MULTIPLE BACKGROUND SPRITE GENERATION USING CAMERA MOTION CHARACTERIZATION FOR OBJECT-BASED VIDEO CODING

Andreas Krutz, Alexander Glantz, Martin Haller, Michael Droese, and Thomas Sikora

Communication Systems Group
Technische Universität Berlin
Berlin, Germany

ABSTRACT

Recent work has shown that object-based video coding can provide higher coding gain than common H.264/AVC for single-view and the MVC standard based on H.264 for multi-view (MVC). The use of background sprites outperformes the AVC/MVC especially in sequences containing rotating camera motion and moving foreground objects. The coding performance strongly relies on the preprocessing steps, e.g. sprite generation and object segmentation. In this paper, we present an enhanced background sprite generation algorithm for object-based single- and multi-view video coding (OBVC/OBMVC). It is a new feature for our OBVC/OBMVC recently proposed. We produce multiple background sprites based on camera motion characterization and physical camera parameter estimation. Experimental results show how these multiple sprites increase the coding performance for single- and multi-view sequences.

Index Terms— sprite generation, camera motion analysis, multi-view video coding, H.264/AVC

1. INTRODUCTION

The development of advanced video compression techniques has resulted in the H.264/AVC video coding standard. The excellent performance of this codec has been shown in several studies, e.g. [1]. The codec is based on the classical hybrid video coding approach as known from earlier video coding standards like MPEG-1,2,4. During recent years, it has been shown that object-based video coding (OBVC) has several advantages in comparison to the classical hybrid video coding. Object-based coding schemes using background sprites have been introduced in [2], [3], and [4]. The idea is to segment the video content first in foreground and background objects. The background object is also known as a background sprite. These objects are then coded and transmitted independently. The decoder combines the objects controlled by higher-order motion parameters. There are several important pre-processing steps where object-based coding using background sprites relies on. One is the foreground/background segmentation. This can be accomplished independently from the coding approach. However, segmenting the video content using the already generated sprite is a less complex task for the encoder. The coding gain increases especially if the background object is much larger than the foreground objects. Another important step, even for the segmentation, is the background sprite generation. It has been shown in [5] that multiple sprites achieve improved results in object segmentation. In our object-based approach recently published, we also apply multiple sprites to increase the coding performance instead of using a single background sprite containing all the background information of the considered sequence [6].

We propose an extended background sprite generation method. In the first step, the video sequence is characterized regarding to its camera motion. Having the video sequence classified in several camera motion types, it is segmented into sub-sequences for sprite generation. The second step analyzes the sub-sequence whether a single sprite or a multiple sprite based on the physical camera motion is produced. The characterization of the camera motion is obtained based on a system recently proposed [7]. Physical camera parameter are estimated then for each sub-sequence to decide between generating a single or a multiple sprite [8].

The remainder of this paper is organized as follows. The enhanced background sprite generation method is described in Section 2 including all details concerning the whole processing chain. Section 3 outlines coding issues regarding to the background sprites. Experimental results are presented in Section 4. The last section summarizes the paper and gives an outlook to possible future work.

2. CAMERA MOTION CHARACTERIZATION FOR MULTIPLE BACKGROUND SPRITE GENERATION

2.1. Global Motion Estimation

The performance of the whole sprite coder critically depends on the estimation of the background object motion. Therefore, it is very important to apply an image registration technique with very accurate estimation of the higher-order motion parameters contained in the homography $\mathbf{H}$. For this, a gradient-based approach is applied using additional techniques, such as phase correlation based initialization and techniques which set up the use of the motion parameters for segmentation. A detailed description of the utilized image registration algorithm can be found in [9].

2.2. Camera Motion Characterization (CMC)

Background sprites are synthesized from image sequences with camera panning and tilting whereas sequences without these camera movements do not contribute to the generation of background sprites. An object-based video encoder that uses multiple background sprites can use metric-based measures such as the rotation angles for panning/tilting or alternatively the segmentation results of camera motion characterization. Since panning/tilting are the motion types of interest here, the characterization of camera work considers only panning left/right, tilting up/down, and no panning/tilting.

This work was developed within 3DTV (FP6-PLT-511568-3DTV), a European Network of Excellence funded under the European Commission IST FP6 programme.

The camera motion characterization approach as shown in Figure 1 has a feature extraction, classification, and a temporal segmentation stage. In the following each of these stages are described briefly. A more detailed description of this approach can be found in [7].

