Low Bit-Rate Object-based Multi-view Video Coding using MVC

Andreas Krutz[*]*, Michael Dröse*[*]*, Matthias Kunter*[*]*, Mrinal Mandal*[**]*,*
Michael Frater[***]*, and Thomas Sikora*[*]

[*]Commun. Systems Group
TU Berlin
Berlin, Germany

[**]Department of Electr. & Comp. Eng.
University of Alberta
Edmonton, Canada

[***]School of IT and EE
University of New South Wales
Canberra, Australia

ABSTRACT

Work is currently underway to develop a new multi-view video coding (MVC) standard, based on the recent H.264/AVC standard. Recent work has shown, however, that object-based video coding can provide higher performance than H.264, especially at low bit rates and in sequences containing rotating camera motion and moving objects in the foreground. In this paper, we propose an object-based extension to MVC, in which sequences are segmented into foreground and background objects, with each object coded using H.264/AVC for single-view sequences and MVC for multi-view sequences. Experimental results show that the object-based approach significantly outperforms the basic MVC at low bit rates.

Index Terms— Object-based video coding, sprite coding, multiview video, H.264/AVC

1. INTRODUCTION

The development of advanced video compression techniques has resulted the H.264/AVC video coding standard. The excellent performance of the codec has been shown in several studies, e.g. [1]. This codec is based on the classical hybrid video coding approach as known from earlier video coding standards like MPEG-1,2,4. It includes several new features, such as 16x16 down to 4x4 blocks, integer transform, in-loop deblocking filter, and an improved entropy coder, and generally provides a better coding efficiency. During recent years, it has also been shown that another approach, object-based video coding (OBVC), has several advantages in comparison to the classical hybrid video coding. Object-based, or sprite, coding schemes have been introduced in [2] and [3]. The approach here is to segment the video content first in foreground and background objects. The background object is also known as sprite or video mosaic. These objects are then coded and transmitted independently. The decoder combines the objects controlled by higher-order motion parameters. The first important step to maximize coding efficiency is frame-to-frame image registration, or global motion estimation. Here, the motion of the background object is estimated applying a higher-order camera motion model. Then a sprite is generated using the achieved motion parameters. There exist several approaches for handling the segmentation. The foreground/background segmentation can be accomplished independently from the coding approach. However, segmenting the video content using the 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.

In this paper the sprite based codec is extended to multi-view video sequences. Some initial works on this had been reported in [4]. Our new object-based multi-view video coding (OBMVC) scheme relies on an existing MVC scheme in [5]. In the pre-processing step the video content is separated in a foreground and a sprite sequence. The segmentation is based on an accurate image registration technique and reconstructed background frames from the sprite. Both, foreground and sprite sequences are then coded utilizing the AVC for single-view video and MVC for multi-view video. Additional techniques, including illumination compensation and deblocking of foreground objects, increase the coding effiency. Experimental results show that for a certain range of bit-rates, our codec generally provides a better performance than the existing MVC. Furthermore, we examined the best combination of coding the foreground and background sequence. The goal of our work is to provide an extended coding mode which can be chosen if the object-based approach yields better coding results than the conventional MVC.

The remainder of this paper is organized as follows. The object-based codec for single-view video is described in Section 2. Section 3 extends the single-view coding to the multi-view video case. Experimental results are presented in Section 4. Finally, the last section concludes the paper and outlines possible future work.

2. SINGLE-VIEW SPRITE CODING USING AVC

The sprite coding approach using the AVC is presented in the following. An overview of the whole codec (for single- and multiview) is shown in Fig. 1. The image registration is used first for the sprite generation. Both techniques set up the algorithm for the object segmentation. The resulted foreground object sequence and the sprite sequence are then coded using the AVC. The binary object mask, which is used for an accurate composition of the foreground and background objects at the decoder, is coded by a binary arithmetic coder. The side information contains the higher-order motion parameters and the parameters for an illumination compensation at the object-based decoder.

