + |I_{pano}| = 107kb \tag{8}$$

Then the compression ratio is:

$$CR_{pano} = 1 - \frac{|V_{pano}|}{|V_u|} = 89\% \tag{9}$$

We can observe that the size needed is reduced by about 89%, compared with the size of the original uncompressed video clip. We also made a comparison with that obtained from MPEG-1 compression. The extracted foreground frames and the original video clip are both compressed by MPEG-1 under the same quality factor and control parameters. The ratio between them is:

$$|V_{MPEG}| : |V_{pano}| = 210 : 107 \approx 2 : 1 \tag{10}$$

Our system achieved a nearly 50% size reduction over traditional MPEG-1 compression. Moreover, for a longer video clip, the overhead of the size of the mosaic image is relatively small and can be neglected. This results in a better compression ratio.

In our current implementation, limitations include the tracking algorithm makes use of template blocks, which require human interaction. Moreover, the effectiveness of compression depends on the accuracy of segmentation results, which drops for regions of similar colors and patterns between background and foreground. The reduction in size of a single frame ranges from about 10% to 75% for different frames in our experiment. Apart from reconstructing the original video stream, the viewer can also provide some interesting features, like interactive controls on panoramic panning view angle and zoom factor, to explore the whole scene or examine details of any particular frame. Moreover, by replacing the original panorama, we can even synthesize various virtual environments. To enhance the power of our system further, we may allow zooming and vertical panning of camera motion during the capture of the video stream. However, these

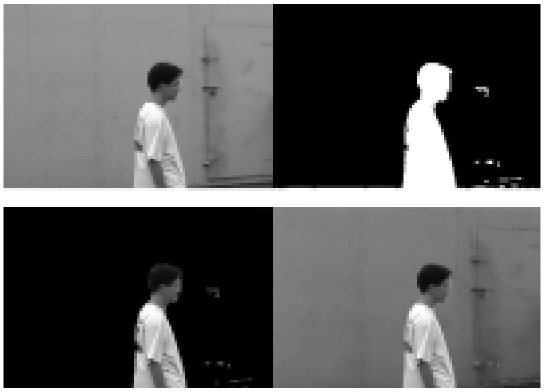

Figure 10: Another experiment result

modifications will lead to problems in the estimation of zooming factor and the vertical panning angle of the camera, and will be studied in greater depth as an extension to this work.

6 Conclusion and Future Direction

Our system provides a new method of video representation for very low bit-rate transmission. The video stream is decomposed and represented as a combination of background panorama and foreground objects. A panorama mosaic is first constructed to depict the scene background, and foreground objects in the source video are then extracted out by panoramic-based segmentation. A tailor-made viewer will then combine the foreground segments with their corresponding views in the background panorama to synthesize the original video frames. The bandwidth requirement for video sequence transmission in this new scheme would be much smaller compared with existing methodologies. Our system can synthesize virtual environments without using blue-screen. The users can navigate throughout the scene or examine any interested details. Our system also provides an effective solution to scene-based video indexing. Improvements in tracking and segmentation algorithm, estimation of zoom factor, and degrees of freedom in the camera motion are some of the interesting topics to be studied in the future.

Acknowledgment

This research was supported by The Chinese University of Hong Kong direct grant.

References

[1] S.E. Chen, "QuickTime VR - An Image-based Approach to Virtual Environment Navigation", SIGGRAPH '95, pp. 29-38.

[2] S. Hsu and P. Anandan, "Hierarchical Representations for Mosaic Based Video Compression", Proc. Picture Coding Symp., pp. 395-400, Mar. 1996.

[3] M. Irani, P. Anandan and S. Hsu, "Mosaic Based Representations of Video Sequences and Their Applications", Proc. of ICCV '95, pp. 605-611, Jun. 1995.

[4] M. Irani, S. Hsu and P. Anandan, "Video Compression Using Mosaic Representations", Signal Processing: Image Communication, 7:529-552, 1995.

[5] M.C. Lee *et al*, "A Layered Video Object Coding System Using Sprite and Affine Motion Model", IEEE Trans. on Circuits and Systems for Video Technology, 7(1):130-145, Feb. 1997.

[6] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system", SIGGRAPH'95, pp. 39-46, August 1995.

[7] R. Szeliski, "Image Mosaicing for Telereality Applications", Technical Report CRL 94/2, Digital Equipment Corp., 1994.

[8] R. Szeliski, "Video Mosaics for Virtual Environments", IEEE Computer Graphics and Applications, pp. 22-30, Mar 1996.

[9] R. Szeliski and H.Y. Shum, "Creating Full View Panoramic Image Mosaics and Environment Maps", SIGGRAPH '97, pp. 251-258, Aug. 1997.