28th Picture Coding Symposium, PCS2010, December 8-10, 2010, Nagoya, Japan

RECENT ADVANCES IN VIDEO CODING USING STATIC BACKGROUND MODELS

Andreas Krutz, Alexander Glantz, and Thomas Sikora

Communication Systems Group
Technische Universität Berlin
Berlin, Germany

ABSTRACT

Sprite coding, as standardized in MPEG-4 Visual, can result in superior performance compared to common hybrid video codecs both objectively and subjectively. However, state-of-the-art video coding standard H.264/AVC clearly outperforms MPEG-4 Visual sprite coding in broad bit rate ranges. Based on the sprite coding idea, this paper proposes a video coding technique that merges the advantages of H.264/AVC and sprite coding. For that, sophisticated algorithms for global motion estimation, sprite generation and object segmentation – all needed for thorough sprite coding – are incorporated into an H.264/AVC coding environment. The proposed approach outperforms H.264/AVC especially in lower bit rate ranges. Savings up to 21% can be achieved.

Index Terms— H.264/AVC, global motion estimation, background modeling, object segmentation, sprite coding

1. INTRODUCTION

Sprite coding has been developed and standardized some years ago in MPEG-4 Visual (Part 2) [1]. The main idea of the sprite coding approach is to segment the video content into foreground and background objects in a preprocessing step. For the background object, a model, i.e. a so-called background sprite image, is generated, which contains all background pixels of a given part of the sequence. Global motion estimation (GME) techniques initiate the sprite generation process and motion parameters are transmitted as side information to the receiver. The background sprite image and the foreground object sequence are coded separately. At the receiver, a background sequence is reconstructed using the background sprite image and the global motion parameters. The background sequence is merged with the transmitted foreground object sequence to build a representation of the original sequence. This approach is very promising, as has been outlined in [2].

Classical problems in sprite coding involved the performance of its tools used for preprocessing, i.e. GME, sprite generation, and object segmentation. A long time, these were of inferior quality. However, a lot of research has been done and today sprite coding seems more than feasible. In [3] and [4], the respective authors have shown extremely sophisticated pixel-based global motion estimation algorithms, Farin [5] and Kunter [6] brought forward the area of sprite generation, and finally the authors of this paper have previously proposed a very good object segmentation approach [7]. Ideas very similar to the sprite techniques have been employed in other model-based video coding approaches as well [8], [9].

Since sprite coding performed so well compared to classical hybrid video coding, the idea is to adapt the sprite coding techniques to the state-of-the-art video coding standard H.264/AVC [10]. For that, the proposed video coding scheme first estimates short-term global motion parameters as is done using MPEG-4 Visual. These parameters can then be used both for background sprite generation and foreground object segmentation. However, to fit the H.264/AVC macroblock structure – MPEG-4 Visual allows the definition of arbitrary sized video objects – the generated foreground object sequence and binary mask have to be adjusted. Finally, an encoder control chooses optimal quantization step sizes for the foreground object sequence and background sprite.

This paper is organized as follows. Section 2 describes MPEG-4 Visual sprite coding and the tools used. In Section 3, the proposed coding scheme and its differences to standardized MPEG-4 Visual sprite coding are explained. Section 4 shows experimental results and the last section summarizes the paper.

2. AUTOMATIC SPRITE CODING USING MPEG-4

The MPEG-4 Visual standard assumes that the separated foreground and background objects of the video content are already known. This is exemplarily depicted in Fig. 1. Therefore, sprite generation and object segmentation have to be done in a preprocessing step. The tools used herein, are presented next.

2.1. Global motion estimation

The performance of sprite coding critically depends on the accurate estimation of the background motion. Therefore, it is important to apply a global motion estimation technique with

978-1-4244-7133-1/10/$26.00 ©2010 IEEE

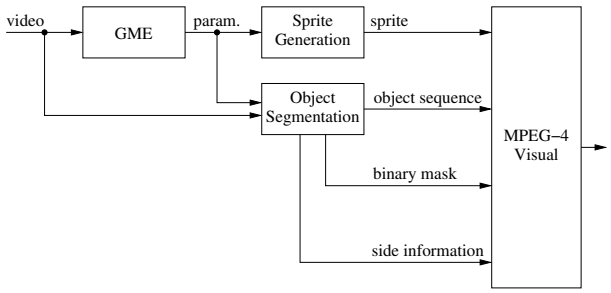


Fig. 1. Automatic sprite coding using MPEG-4 Visual. Since in MPEG-4 Visual, sprite coding is already standardized, the encoder only has to automatically generate sprite, object sequence and binary masks.