2.1. Short-Term Image Registration

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. For this, a gradient-based approach is applied using additional techniques, such

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

M. R. Frater was supported in this work by the Australian Research Council under project DP0667074

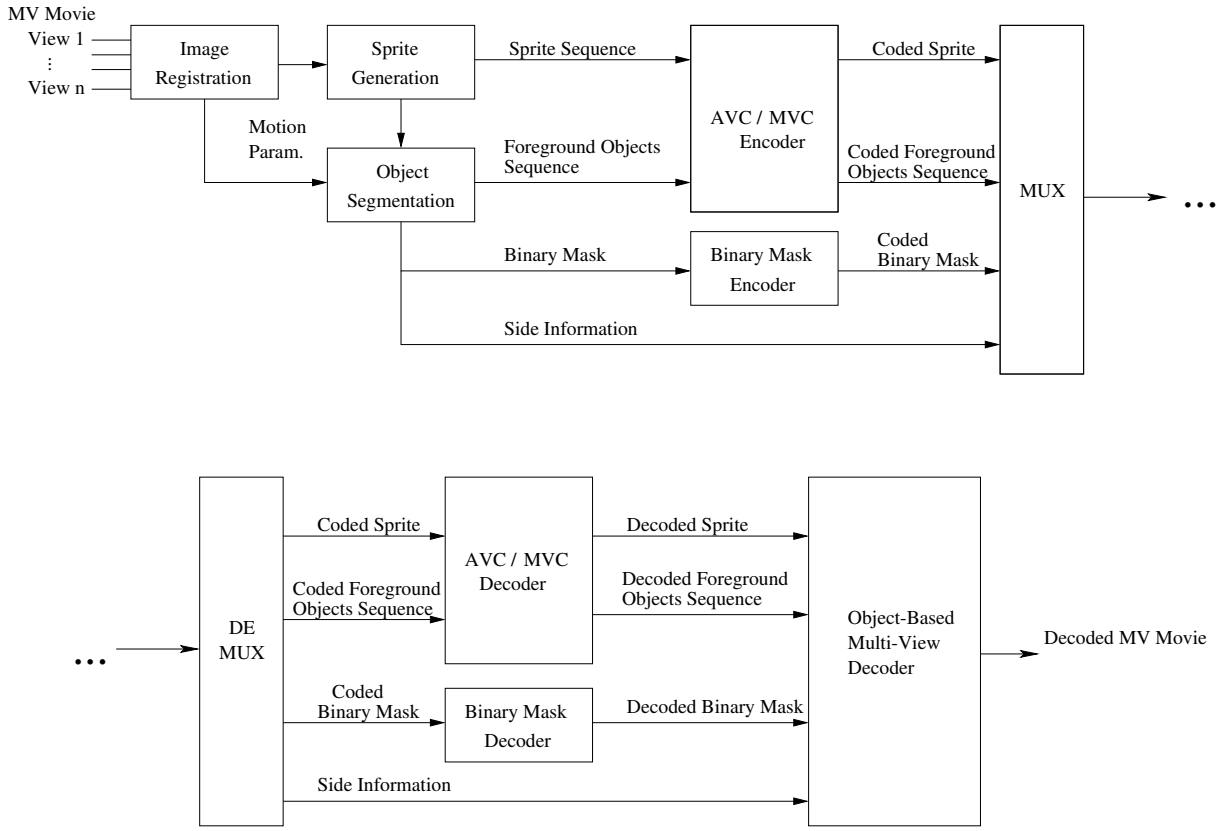


Fig. 1. Object-based Multi-View Video Codec

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 [6].

2.2. Sprite Generation and Background Reconstruction

In order to minimize the coding cost for the rigid background we construct a background sprite of the scene shot by registration of every frame into the reference frame coordinate system. The long-term transformation parameters are computed by Levenberg-Marquardt gradient descent minimization of the image differences between the compensated frames and the sprite image. As initial values for this process the concatenated short-term parameters are utilized. Knowing the exact transformations we robustly blend all pixel candidates to remove the foreground objects. A general overview of our sprite generation process is given in [7].

