SPRITE PYRAMID FOR VIDEOS AND IMAGES HAVING FINITE-DEPTH SCENES

Ramazan Savaş Aygün

Computer Science Department
University of Alabama in Huntsville
email:raygun@cs.uah.edu

Aidong Zhang

Dept. of Computer Science and Engineering
State University of New York at Buffalo
email:azhang@cse.buffalo.edu

ABSTRACT

The ordinary sprite generation techniques focus on camera movement, accurate motion estimation, alignment, and integration. These techniques ignore the resolution of original images and the regenerated images from the sprite are likely to have lower resolutions than the original ones. Especially, if the scenes have finite depth and zoom-in and zoom-out operations occur, the segments of the scene are captured at different resolutions. The traditional mosaic generation methods either blur the mosaic by integrating lower resolution segments or use unnecessary large storage for the mosaic. The *sprite pyramid (or layered sprite)* allows efficient storage of images or video clips of overlapping scenes at different resolutions. Moreover, the images or video frames can be reconstructed from the sprite pyramid at the necessary resolutions.

1. INTRODUCTION

The introduction of the MPEG-4 [1] video standard has motivated many research fields like object segmentation and sprite generation. The sprite generation has initially been studied as *mosaic generation* [2, 3, 4, 5]. Mosaic presents a wide picture of the environment that cannot be captured in a single frame. Mosaics are mostly used in video summaries and retrieval. The previous approaches have not considered the accurate generation and efficient storage of the mosaic. Since the sprite generation methods are based on mosaic generation techniques, the sprite generation is a lossy process.

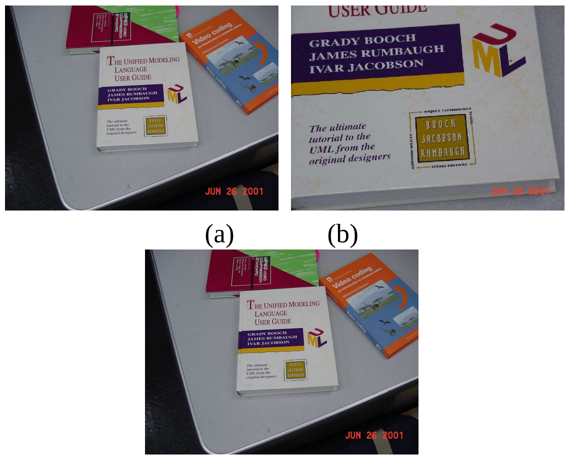
Different representations of mosaics like static, dynamic and synopsis mosaic have been investigated in [2]. A direct method is used to align images and to generate the mosaic. In [3], the extracted corners from images are also used for mosaic generation. A sprite creation method based on connected operators is presented in [6]. The image is represented as flat zones and pruned using a max-tree representation. There are different types of mosaic depending on the camera motion. If the camera is translating sideways, a planar mosaic is generated [2, 5]. The cylindrical camera is used for the panning camera [3]. The spherical mosaic is generated when camera is both panning and tilting [4]. These methods do not consider forward motion or zooming of the camera. In [7], the forward motion of the camera is modeled and the mosaic is generated using the *pipe* projection. For example, the mosaics are effectively generated shots taken from a plane or a car moving in the forward direction. In this kind of videos, the depth of the scene can be considered as infinite and projecting thin strips from images onto manifolds is effective in the mosaic generation. If the scene depth is finite, another method has to be applied since an image is already included in another image. If there is a zoom in a closed environment as in distance learning applications, the pipe projection cannot be applied properly.

In traditional mosaicing methods, mosaics are generated by mapping onto a predetermined single space. The order of images are important in mosaic generation. In most cases, the images are mapped according to the first image in the sequence. If the first image has the lowest resolution, then a low resolution mosaic is generated and if the first image has the highest resolution, a high resolution mosaic is generated. In the first case, if the images are generated from the mosaic, they have have lower resolutions than those of the originals. For example, Figure 1 gives an example of such an image alignment where the first image has a lower resolution. Figure 2 shows an example of image regeneration from a low resolution mosaic. If the first image has high resolution, after the images are aligned, the final mosaic becomes huge to reserve the resolution of the first image. The goal is to provide a method to generate mosaic without losing resolution while maintaining efficient storage.

A *sprite pyramid (or layered sprite)* allows the regeneration of the video at the proper resolutions. Each layer of the sprite pyramid corresponds to a different resolution. The sprite pyramid allows the regeneration of different segments at different resolutions as they were captured. The sprite pyramid is created if the scene has finite depth and there are zoom-in and zoom-out operations.

The rest of the paper is organized as follows. Section

THIS RESEARCH IS SUPPORTED BY NSF GRANT IIS-9733730 AND NSF GRANT IIS-9905603.