very accurate estimation of the higher-order motion parameters. The global motion estimation algorithm used in this work is based on Dufaux et al. [3] and Krutz et al. [4], respectively. It is a gradient descent-based approach using feature tracking as initialization and image pyramid decomposition for estimation refinement. The algorithm decomposes the frames that are subject to motion estimation into an image pyramid. It contains two downsampled versions and the original frames. Other than in [3], it comprises an upsampled representation as well. The algorithm then performs a gradient descent step in every layer of the image pyramid and takes the parameters from the step before as initialization. Additionally, the first gradient descent step is initialized translationally by a feature tracking method.

2.2. Background sprite generation

Background modeling means the description of the background of a video sequence. Application scenarios include video analysis for surveillance systems as well as panoramic image generation in state-of-the-art digital cameras. Here, background models are needed for sprite-based video coding as defined in MPEG-4 Visual [1]. The algorithms used herein, i.e. generation of single background sprites, is based on the work by Kunter [6] and Farin [5].

A single background sprite models the background of a given sequence in one single image. This image is usually of large size and contains only the pixels from the background of the sequence. For the creation of a single sprite, an arbitrary reference frame is chosen. For best results, i.e. minimal distortion in the background sprite image, the reference frame is the center frame in terms of camera pan. All other frames of the sequence are warped into the coordinate system of the reference. First, short-term parameters are calculated using the global motion estimation approach presented in Section 2.1. These short-term parameters are then accumulated by simple matrix multiplication. By transforming all frames into the coordinate system of the background sprite using long-term parameters, a stack of size $M \times N \times S$ is created where $M \times N$ is the resolution of the final background sprite and S is the number of frames in the sequence. The images in this stack are then blended together to generate the background sprite. Blending filters normally used are linear filters like the mean or non-linear filters like the median. However, even more complex approaches like Wiener filtering can be utilized to reach an optimal result.

2.3. Object segmentation

The object segmentation algorithm used herein has been presented before [7]. Is is a background subtraction-based approach.

Using common background sprites, i.e. single or multiple background sprites, for foreground object segmentation in a background subtraction method can lead to non-optimal results. This is due to the fact that, besides the mapping of pixel content into the coordinate system of the sprite image, a second mapping has to be performed for background frame reconstruction. This second mapping may result in distortion that is due to errors in motion estimation and non-ideal interpolation. If the background frame representation is distorted, background regions in the final binary object mask may be oversegmented. The quality of the background model can be enhanced if local background sprites are used (cf. [7]) for background subtraction. Therefore, the algorithm used herein uses this model for background representation. The algorithm first performs a short-term global motion estimation between every two successive frames of the sequence to generate a set of short-term homographies. These are then used to generate a local background sprite for a given reference frame I_{ref}. The subsequent background subtraction produces a very exact error frame E_{ref} that is afterwards processed using a weighted mean thresholding approach to produce a binary foreground object mask B_{ref}.

3. AUTOMATIC SPRITE CODING USING H.264/AVC

Fig. 2 shows the encoder of the proposed automatic sprite coding scheme using H.264/AVC. The preprocessing techniques that divide the video sequence into foreground object sequence and background sprite model are equal to the ones used in the MPEG-4 Visual setup, cf. Section 2. Since other than in MPEG-4 Visual no arbitrary shaped video objects can be defined in H.264/AVC, both foreground object mask and binary foreground mask have to be adjusted to the macroblock structure, i.e. 16×16 block size. This is due to the fact that coding performance is bad if foreground objects arbitrarily end inside a macroblock. In that case, a high-frequency border exists inside the macroblock that cannot be coded efficiently using the means of H.264/AVC. If the foreground object sequence is adjusted, a macroblock either contains foreground data or nothing at all, which makes it very easy to encode.