Reprojecting the sprite using the inverse transformation parameters yield as a background movie of the scene. Since camera dependent illumination properties get lost, e.g. in scenes with dark border regions, the sprite background will not exactly match the original frame. To compensate this difference we model the illumination characteristics by a separable parabolic function in horizontal and vertical image direction. The *illumination compensation* value is obtained by

$$\Delta Y(x,y) = a_0 + a_1 x + a_2 x^2 + b_0 + b_1 y + b_2 y^2. \qquad (1)$$

The parameters $\mathbf{a}$ and $\mathbf{b}$ are determined by modelling the averaged illumination difference along the lines through the image center. Thus, calculating these additional six parameters for a shot significantly increases the background reconstruction quality.

2.3. Motion-based Object Segmentation using Sprites

The proposed segmentation algorithm belongs to the general class of background subtraction methods. After the sprite generation, no foreground objects appear in the sprite. If the video frames are reconstructed from the sprite, only information of the background object is present. These frames can be used for a subtraction from the original frames. In the best case, only the foreground objects appear in the resultant frame. In practice, there are several problems which have to be solved. Due to small errors during long-term parameter estimation, the reconstructed frames may be distorted and as a result the segmentation becomes more difficult. To compensate this distortion additionally the short-term error frames are used in the segmentation process. Thus the foreground objects are segmented very precisely by combining both error frames and an additional post-processing step using low-pass filtering with anisotropic diffusion [8].

2.4. Binary Mask Coding

The binary mask coding scheme uses the binary arithmetic coder "M-Coder", which is specified in the H.264/AVC standard [9]. Eight contexts are initialized to model 8 possible spatial states as shown in Fig. 2. In advance to the coding, the mask is divided into 16x16 blocks.

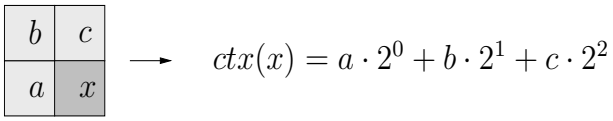


Fig. 2. Context assignment for $bit(x)$ in Binary Mask Coding

3. EXTENSION TO MULTI-VIEW SPRITE CODING USING MVC

3.1. Object-based Multi-View Video Coding (OBMVC)

A simplified schematic of the proposed MVC is shown in Fig 1. For each view, a sprite and a foreground object sequence are built. Figures 3 and 4 show the two sprites generated corresponding to view 1 and view 2 over 100 frames of the "Race1" sequence. The sprites for each view and the associated foreground objects sequences are coded using a multi-view prediction scheme shown in Fig. 5. Here, an $IBBB...$ frame structure is used with $GOP = 15$. Although, Fig. 5 shows the scheme for only two-views, it can be extended to more views easily.

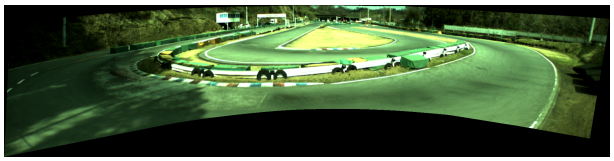


Fig. 3. Sprite sequence "Race1" view 1 100 frames

Fig. 4. Sprite sequence "Race1" view 2 100 frames

4. EXPERIMENTAL RESULTS

4.1. Single-View Object-based Coding

Our coding approach is first examined for the single-view case to show whether we can obtain some gain in coding efficiency against the "state-of-the-art" codec H.264/AVC. For this, a quality limit is calculated from uncoded reconstructed frames using the object-based decoder. The maximum achievable PSNR is 29.86dB for view 1 of the "Race1" sequence. That means that our approach can only compete with AVC in the low bit-rate range. We applied the AVC using hierarchical B-frames with temporal access every 0.5 seconds. For our object-based codec, the settings are variable. The motion parameters, the parameters for the illumination compensation, and the coded binary mask require approximately 8.85 kbit/s per view. This amount is fixed for all coding cases. We coded the sprite and the objects sequence with different QPs to develop an optimal trade-off between rate and quality. Table 1 depicts the results. For three different sprite quantization values the object sequence quality is varied