(a) (b)

(c)

Figure 1: Image alignment for different resolutions: (a) first image (b) new image (c) mosaic

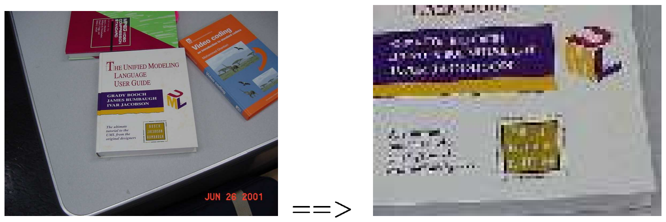


Figure 2: Regeneration.

2 explains the structure of the sprite pyramid. Section 3 discusses our experiments on a distance education video. The last section concludes the paper.

2. SPRITE PYRAMID

The sprite should include every section that is visible throughout the video sequence. If there is no a priori motion information for a video sequence, the motion has to be estimated between each sequential frame. We have used the high resolution mosaic generation algorith that is described in [8] to reduce the blurring.

A sprite pyramid consists of L layers $(0 \leq l < L)$, where the lowest layer contains the highest resolution and the highest layer contains the lowest resolution. Laplacian pyramid [9] is a hierarchical way of representing an image usually at low resolutions at the high levels and high resolutions at the low levels. Each layer contains the same image at different resolutions. Since the sprite is not viewable at all resolutions, some layers of the sprite pyramid may contain images having holes. Irregular shapes occurring as holes are caused by the rotation of the camera. A hole also occurs when a segment of an object is not captured at that resolution. Existence of the holes is the main difference from

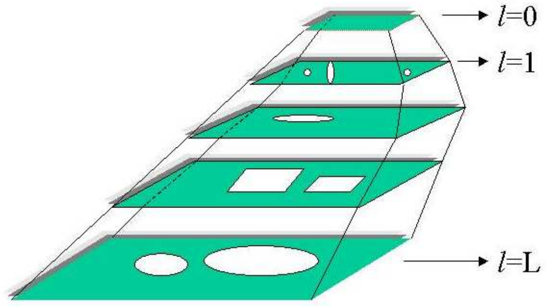


Figure 3: The sprite pyramid.

traditional image pyramids used in the literature where each layer contains an image at different resolution. Each layer l of pyramid σ has a zooming factor σ_l. The structure of a sprite pyramid is shown in Figure 3. The advantage of this sprite pyramid is that it keeps all the data visible at its resolution. So, when a video frame or image has to be regenerated, the video object is generated from the corresponding layer having the same zooming factor.

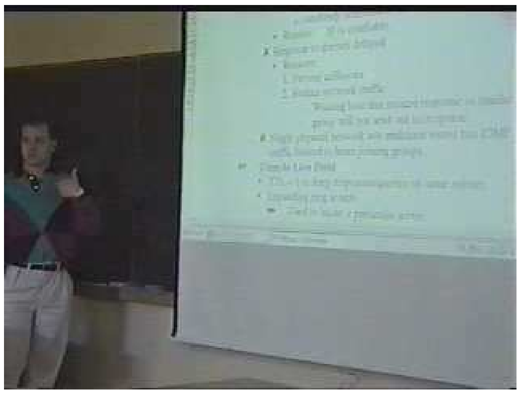
The generation of sprite pyramid from a group of images can be performed efficiently since the number of images are few or the camera motion is not contiguous. In a video, the camera motion is usually continuous. Creating a layer for each frame is time consuming and not efficient. There are two factors: the magnitude of zooming factor and speed of zooming. If zooming is not significant, frames are mapped onto the current layer. If zooming is greater than 1 and significant, it is mapped onto the lower layer. Otherwise, it is mapped onto the upper layer. If the storage and performance is not important, the frames can also be mapped to each layer. This eliminates the holes in the sprite pyramid.

In most cases, the reason of zoom-in is to focus on the interesting object in the scene. Therefore, mapping can be ignored until the zooming operation stops. In that case, there is an interesting object and that scene has to be kept at high resolution. If new parts of the scene are visible during zooming, those regions are mapped onto the current layer of the sprite.

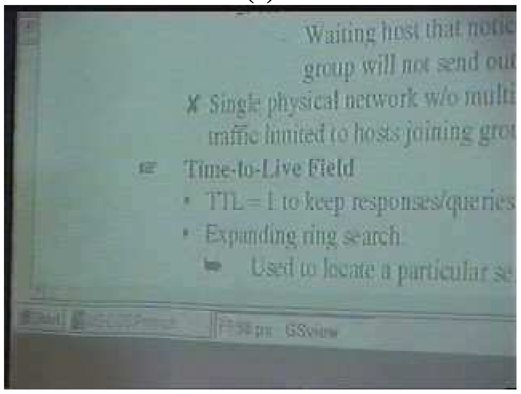
All the images are aligned at their own layers. The images are also aligned at the lowest resolution. At level 0, the mosaic contains the largest view of the scene. This mosaic may also be used to display the big picture of the object.

