SUPER-RESOLUTION SPRITE WITH FOREGROUND REMOVAL

Jui-Hsin Lai, Chieh-Chi Kao, and Shao-Yi Chien

Media IC and System Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University
BL-421, 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan

ABSTRACT

Sprite is an image constructed from video clips and is also a medium for multimedia applications. An automatic sprite generation with foreground removal and super-resolution is proposed in this paper. To remove the foreground objects, each pixel-value on the sprite is iteratively updated by the value with maximum appearance probability on temporal and spatial distribution. By storing the half-pixel, super-resolution sprite has less blurring-defect from source video. In the result, the generated sprite preserves the complete scenes of background and has higher image quality, and it can used to increase the visual quality in current sprite applications and also employed to facilitate video segmentation.

Index Terms— SR sprite, SR mosaic, background scene, background video, video segmentation

1. INTRODUCTION

A sprite, which is also referred to as a mosaic, is an image constructed from multiple images or video clips. In comparison to the single image view, people can perceive the scene as a whole and have more interesting view experience from the sprite. Super-resolution(SR) is the technology to enhance the resolution of image system. The sprite generation with SR technology can be used in many multimedia applications.

For the early developments of sprite, Smolic et al. proposed the technique for long-term global motion estimation and applied the sprite on video coding [1], and Lu et al. proposed an efficient static sprite-generation and the complete compression scheme for background video coding [2]. The both works, with the help from sprite, significantly improved the efficiency of video coding. After that, Ye et al. provided a robust approach for SR static sprite generation form multiple low-resolution images [3], and Farin et al. provided the multi-sprite to reduce the coding cost and preserve the sprite details [4]. The above two works not only improved the coding efficiency but also increased the visual quality of sprite. In recent years, several interesting works based on sprite application were also proposed. Tang et al. reconstructed soccer goal events with player trajectories on a sprite, which gave people the game information under lower transmission bandwidth [5]. Gleicher and Liu used the video clips to construct an image mosaic, and the reconstructed video clips had the improvement of camera shaking [6]. Lai and Chien separated video content of tennis game by sprite plane and enriched the game video by reintegrating these video content [7].

Notice that the previous works need the background sprite to achieve above results, but how to remove foreground objects and preserve background scenes at sprite generation is still not a well-solved problem. For the foreground removal in previous works, Lu et al. [2] used the video with foreground pre-segmented to build the background sprite. However, it is hard to have the pre-segmented video in practice. Ye [3] and Lai [7] utilized the temporal filter to construct the sprite, but the generated sprite was blended with non-background pixels under foreground objects repeatedly appearing in the same region.

In this paper, an automatic generation of SR sprite with foreground removal is proposed. The SR sprite stores the half-pixel and preserves more video details that the blurring-defect from source video is also reduced. Furthermore, each pixel-value on the sprite is iteratively updated by the value with maximum appearance probability on temporal and spatial distribution. After that, the foreground objects can be efficiently removed from the sprite, and the background scenes are completely preserved. The automatic sprite generation with foreground removal can be applied to promote sprite applications. Besides, the generated sprite not only can be used to increase the efficiency of sprite coding but also provides better visual quality in previous applications. Finally, the background video, which can be employed to facilitate the video segmentation, is reconstructed from the SR sprite. With more details preserved in SR sprite, the background video has higher quality than the original video.

2. OVERVIEW OF THE PROCESSING FLOW

Fig. 1 is the processing flow of SR sprite and background video. The first step is the global motion estimation(GME) of video frames, and each pixel-value on video frames is stored

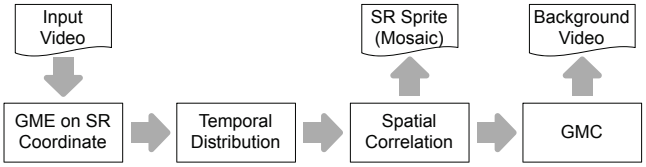

Fig. 1. The generation flow of background video and SR sprite with foreground removal.

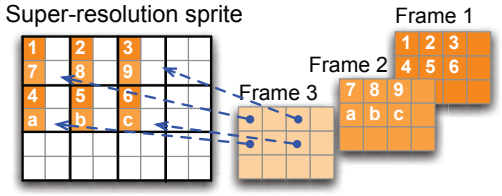

Fig. 2. The illustration of SR sprite which contains the half-pixels from video frames.

at the projected coordinate on the SR sprite. The SR sprite contains the half-pixel which is used to preserve more video details and reduce blurring-defect from input video. Without preserving the foreground pixels on the sprite, we assume that the peak distribution of pixel-values in a long-time video is the background scene. Therefore, the frame pixels with maximum probability on temporal distribution are considered as the background scene and stored in the sprite. To reduce the noise-defect in background scene, the pixel-value on the sprite is iteratively updated by the value with maximum probability of spatial co-appearance. Finally, the background video is reconstructed from the SR sprite by global motion compensation(GMC), which is the inverse processing of GME. The details of sprite generation are described in Section 3.