The feature extraction uses the horizontal and vertical translational parameters $h_{x,l}$ and $h_{y,l}$ of the earlier estimated perspective global motion parameters contained in $\mathbf{H}$ as input to compute four features for pan and tilt, respectively. The index l addresses all motion parameters for frames l and $(l+1)$. The complex normalized value

$$\underline{t}_l = \frac{h_{x,l}}{w} + j\frac{h_{y,l}}{h} \quad (1)$$

is used to determine the median angle $\phi_{\text{t,med},l}$ of translational motion and the medians $t_{\text{x,med},l}$ and $t_{\text{y,med},l}$ with

$$\phi_{\text{t,med},l} = \underset{s_l \le k \le e_l}{\text{median}} \left(\arg\left(\underline{t}_k\right)\right) \quad (2)$$

$$t_{\text{x,med},l} = \underset{s_l \le k \le e_l}{\text{median}} \left(h_{x,k}\right) \quad (3)$$

$$t_{\text{y,med},l} = \underset{s_l \le k \le e_l}{\text{median}} \left(h_{y,k}\right), \quad (4)$$

where w and h are the image width and height and s_l and e_l are given as

$$s_l = l - W_{\text{med}} + 1 \quad ; \quad e_l = l + W_{\text{med}}.$$

The median filtered parameters are robust against possible GME outliers. The used windowed median filter has a length of $W_{\text{med}} = \lfloor R_f/2 \rfloor$ with R_f as frame rate per second of the video sequence.

Short-time translational angle histograms based on $\phi_{\text{t,med},l}$ are determined to obtain more robust features for the direction of translational motion. The used angle quantization scheme is shown in Fig. 2. The derived rates $R_{\text{TAHPL},l}$, $R_{\text{TAHPR},l}$, $R_{\text{TAHTU},l}$, and $R_{\text{TAHTD},l}$ represent the occurrence of angles for pan left/right and tilt up/down in the respective range of angles normalized to the window length W for the histogram computation. The used overlap of windows is extensive for a proper temporal resolution.

The zero-crossing rates for horizontal and vertical translational motion parameters are defined by

$$Z_{\text{x},l} = \frac{1}{2W} \sum_{i=s_l}^{e_l} \left|\text{sgn}\left(h_{x,i}\right) - \text{sgn}\left(h_{x,i-1}\right)\right| \quad (5)$$

$$Z_{\text{y},l} = \frac{1}{2W} \sum_{i=s_l}^{e_l} \left|\text{sgn}\left(h_{y,i}\right) - \text{sgn}\left(h_{y,i-1}\right)\right| \quad (6)$$

and capture the reliability of intended translational motion within the window of the length W .

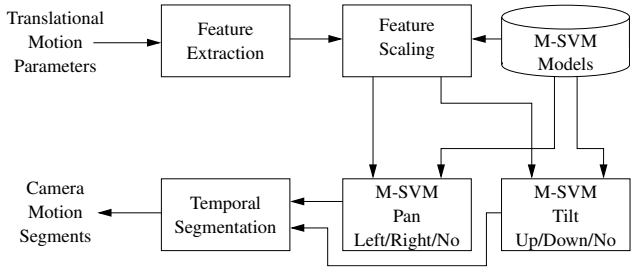


Fig. 1. Camera motion characterization for pan left, pan right, no pan, tilt up, tilt down, and no tilt

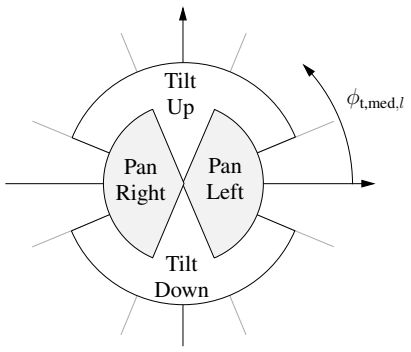


Fig. 2. Quantization scheme for the translational motion angle histogram (TAH)

The complete four-dimensional feature vectors for classification of pan and tilt are as follows

$$\mathbf{x}_{\text{pan},l} = \left(\begin{array}{cccc} t_{\text{x,med},l} & R_{\text{TAHPL},l} & R_{\text{TAHPR},l} & Z_{\text{x},l} \end{array} \right)^{\text{T}} \quad (7)$$

$$\mathbf{x}_{\text{tilt},l} = \left(\begin{array}{cccc} t_{\text{y,med},l} & R_{\text{TAHTU},l} & R_{\text{TAHTD},l} & Z_{\text{y},l} \end{array} \right)^{\text{T}}. \quad (8)$$

Multi-class support vector machines (M-SVMs) are used to classify the camera motion types. The M-SVMs provide for each image pair a result with the three possible states pan left/right, and no pan as well as tilt up/down, and no tilt. The models for the M-SVMs were trained on features extracted from selected videos of the TRECVid 2005 BBC rushes video corpus [7].