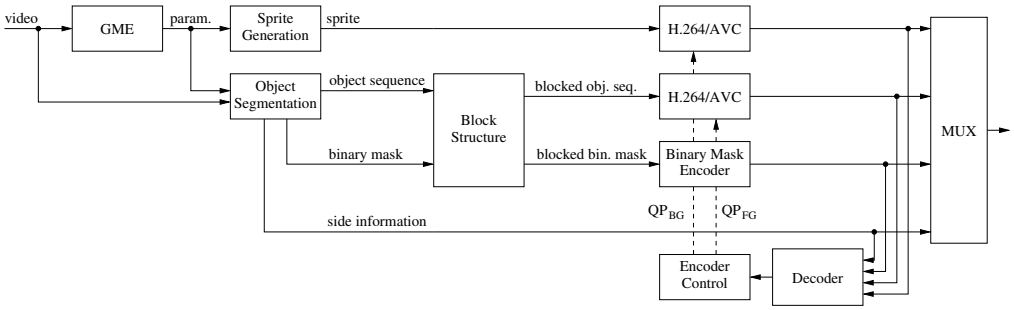


Fig. 2. Automatic sprite coding using H.264/AVC. Depicted is the proposed encoder setup. Global motion estimation, sprite generation and object segmentation are implemented equally to the setup from Section 2. However, to fit the H.264/AVC macroblock structure, object sequence and binary masks are adjusted. A local decoder sets up encoder control that chooses optimal foreground and background quantization.

The background sprite, which actually is a single image and can therefore be coded as INTRA picture, and blocked foreground object sequence are coded using H.264/AVC. The binary foreground mask is encoded using a binary mask coder that uses the binary arithmetic coder *M-Coder*, which is specified in the H.264/AVC standard [11]. The additional side information contains short-term global motion parameters derived using the GME approach presented above.

A local decoder reconstructs the video sequence to set up encoder control that chooses the optimal quantization step sizes for foreground sequence and background sprite, which is based on the authors' previous work [12].

Sequence	Resolution	Frames	FPS	BD-rate
Allstars	352×288	250	25	-13.23%
Biathlon	352×288	200	25	-16.87%
Mountain	352×192	100	25	-2.49%
Stefan	352×240	300	30	-21.36%

Table 1. Test sequences used in the experimental evaluation and mean bit rate savings between proposed approach and H.264/AVC (BD-rate) in the lower bit rate range, since an objective upper bound for sprite reconstruction exists.

4. EXPERIMENTAL EVALUATION

Now that an optimized, H.264/AVC-based automatic sprite coder is available, it is compared with common H.264/AVC coding.The test sequences used herein and the average bit rate savings are shown in Table 1. Additionally, Fig. 3 shows rate-distortion curves for all test sequences including the curves for sprite coding using MPEG-4 Visual with the same preprocessing techniques. H.264/AVC-based sprite coding outperforms MPEG-4 Visual sprite coding over the complete bit rate range. The bit rate could actually be decreased by up to 50%. When comparing H.264/AVC with H.264/AVC+ASC, in the lower bit rate ranges, average savings up to 21% could be reached. Average bit rate savings have been computed in the lower bit rate ranges only, which is due to the fact that in sprite coding, an objective upper bound exists for PSNR which results from maximum reconstruction quality of the background.

5. SUMMARY

We have shown a coding scheme that combines the advantages of state-of-the-art video coding standard H.264/AVC and sprite coding which has been standardized in MPEG-4 Visual. Sophisticated preprocessing techniques, i.e. global motion estimation, object segmentation, and background sprite generation, enable the coding scheme to perform almost optimal in terms of rate-distortion-performance. The new coding scheme clearly outperforms traditional MPEG-4 Visual-based sprite coding and also performs better than H.264/AVC in the lower bit rate ranges where up to 21% savings can be reached.