3. THE GENERATION OF SPRITE

3.1. GME and SR sprite

For global motion estimation(GME) of video frames, the projective model is the well-known perspective motion model in Equation 1 and is employed in this work.

$$\begin{bmatrix} x_j \\ y_j \\ w_j \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \quad (1)$$

where $m_l, \forall l \in 1:8$, are the transformation parameters from coordinate (x_i, y_i) of video frame q to coordinate $(x_j/w_j, y_j/w_j)$ of sprite image. The projective model can describe the general cases of 3-D motions like the motion of a planner scene under arbitrary rigid 3-D motions. To extract the transformation parameters, it can have the assumption that the brightness I of the video frames does not change over time. The transformation parameters are modified to minimize the difference between video frame and the sprite. The cost function $\mathbf{E}$ of difference minimization is defined as

$$E = \sum_{i \in N} |I_v(x_i, y_i) - I_s(x_j/w_j, y_j/w_j)|^2, \quad (2)$$

where $I_v(x_i, y_i)$ is the luminance value of pixel (x_i, y_i) on the video frame, $I_s(x_j/w_j, y_j/w_j)$ is the luminance value of the corresponding position $(x_j/w_j, y_j/w_j)$ on the sprite, and N is a set of effective pixels. The transformation parameters are retrieved by iteratively minimizing the cost function $\mathbf{E}$ [7].

Each pixel on video frame is projected to sprite image by these transformation parameters. However, the pixel location on the video frame may not correspond to the integer-valued pixel location on the sprite image. Approaches like bilinear interpolation are used to estimate the pixel-value at the location. Nevertheless, these approaches also blur the sprite image and decrease the sprite quality. To reduce the blurring-defect, the SR technology, which saves the half-pixel data on the sprite, is employed to directly preserve the pixel-values from video frames. As the illustration in Fig. 2, pixel-values projected to non-integer-valued position can be stored at the half-pixel locations on the SR sprite.

3.2. Choose pixel-value on temporal distribution

In a long-time video, each pixel location on the sprite would map to several pixel-values in video frames. These pixel-values belong to foreground objects or background scenes. How to choose the background pixel-value from the mapping data and build the complete background sprite is a hard problem need to solve.

For each pixel location, the pixel-value with maximum appearance probability in a long-time video is usually the background scene, because the foreground objects often have rapid movement. The pixel-value distribution, described in Equation 3, on temporal domain can be the clue to choose the background pixel.

$$h_{x_i,y_i}(k) = \frac{\sum_{t=t_1}^{t_2} \delta(I_t(x_i, y_i) = k)}{\sum_{t=t_1}^{t_2} \delta(I_t(x_i, y_i))}, \forall k \in \mathbf{C}. \quad (3)$$

where δ is the impulse function and $h_{x_i,y_i}(k)$ is the appearance probability of pixel-value k under a period time $[t_1, t_2]$ at the coordinate (x_i, y_i) on the sprite, and $\mathbf{C}$ is the RGB color space. The peak index of $h_{x_i,y_i}(k)$ is recognized as the background pixel, and the equation is described in the following.

$$E^1_{x_i,y_i} = \arg\max_k h_{x_i,y_i}(k), \quad (4)$$

where $E^1_{x_i,y_i}$ is the pixel-value with maximum appearance probability at the coordinate (x_i, y_i), and the index 1 means that $E^1_{x_i,y_i}$ is the initial pixel-value of the generated sprite.

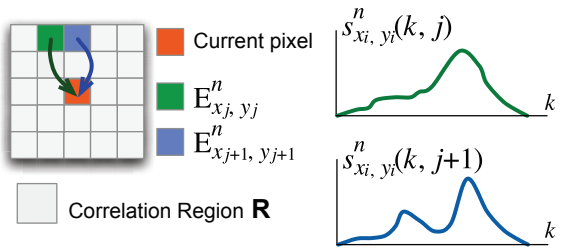

Fig. 3. The current pixel is updated by the value k with maximum co-appearance probabilities.

3.3. Update pixel-value in spatial correlation

The assumption, the pixel-value with maximum appearance probability on temporal distribution is the background pixel, is true when foreground objects have rapidly movement. Nevertheless, the foreground objects and background scenes may have the equivalent appearance probabilities, if the foreground objects occupied the fixed region for a period of time. Under such situation, the non-background pixels would be chosen as the sprite pixels and make the sprite look like blended with noise.