The temporal segmentation starts with a median filtering of results over 15 frames. This improves the temporal stability. Changes between camera motion types are then identified within an image sequence as boundaries of segments with the same type of camera motion. This leads to a camera motion-based temporal segmentation.

2.3. Physical Camera Parameter Estimation

Concatenating the short-term perspective camera parameters (homographies $\mathbf{H}_{n-1,n}$) in a recursive way yields non-exact long-term parameters representing the transformation $\mathbf{H}_{0,n}$ between any frame and the reference frame. These homographies are the base for a robust but coarse camera calibration technique, published in [8]. Here we exploit the fact that for common camera setups the homographies can be decomposed in a product of intrinsic and extrinsic camera parameter matrices

$$\mathbf{H}_{0,n} = \mathbf{F}_n \mathbf{R}_{0,n} \mathbf{F}_0^{-1} \quad (9)$$

$$= \frac{1}{\alpha_{0,n}} \begin{pmatrix} r_{00} & r_{01} & f_0 r_{02} \\ r_{10} & r_{11} & f_0 r_{12} \\ r_{20}\alpha_{0,n}/f_0 & r_{21}\alpha_{0,n}/f_0 & r_{22}\alpha_{0,n} \end{pmatrix},$$

where $\mathbf{R}_{0,n}$ is the rotation matrix between frame 0 and n and $\mathbf{F}_n$ and $\mathbf{F}_0$ contain focal length values of both views. After computing the focal length ratio $\alpha_{0,n} = f_0/f_n$ we calculate the focal length of the reference frame as median of all solutions resulting from Equ. 9. This is done by exploiting orthogonality and constant vector norm constraints for the matrices $\mathbf{H}_{0,n}$. Knowing all focal lengths the rotation angles can finally be computed using trigonometrically properties of the center points of every image [8].

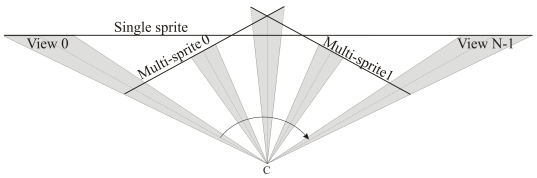


Fig. 3. Partition of a sequence into multi-sprites for panning camera with constant focal length

2.4. Multiple Sprite Generation using CMC

The characterization of the camera motion separates the video sequence into segments depending on the camera motion. Afterwards, for each segment, the physical camera parameters are estimated and based on that it is decided whether a single or a multiple sprite is generated for the current segment. Figure 4 illustrates this method.

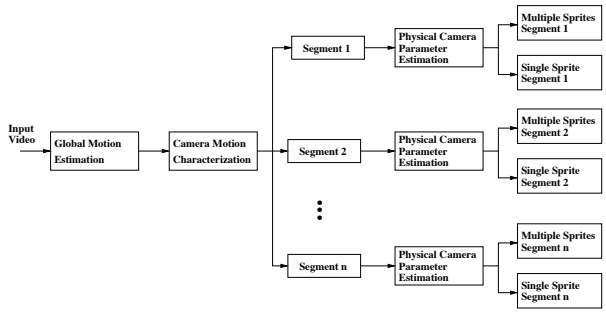


Fig. 4. Multiple Sprite Generation using CMC

The new method is applied on two test sequences, the well-known "Stefan" - sequence for single view and four views of the MPEG multi-view sequence "Race1". For the "Stefan" - sequence, six background sprites are generated. Firstly, the camera motion characterization separates the video sequences in four parts. Then, for each part, physical camera parameters are calculated. These parameters are used to segment the fourth sub-sequence into three sprites again. This leads to the six part sprites as shown in Fig. 5. The first four views of the "Race1"-sequence are only segmented into two parts using this approach (see Fig. 6).