3. EXPERIMENTS

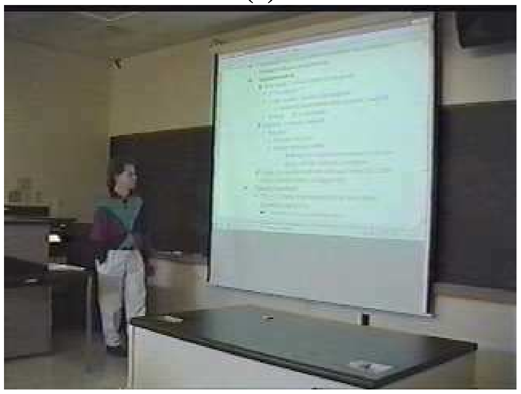
To show the effectiveness of the sprite pyramid, we give an example from a distance education application. Distance education lectures contain zoom-in and zoom-out operations. Zoom-in usually happens when the instructor points



(a)


Waiting host that notic
group will not send out
✗ Single physical network w/o multi
traffic limited to hosts joining gro
Time-to-Live Field
TTL = 1 to keep responses/queries
Expanding ring search
Used to locate a particular se

(b)



(c)

Figure 4: Zoom operations in a lecture (a) frame 32400 (before zoom-in) (b) frame 32550 (after zoom-in) (c) frame 32900 (after zoom-out)

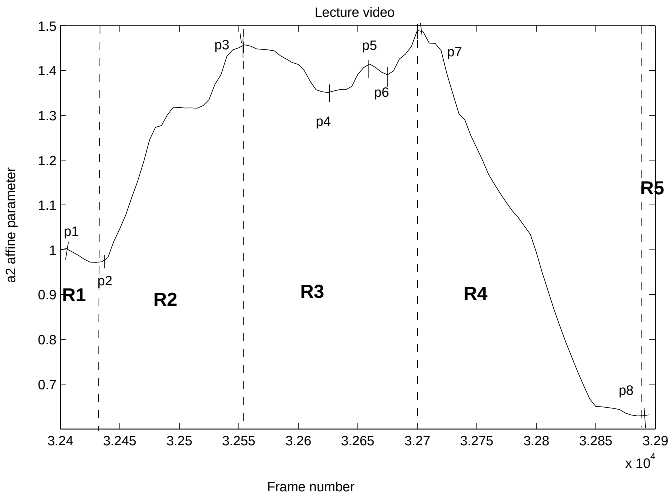


Figure 5: Affine motion parameter a_2 from a video clip of a distance education video.

an important data on the board or the slide show. When zoom-in happens, important data are displayed. When zoom-out happens, the general view is presented. Figure 4 shows three scenes from the lecture before zoom-in, after zoom-in and after zoom-out.

Figure 5 displays the a_2 parameter of affine motion model with respect to the initial frame in the clip. It is about 20 seconds clip starting from frame 32400 to frame 32900. We only present a_2 since a_2 is nearly equal to a_5; and a_3 and a_5 is very close to 0. In the figure, p_1, p_3, p_5, and p_7 are peak points (local maxima). At these points, zoom-in reaches its final point. These are the possible points that the significant object is being captured at the required resolution. In the figure, p_2, p_4, p_6, and p_8 are the local minima where the zoom-out stops. In region R_2, there is a continuous zoom-in, so this part can be ignored in sprite generation. In R_4, there is a continuous significant zoom-out. In region R_3, p_5 is very close to the neighboring local minima, i.e., p_4 and p_6. There is no significant zoom-in operation at this part. In region R_3, p_4 and p_6 are very close to their neighboring local maxima. Therefore, there is no significant zoom-out. Region R_3 focuses on the important object in the scene. In regions R_1 and R_5, the general view of the environment is shown in the clip.

