# Background Modeling for Video Coding: From Sprites to Global Motion Temporal Filtering

Andreas Krutz, Alexander Glantz, and Thomas Sikora
Communication Systems Group
Technische Universität Berlin
10587 Berlin, Germany
E-mail: {krutz,glantz,sikora}@nue.tu-berlin.de

*Abstract*— **Techniques for modeling the background of a video sequence can be useful in alternative video coding approaches. Sprite coding has been evolved to provide high quality decoded video frames after transmission by a reduced amount of bits. However, it has also been shown that this works only for a certain kind of video sequences. It also meets quality limits due to the Sprite generation step. To tackle this problem, multiple Sprites have been proposed. Considering the improved quality coming with multiple Sprites, a method has been developed which provides an optimal quality, i.e. local background Sprite generation. This method can be used to conduct Global Motion Temporal Filtering (GMTF) of distorted video frames. In a first application, GMTF is applied as a post-processing deblocking filter in a video coding environment, which is experimentally evaluated.**

## I. Introduction

Model-based video coding has become a significant alternative to common hybrid video coding approaches. A very well-known technique in this area is called Sprite coding, which has shown great potential for bit rate reduction in scenes with high camera motion. It has been part of the standardization process of MPEG-4 [1] and its potential has further been outlined in [2]. Since then, a number of methods have been proposed generating a background Sprite of a given input video as a background model to be coded and transmitted separately [3], [4], [5]. A major challenge of this technique is the accurate generation of the Sprite, i.e. the background model, and reconstruction of the video frames from the Sprite. It has turned out that there is a trade-off between the number of frames used for Sprite generation and the distortion of the reconstucted frames. The more frames are included in the Sprite, the smaller is the bit rate, since more frames in a video scene can be predicted from a single Sprite. A Sprite thus serves as a very compact code for many background images. However, the distortion also increases with a rising number of frames included in the Sprite. This trade-off especially occurs if the input video sequence contains large camera pans. Therefore, researchers have started to develop a more effecient way of creating a background model, so-called multiple Sprites. Various approaches have been published based on different techniques, but almost all relying on the analysis of the camera motion [6], [7], [8], and [9].

Depending on the technique used, a given sequence is divided into a number of Sprite partitions. Farin et. al [6] have proposed an algorithm with minimum coding cost within an MPEG-4 environment. Kunter et. al [7] have shown the use of multiple Sprites within a model-based video coding scheme using H.264/AVC. During these achievements, it can be seen that the trade-off between bit rate and accuracy of the reconstructed frames from multiple Sprites still remains. With a rising number of Sprite partitions the accuracy of the reconstruced frames increases, but the coding efficiency decreases.

However, it is possible to build a background model for a given sequence with highest accuracy, the so-called local Sprites [10]. Generating a background model using the same technique as for common Sprite generation is examined for each frame of the sequence. This leads to an optimal modeling of the background of each frame excluding any foreground objects. An application of this method is automatic object segmentation, where the local Sprites are used as a background model for a common background subtraction algorithm.

Building a local Sprite for each frame has brought another effect. Aligning a number of consecutive frames into the coordinate system of a reference frame can also be seen as an image stack including the reference frame with its several versions. If we apply this technique to a noisy input sequence, by blending the aligned images, the resulting local Sprite is a temporally filtered version of the reference frame. The impact of this technique for video coding is outlined in this work.

The paper is organized as follows. Section 2 provides a short overview of the process starting from conventional Sprites to Global Motion Temporal Filtering (GMTF). The next section introduces theoretically the GMTF-approach in a coding environment and its first application as a post-processing deblocking filter. Section 4 illustrates first experimental results and the last section concludes this work.

## II. From Single- to Multiple-Sprites to Global Motion Temporal Filtering

### A. Single Sprites

A so-called single Sprite models the background of a given sequence in one single image. This image is of large dimension and contains usually only the pixels from the background of the sequence. For the creation of a single Sprite, a reference frame is chosen and all other frames of the sequence are warped into the coordinate system of the reference. For that,

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

so-called long-term higher-order motion parameters are computed that describe this transformation. The complete method is outlined in [3].