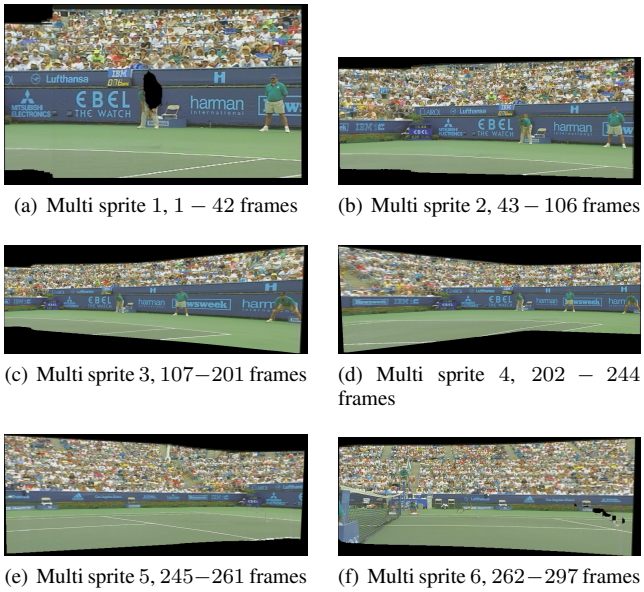
3. CODING THE MULTIPLE BACKGROUND SPRITES FOR SINGLE- AND MULTI-VIEW

For object-based video coding, the background sprites are coded as single images for the single-view case. The sprite images are considered as sequences with one frame and the H.264/AVC is used for coding. For the multi-view case, there are sprite images for each view. In other words, they build a multi-view sequence where every view only have one frame. To code this sprite sequence, the latest standardized MPEG-MVC is used. The coding structure for this small sprite sequence is $IPPP$. Figure 7 depicts the coding structure of the background sprite sequence.

4. EXPERIMENTAL RESULTS

We evaluate the new sprite generation technique in two ways. Firstly, we compare the reconstructed background frames of each sprite generation method, that is single sprite (Algo.1), multiple sprites based on physical camera estimation (Algo.2), and the proposed multiple sprites using camera motion characterization (Algo.3). Table 1 shows mean background PSNR-values of the two test sequences. For the "Stefan"-sequence, we can compare all three considered sprite generation methods. Our new method outperforms the other two up to 1.25 dB. Outstanding results are achieved in the multi-view case. Here, the multiple sprites based on physical camera parameter estimation are not reliable, because of the smaller camera pan. So the single sprite and the multiple sprite based on camera motion estimation are compared for the first four views of the "Race1"-sequence. The mean PSNR over all frames of the first four views increases to 3.7 dB.

Secondly, the coding performance was examined of the several sprite generation methods. We coded the different background sprites and produced rate-distortion curves for the single- and the multi-view case. Figure 8 shows the curve for the "Stefan"-sequence. It can be seen that we obtained up to 1.2 dB improvement in higher bit rate ranges. In Fig. 9 the curve is drawn for the multi-view case. As expected from the direct PSNR-value comparison we achieve much higher coding performance using the new multiple sprite generation technique based on CMC. Overall we can state that our new algorithm improves recent techniques in direct PSNR-evaluation as well as in coding behavior.



(a) Multi sprite 1, $1-42$ frames (b) Multi sprite 2, $43-106$ frames

(c) Multi sprite 3, $107-201$ frames (d) Multi sprite 4, $202-244$ frames

(e) Multi sprite 5, $245-261$ frames (f) Multi sprite 6, $262-297$ frames

Fig. 5. Multiple Background sprites (Algo.3) over all 297 frames, test sequence "Stefan"

(a) Multi sprite 1, $1-87$ frames (b) Multi sprite 2, $88-530$ frames

Fig. 6. Multiple Background sprites (algo.2) over all 532 frames, sequ. "Race1", "view 0"

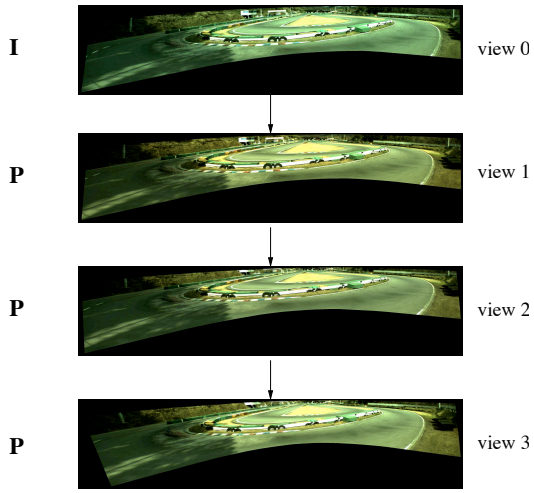


Fig. 7. Coding structure of background sprite sequence, "Race1"