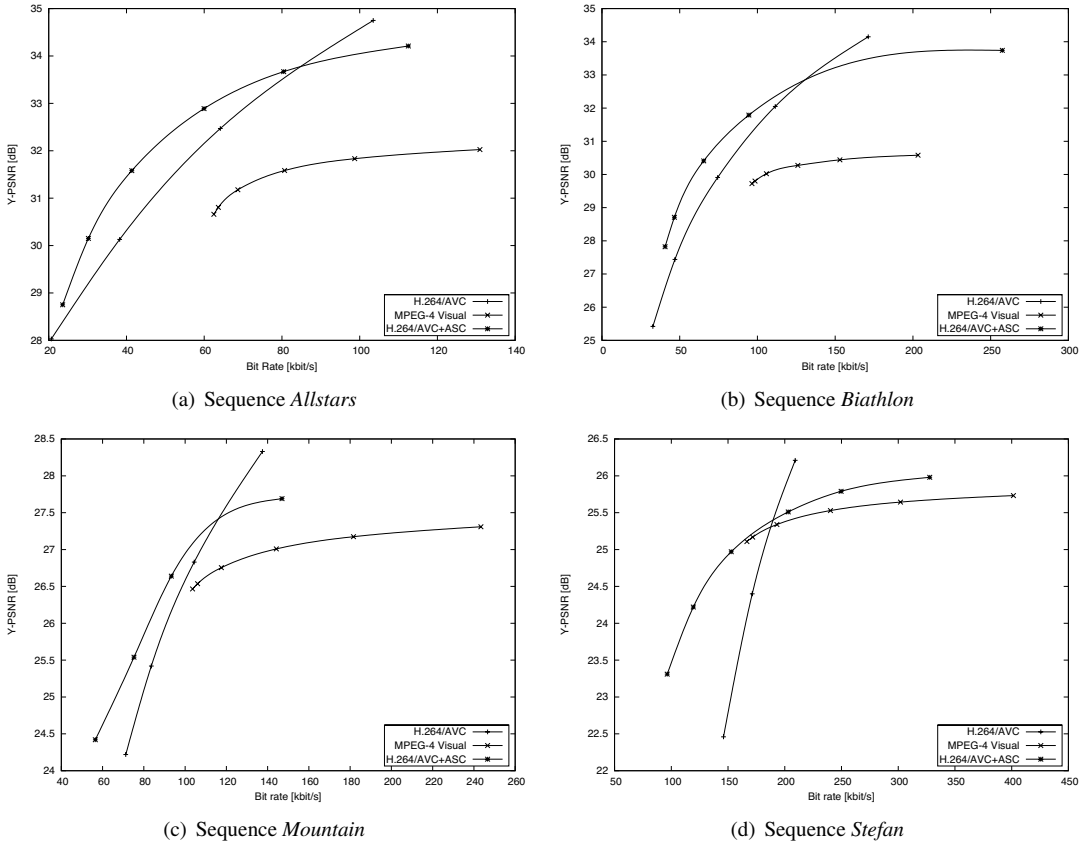
6. REFERENCES

[1] T. Sikora, "The MPEG-4 video standard verification model," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 7, no. 1, pp. 19–31, Feb 1997.

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

[3] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *Image Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 497–501, Mar 2000.

[4] A. Krutz, M. Frater, M. Kunter, and T. Sikora, "Windowed image registration for robust mosaicing of scenes



(a) Sequence *Allstars*

(b) Sequence *Biathlon*

(c) Sequence *Mountain*

(d) Sequence *Stefan*

Fig. 3. Rate-distortion results comparing common H.264/AVC coding to sprite coding in MPEG-4 Visual Main Profile and proposed automatic sprite coding using H.264/AVC (H.264/AVC+ASC).

with large background occlusions," in *Proc. IEEE International Conference on Image Processing*, Oct 2006, pp. 353–356.

[5] D. Farin, *Automatic video segmentation employing object/camera modeling techniques*, Ph.D. thesis, Eindhoven University of Technology, Dec 2005.

[6] M. Kunter, *Advances in sprite-based video coding - towards universal usability*, Ph.D. thesis, Technische Universität Berlin, Jan 2008.

[7] A. Krutz, A. Glantz, T. Borgmann, M. Frater, and T. Sikora, "Motion-based object segmentation using local background sprites," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, Taipei, Taiwan, Apr 2009.

[8] P. Ndjiki-Nya, D. Bull, and T. Wiegand, "Perception-oriented video coding based on texture analysis and synthesis," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 7-10 2009, pp. 2273 –2276.

[9] M. Bosch, F. Zhu, and E.J. Delp, "Video coding using motion classification," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, 12-15 2008, pp. 1588 –1591.

[10] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, Jul 2003.

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

[12] A. Krutz, A. Glantz, M. Frater, and T. Sikora, "Rate-Distortion Optimization for Automatic Sprite Video Coding using H.264/AVC," in *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP 2009)*, Cairo, Egypt, Nov 2009, IEEE Signal Processing Society, IEEE.