### B. Multiple Sprites

Multiple Sprites are used to optimize the bit rate versus quality trade-off, especially for sequences with large camera pans. The well-known "Stefan" sequence has been used very often to evaluate Sprite techniques. For example, the multiple Sprite generation algorithm proposed in [7] leads to three partitions. The algorithms proposed in [6] and [8] generate four partitions by use of the same test sequence by automatically segmenting the sequence. The approach presented in [9] focuses more on the quality of the reconstructed frames from the Sprite and produces six partitions. This tendency generating three, four and six partitions of this example has led to the idea to build a local Sprite for each frame of the input sequence, which is introduced next.

### C. Global Motion Temporal Filtering (GMTF)

The term local background Sprite specifies a model of the background. Other than general background Sprites one model is built for every frame and not one model for the whole video sequence. Only the local temporal neighborhood of each reference frame is taken into account for Sprite generation. The dimensions of a local background Sprite match those of the reference frame. Our goal is to minimize distortion in background regions. When a background frame is reconstructed from a general background Sprite, distortion can be severe. This is due to accumulated errors in the global motion estimation, non-ideal interpolation and the double mapping into the coordinate system of the background Sprite and back. Our proposed local Sprite algorithm is described in [10].

However, beside building an accurate background model, this technique has another advantage. Having the stack of aligned images corresponding to the reference frame coordinate system, global motion temporal noise filtering can be performed by blending all pixel canditates related to the reference frame together. This means that if the frames in the stack are distorted version, e.g. caused by a coding and decoding process, noise can be reduced by this type of filtering.

## III. GMTF for Post-Processing

### A. Theoretical Consideration of GMTF-deblocking

Blocking artifacts after encoding and decoding are coding noise. We can use the temporal mean filtering idea for noise reduction.

It is assumed that a number of distorted versions $Y$ from an original image $X$ are available after registration using global motion estimation. The local Sprite approach essentially identifies and registers these noisy versions. Consider the pixel registered value $y_k(m,n)$ of the k$th$ frame as the sum of the original pixel $x(m,n)$ and a value from the noise signal $n_k(m,n)$ :

$$y_k(m,n) = x(m,n) + n_k(m,n) \tag{1}$$

The mean value over all noisy versions $y_k(m,n)$ is:

$$y(m,n) = \frac{1}{N}\sum_{k=1}^{N} y_k(m,n) = x(m,n) + \underbrace{\frac{1}{N}\sum_{k=1}^{N} n_k(m,n)}_{r(m,n)}. \tag{2}$$

Uncorrelated white coding noise is assumed with the variance $\sigma_n^2$ and the autocorrelation matrix :

$$R_{nn} = \begin{pmatrix} \sigma_n^2 & 0 & \dots \\ 0 & \sigma_n^2 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{3}$$

We now show that the variance of the noise is reduced by the factor $N$ (number of registered frames). The mean noise signal is $r(m,n)$. The variance can be calculated as :

$$\sigma_r^2 = E[R^2(m,n)] = \frac{1}{N^2}\sum_{i=1}^{N}\sigma_n^2 = \frac{\sigma_n^2}{N} \tag{4}$$

Thus, the variance of the coding noise has been reduced by the factor $N$ by averaging pixel values of N noisy versions. Having this result we can turn to our deblocking problem in a codec environment. If we are able to apply a noise reduction using the GMTF-approach we are able to increase coding efficiency.

One of the major problems in a common hybrid video codec are the blocking artifacts. For our theoretical estimation we treat these blocking artifacts as above as temporally independent white noise. For independent Gaussian sources, the rate-distortion function can be formulated as follows :

$$D_{\text{nf}}(R_{\text{nf}}) = 2^{-2R_{\text{nf}}}\sigma_x^2, \tag{5}$$

where $D_{\text{nf}}(R_{\text{nf}})$ is the distortion, $R_{\text{nf}}$ is the bit rate, and $\sigma_x^2$ is the variance of the coded pixel amplitude for a single frame. We now apply a decoder noise reduction using N versions. With Equ. 8 the new rate-distortion function is :