To solve this problem, only the pixel information of temporal distribution is not enough. The pixel correlation of spatial co-appearance also needs to be considered for choosing the background pixels. The pixel correlation of the spatial co-appearance is described in Fig. 3. Each chosen pixel on the sprite has the individual co-appearance probabilities corresponding to the surrounding pixels. The current pixel-value, $E^n_{x_i,y_i}$, is updated by the co-appearance probabilities of surrounding pixels in correlation region $\mathbf{R}$. The probabilitiy of co-appearance can be written in mathematics form in Equation 5.

$$s^n_{x_i,y_i}(k,j) = \frac{\sum_{t=t_1}^{t_2} \delta(I_t(x_i,y_i)=k | I_t(x_j,y_j) = E^n_{x_j,y_j})}{\sum_{t=t_1}^{t_2} \delta(I_t(x_i,y_i) | I_t(x_j,y_j) = E^n_{x_j,y_j})}, \tag{5}$$

where $s^n_{x_i,y_i}(k,j)$ is the co-appearance probability of pixel-value k at coordinate (x_i, y_i) under the pixel-value $E^n_{x_j,y_j}$ at coordinate (x_j, y_j) in a period of time $[t_1, t_2]$, and the index n is iteration number of pixel update. Notice that each pixel in the region $\mathbf{R}$ has the effect on the update of current pixel. Then, the current pixel-value is updated by the value k with maximum summation probabilities. The updated pixel-value $E^{n+1}_{x_i,y_i}$ is written in Equation 6.

$$E^{n+1}_{x_i,y_i} = \arg\max_k \sum_{\forall j \in \mathbf{R}} s^n_{x_i,y_i}(k,j). \tag{6}$$

Notice that each pixel-value on the sprite is iteratively updated by Equation 5 and Equation 6. The update process is repeated, until all the pixel-values on sprite are converged or the number of iteration is larger than the threshold.

4. EXPERIMENTAL RESULTS

Different tennis videos with resolution 720x480 are used as the test sequences. The tennis video is suitable to evaluate the proposed method, because there are foreground objects and background audiences in random movement. Only the results of background sprite are pasted in the paper, and the background videos are available on the website [8].

Fig. 4(a) is the input video frame with moving foreground player. Fig. 4(b) is the initial sprite composed of pixel-values with maximum appearance probability on temporal distribution. It can see that the moving foreground can be completely removed from the sprite. Then, each pixel-value on the sprite is further updated by the spatial co-appearance with 5x5 correlation region, and the result is in Fig. 4(c). For the result of SR sprite, Fig. 4(d) has less blurring-defect and the sprite quality is better than previous figures.

Unlike the moving players, the background audiences occupy the fixed region and have gesture change all the time. The pixel-value with temporal peak distribution can not correctly present the background scene in Fig. 4(e). These inconsistent pixels belong to different objects and make the sprite look like blended with noise. After sprite update in spatial correlation, the inconsistent pixels are iteratively deleted and the sprite quality is improved in Fig. 4(f). We can see that the update procedure is effective to remove the defect from temporal filter. For the quality improvement from super resolution, the more video details are preserved in Fig. 4(h) in comparison to video frame Fig. 4(g). The blurring-defect in SR sprite is decreased and more details are observed. The above results can promote the background building and provide better visual quality in sprite coding [2][4] and sprite applications [5][6][7].

Finally, the background video is reconstructed from the SR sprite by GMC. With more details preserved in SR sprite, the scene blur induced by camera motion in input video is removed in the background video. Fig. 4(i) is the input video, and Fig. 4(j) is the corresponding background video. The black region in left-down background video is the score-box region in video frame, and the score-box region is manually disable at sprite generation. In addition, the foreground segmentation is also achieved by the difference between input video and background video.

5. CONCLUSION

The proposed pixel-value updated with maximum appearance probability on temporal and spatial distribution can effectively remove foreground objects and preserve complete background scenes in the sprite. Combing the SR technology, the sprite preserves more video details and provides higher image quality. The results promote the background building and provide better visual quality in sprite coding and current sprite applications. Furthermore, the background video can