If the sprite pyramid has three layers, sprites are generated for R_1, R_3, and R_5, and form the layers of the sprite. In traditional sprite generation, single sprite would be generated. Since temporal integration is performed at different resolutions, the resolution of the sprites is not degraded (blurred) by integration of lower resolution frames. In Figure 5, a_2 at p_7 is nearly three times of a_2 at p_8. If the sprite were generated, the final sprite would be 9 (3x3) times larger than the original frame. In this case, there are three

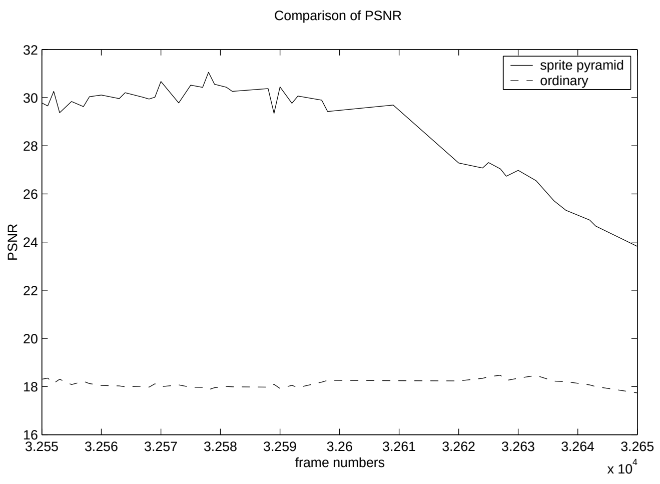


Figure 6: Comparison of PSNR values for ordinary sprite generation and from sprite pyramid.

layers of sprite: for R_1, R_3, and R_5. R_2 and R_4 only contributes to the sprite if they cover some segments that are not covered in R_1, R_3, and R_5. Since frames in R_1, R_2, R_4, and R_5 are not integrated on frames in R_3, the sprite for R_3 is not blurred by low resolution data integration.

During our experiments, we have faced problems when using traditional mosaic generation methods. Sprite generation from the whole sequence is erroneous since the image (at high resolution) is considered as a pattern in a low resolution image. During error computation and motion parameter estimation, there are candidate regions in low resolution image which give less error than the original region. In that case, sprite cannot be generated properly and long-term sprite generation methods [10] fail. On the other hand, frame-based motion parameter estimation yields error accumulation in sprite generation. Figure 6 shows the comparison of PSNR values for the high resolution sequence of a video by using sprite pyramid and an ordinary sprite generation technique. This figure shows the efficiency of using sprite pyramid.

4. CONCLUSION AND FUTURE WORK

In this paper, we presented the sprite pyramid for videos and images having finite-depth scenes. In applications like distance learning, zoom-in and zoom-out are common camera operations. The original sprite is only appropriate for applications having no zooming. Traditional mosaicing techniques usually ignore these basic operations and cause blurred or very large mosaics. This problem can be resolved by mapping the frames on a pyramid where layers show different resolution. More importantly, this sprite pyramid model allows the regeneration of the video frames and objects at the resolution they were captured. If this layered representation is included in MPEG-4, it allows the regeneration of video objects at higher resolutions. There are two ways to incorporate this into MPEG-4: to consider each layer of the sprite pyramid as a separate mosaic or to introduce sprite pyramid into MPEG-4. There is a tradeoff between the number of layers and the quality of the regenerated video. More experiments need to be conducted to determine the efficient number of layers of the sprite pyramid.

5. REFERENCES

[1] T. Sikora, “The mpeg-4 video standard verification model,” *IEEE Trans. Circuits Syst. Video Technology*, vol. 7, pp. 19–31, February 1997.

[2] M. Irani and P. Anandan, “Video indexing based on mosaic representations,” in *Proceedings of IEEE*, May 1998, pp. 905–921.

[3] I. Zoghlami, O. Faugeras, and R. Deriche, “Using geometric corners to build a 2d mosaic from a set of images,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, 1997, pp. 420–425.

[4] S. Coorg and S. Teller, “Spherical mosaics with quaternions and dense correlation,” *International Journal of Computer Vision*, vol. 37, no. 3, pp. 259–273, June 2000.

[5] R. Szeliski and H-Y. Shum, “Creating full view panoramic image mosaics and environment maps,” in *Computer Graphics Proceedings, Annual Conference Series*, 1997, pp. 251–258.

[6] P. Salembier, O. Pujol, and L.Garrido, “Connected operators for sprite creation and layered representation of image sequences,” in *IV European Signal Processing Conference*, September 1998, pp. 2105–2108.

[7] Shmuel Peleg, Benny Rousso, Alex Rav-Acha, and Assaf Zomet, “Mosaicing on adaptive methods,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1144–1154, October 2000.

[8] R. S. Aygun and A. Zhang, “Reducing Blurring-Effect in High Resolution Mosaic Generation,” in *International Conference on Multimedia Expo*, Lausanne, Switzerland, August 2002.

[9] P. J. Burt and E. H. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, April 1983.

[10] A. Smolic, T. Sikora, and J.-R. Ohm, “Long-term global motion estimation and its application for sprite coding, content description and segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1227–1242, December 1999.