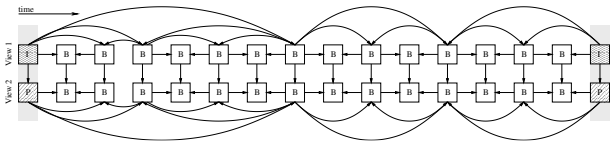


Fig. 5. Multi-View Prediction Scheme for the foreground object sequence ($GOP = 15$)

from high to low. Figure 6 shows that our codec has better performance in comparison to the AVC until a bit-rate of 124 kbit/s for the single view. The lower the bitrate the higher is the coding gain for the object-based approach. This is explained by the fact that the objects sequence allocates the biggest portion of the overall bit-rate. However, the sprite, which contains the background object, dominates the PSNR-value, i.e. the quality of the decoded frames. Therefore, we can achieve a significant coding gain in comparison to the AVC in the lower bit-rate range.

4.2. Extension to Multi-View Video Sequences

We add now the second view of the "Race1" sequence to evaluate our approach for two views. The same coding settings as for the single-view case were applied using the MVC. The prediction scheme for the two views is shown in Fig. 5. Figure 7 shows the coding results. It can be seen that our codec performs better than the MVC. The sprite sequence contains two frames of sprites which are coded predictively and yields a higher coding gain. Thus, we have a better performance in the higher quality curve in comparison to the single-view case. To evaluate the subjective improvement of the coded video content, two examples are shown below. Figure 8 (a) and (b) shows the comparison of a frame taken from the sequence "Race1" view 1 coded with the object-based coder and MVC. The frames show the difference in quality between the background objects reconstructed using two coding approaches. It can be observed that, the lower the bit-rate, the higher are the artifacts in the MVC. The background reconstructed from the sprites is almost free from these errors. This result illustrates one of the key advantages of the sprite coding approach. Furthermore, Fig. 8 (c)-(e) shows a zoom-in of an example frame which shows the foreground object. It can be seen that even the object sequence is reconstructed with a better quality than using the MVC. Thus, the object based approach significantly outperforms common hybrid codecs for the considered test sequences in both object and subjective performance assessment.

5. CONCLUSIONS

In this paper, we have presented a new object-based coding approach for multi-view video sequences. This approach segments multi-view video sequences into foreground and background objects, coding the background object as a sprite. For low bit-rates the object-based codec yields better performance than the conventional approach, due to the sprite coding in comparison to common block-based hybrid codecs. The quality of the background sprite, however, places an upper bound on the quality that can be achieved for the background. This leads to a truncation of the usable bit-rate range. Increasing the quality of the sprite would overcome this limitation, which is the subject of further work. A second important task is to find an optimal rate-distortion trade-off between coding the sprite and the foreground objects.

PSNR in dB	Sprite Rate	Objects Rate	Total Rate
28.47	49.27	76.54	134.66
28.28	49.27	66.02	124.14
28.03	49.27	57.69	115.81
28.03	49.27	51.04	109.16
27.84	32.22	76.54	117.61
27.68	32.22	66.02	107.09
27.45	32.22	57.69	98.76
27.23	32.22	51.04	92.11
26.94	21.44	76.54	106.83
26.81	21.44	66.02	96.31
26.62	21.44	57.69	85.53
26.43	21.44	51.04	81.33