Table 1. Mean Background-PSNR values of single-view and multi-view sequences

Sequence	**Method**	PSNR (mean) in dB
"Stefan"	Algo.1	26.50
"Stefan"	Algo.2	26.37
"Stefan"	Algo.3	27.58
"Race1, 4 views mean"	Algo.1	24.38
"Race1, 4 views mean"	Algo.3	28.09

5. CONCLUSION

We have presented a new background sprite generation technique based on camera motion characterization and physical camera parameters. We have shown that our new approach improved recent techniques especially for the considered multi-view test sequence. The next step is to apply this method in our whole object-based video coding system and we expect highly improvement applying this new technique. Additionally, further techniques can be used in the generation process, e.g. super-resolution sprite generation.

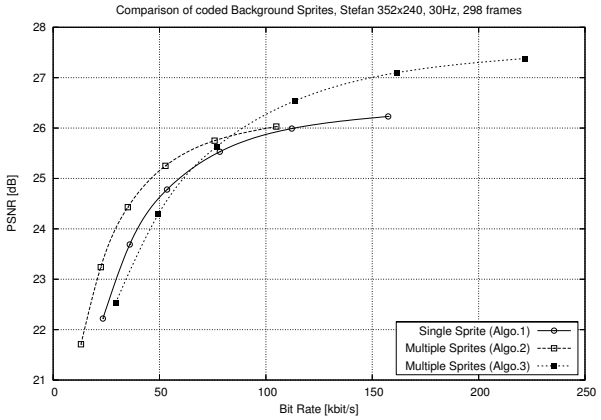


Fig. 8. Rate-distortion curve of the considered sprite generation methods, "Stefan", 298 frames

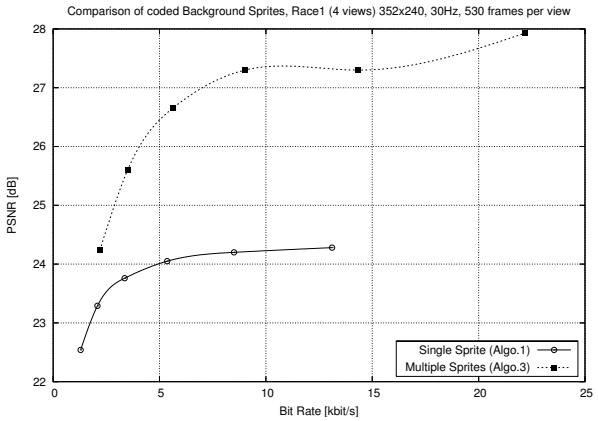


Fig. 9. Rate-distortion curve of the considered sprite generation methods, "Race1", 4 views, 530 frames per view

References

[1] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 302–311, July 2003.

[2] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 8, pp. 19–31, February 1997.

[3] A. Smolic, T. Sikora, and J.-R. Ohm, "Long-term global motion estimation and its application for sprite coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1227–1242, December 1998.

[4] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, pp. 6–17, January 2005.

[5] D. Farin, P. H. N. de With, and W. Effelsberg, "Video object segmentation using multi-sprite background subtraction," in *Int. Conf. on Multimedia and Expo (ICME)*, Taipei, Taiwan, June 2004.

[6] M. Kunter, A. Krutz, M. Droese, M. Frater, and T. Sikora, "Object-based multiple sprite coding of unsegmented videos using h.264/avc," in *IEEE International Conference on Image Processing (ICIP2007)*, San Antonio, USA, Sept. 2007.

[7] M. Haller, A. Krutz, and T. Sikora, "A generic approach for motion-based video parsing," in *Proc. EUSIPCO*, 2007, pp. 713–717.

[8] M. Kunter, A. Krutz, M. Mandal, and T. Sikora, "Optimal multiple sprite generation based on physical camera parameter estimation," in *Visual Communications and Image Processing (VCIP'07)*, San Jose, USA, Jan. 2007.

[9] A. Krutz, M. Frater, M. Kunter, and T. Sikora, "Windowed image registration for robust mosaicing of scenes with large background occlusions," in *Int. Conf. on Image Processing (ICIP06)*, Atlanta, USA, Oct. 2006.