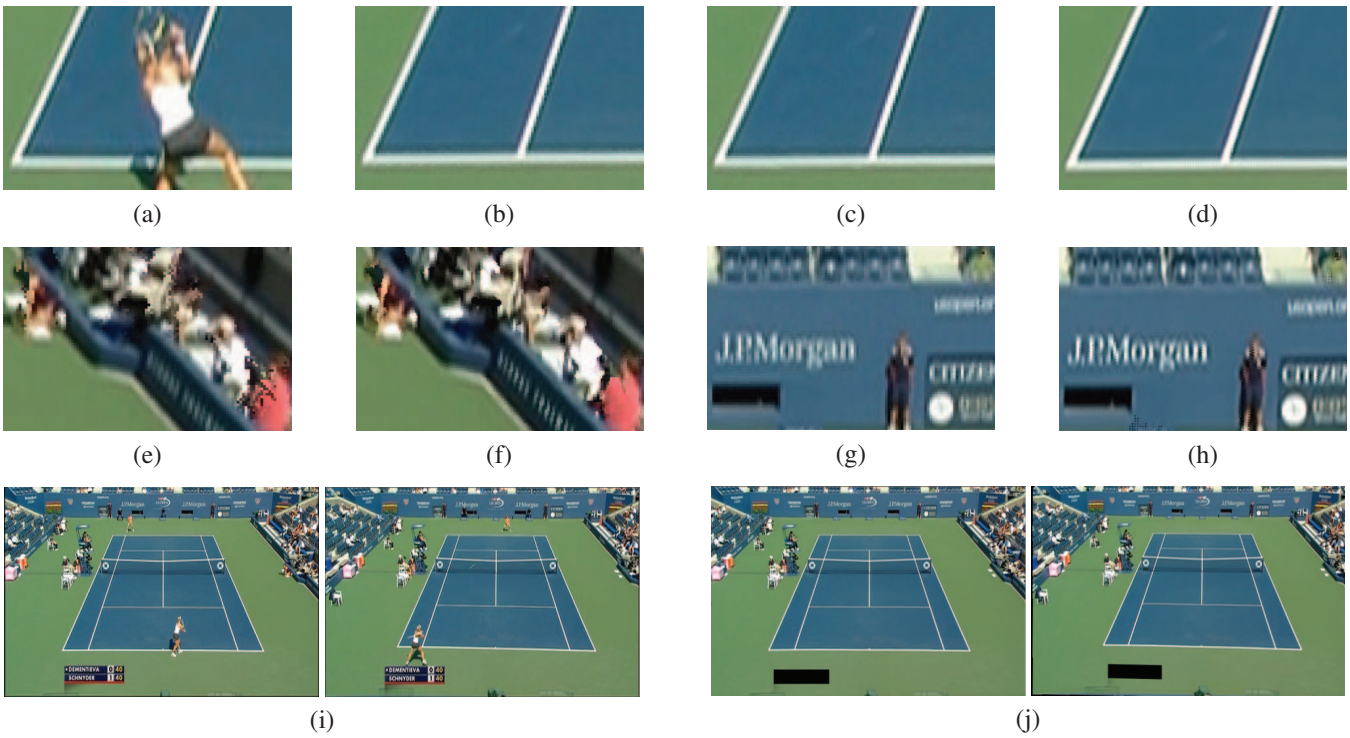

Fig. 4. (a)The moving foreground in the video frame. (b)The pixel-values on temporal peak distribution. (c)The pixel-values in Fig .4(b) updated by spatial co-appearance. (d)The result of SR sprite. (e)The sprite defect from temporal peak distribution. (f)The defect is removed by spatial co-appearance. (g)The background of video frame. (h)The SR sprite with half-pixels. (i)The input video. (j)The background video reconstructed from SR sprite.

be used to facilitate the foreground segmentation. However, some condition would fail in the proposed method like the foreground has large occupation area on the fixed region for a long time.

6. REFERENCES

[1] 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 Circuit and Systems for Video Technology*, vol. 9, no. 8, pp. 1227–1242, 1999.

[2] Y. Lu, W. Gao, and F. We, "Efficient background video coding with static sprite generation and arbitrary-shape spatial prediction techniques," *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 13, no. 5, pp. 394–405, 2003.

[3] G. Ye, M. Pickering, M. Frater, and J. Arnold, "A robust approach to super-resolution sprite generation," in *Proceedings of IEEE International Conference on Image Processing, ICIP '05.*, vol. 1, Sept. 2005, pp. 897–900.

[4] D. Farin and P. H. de With, "Enabling arbitrary rotational camera-motion using multi-sprites with minimum coding-cost," *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 16, no. 4, pp. 492–506, 2006.

[5] Q. Tang, I. Koprinska, and J. S. Jin, "Content-adaptive transmission of reconstructed soccer goal events over low bandwidth networks," in *Proceedings of the 13th Annual ACM International Conference on Multimedia, MULTIMEDIA '03*, Nov. 2005, pp. 271–274.

[6] M. L. Gleicher and F. Liu, "Re-cinematography: Improving the camera dynamics of casual video," in *Proceedings of the 15th Annual ACM International Conference on Multimedia, MULTIMEDIA '07*, Sept. 2007, pp. 27–36.

[7] J. H. Lai and S. Y. Chien, "Tennis video enrichment with content layer separation and real-time rendering in sprite plane," in *Proceedings of IEEE 10th Workshops on Multimedia Signal Processing, MMSP 2008*, Oct. 2008, pp. 672–675.

[8] [Online]. Available: http://media.ee.ntu.edu.tw/larry/icme09/