Table 1. Partioning of the Bit Rate in kbit/s

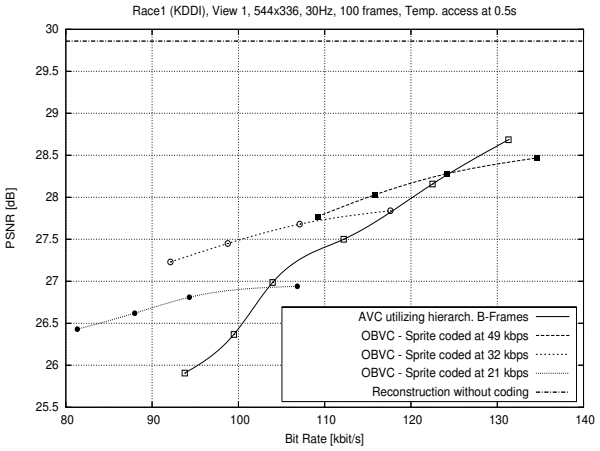


Fig. 6. Coding results for single-view video

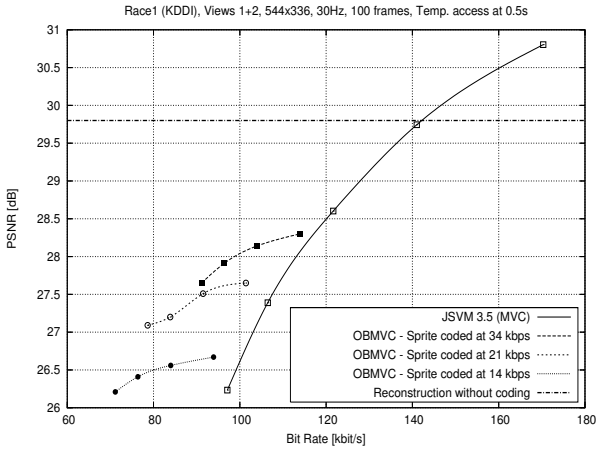
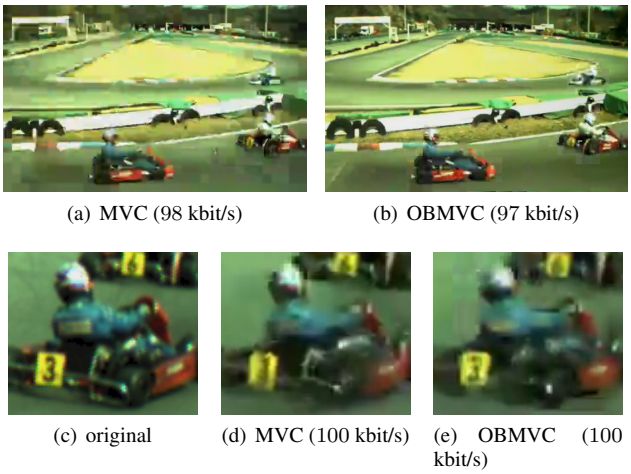


Fig. 7. Coding results for multi-view video (average per view)



(a) MVC (98 kbit/s) (b) OBMVC (97 kbit/s)

(c) original (d) MVC (100 kbit/s) (e) OBMVC (100 kbit/s)

Fig. 8. Comparison of decoded frames (frame 53) and cutouts (frame 53 of sequ."Race1 view 1"

6. 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, pp. 19–31, 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] N. Grammalidis, D. Beletsiotis, and M.G. Strintzis, "Sprite generation and coding in multiview image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 302–311, 2000.

[5] P. Merkle, K. Mueller, A. Smolic, and T. Wiegand, "Efficient compression of multi-view video exploiting inter-view dependencies based on H.264/MPEG4-AVC," in *Int. Conf. on Multimedia and Expo (ICME'06)*, Toronto, Canada, 2006.

[6] 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.

[7] M. Kunter and T. Sikora, "Super-resolution mosaicing for object based video format conversion," in *7th Works. on Im. Analysis f. Multimedia Interac. Serv. (WIAMIS'06)*, Seoul, Korea, 2006.

[8] A. Krutz, M. Kunter, M. Mandal, M. Frater, and T. Sikora, "Motion-based object segmentation using sprites and anisotropic diffusion," in *8th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Santorini, Greece, June 2007.

[9] H. Schwarz D. Marpe and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, July 2003.