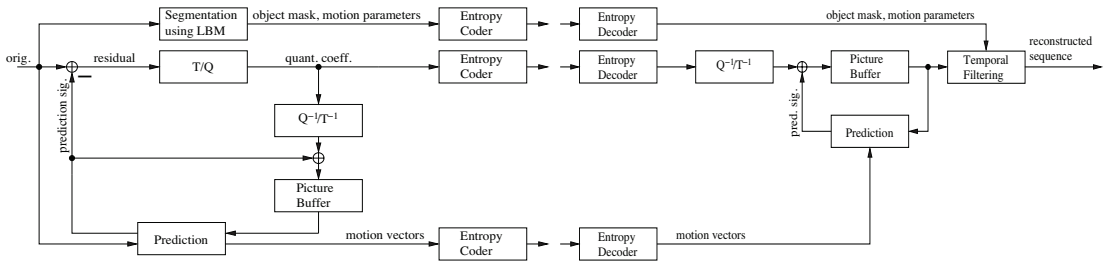
$$D_{\text{f}}(R_{\text{f}}) = 2^{-2R_{\text{f}}}\frac{\sigma_x^2}{N}. \tag{6}$$

With $D_{\text{nf}}(R_{\text{nf}}) = D_{\text{f}}(R_{\text{f}}) = D$:

$$2^{-2R_{\text{nf}}} = 2^{-2R_{\text{f}}}\frac{1}{N} \tag{7}$$

$$\Rightarrow R_{\text{f}}(D,N) = R_{\text{nf}}(D) - \frac{1}{2}log_2(N) \tag{8}$$

We thus obtain a bit rate saving of $\frac{1}{2}log_2(N)$ per pixel by applying a noise reduction using averaging of frames. Equ. 8 is valid for $N \cdot D \leq \sigma_x^2$.




(a) Encoder (b) Decoder

Fig. 1. Coding scheme using GMTF post-processing

### *B. Coding environment using GMTF-deblocking as post-processing step*

The deblocking filter using GMTF is used as a post-processing step. Two aspects regarding the foreground objects have to be handled. First, the foreground objects information has to be added after the filtering process and second, the deblocking issue in foreground objects regions. For that, the following hybrid approach is considered. At the encoder, the input video signal is processed by an automatic object segmentation method using local background modeling (LBM) ([11]). The output of this segmentation step is a binary mask for each frame which defines foreground objects and background region along the video sequence. This binary mask sequence is encoded using the same binary mask encoder used in [7] and transmitted as side information. At the decoder, the binary mask is used to extract the foreground object regions from the decoded video sequence where the common deblocking filter used in H.264/AVC is applied. Then, GMTF is performed on the decoded video sequence. In the final reconstruction step, the filtered forground objects are mapped on the GMTF-processed frames. As a result, pixels of the background regions are temporally filtered and the foreground object regions are filtered spatially using the common H.264/AVC deblocking filter. The block diagram of the described method is given in Fig. 1.

## IV. Experimental Results

We compared our approach with the standardized H.264/AVC in-loop filter [12]. According to the theoretical consideration of Section III A, we use the PSNR metric to measure the quality in a decoded video frame. We chose three test sequences to show the performance of the deblocking using GMTF. The sequence "Biathlon" (352x288, 200 frames) represents sport videos with large camera pans and zooms including single and multiple objects. Additionally, we took into account two sequences recorded from the BBC-documentary "Planet Earth" called "Birds" (720x576, 100 frames) and "Desert" (720x400, 240 frames), also with significant global motion of camera. Three different coding environments were considered for the experiments. Except the deblocking filter, all encoder settings were fixed to ensure a fair comparison. The JSVM v.9.1 was used as the reference H.264/AVC encoder with hierarchical B-frames prediction structure and GOP-size 15. The GMTF post-processing deblocking filter improved the quality of decoded video. Figure 3 shows the rate-distortion curves for the three test sequences considered. It can be seen that the GMTF post-processing approach outperformed the common in-loop filter used in H.264/AVC significantly. It is obvious that the performance of the deblocking approach increased in the lower bit rate ranges when blocking artifacts appear. To emphasize this, bit rate saving curves were drawn. The savings achieved using the GMTF-deblocking filter are large at all sequences considered. It is noticable that especially at the test sequences with TV-resolution, where a larger amount of blocks appear, bit rate savings up to 26% were achieved. Most importantly, subjective quality was increased drastically. Figure 2 depicts a typical result, valid for all test sequences. Blocking artifacts are almost completely eleminated, while details like edges were preserved.

(a) H.264+In-loop deblocking filter, PSNR=28.63, R=53.71 kbits/s (b) H.264+GMTF, PSNR=29.38, R=51.44kbits/s

Fig. 2. Details of decoded frame 160, sequence "Biathlon"

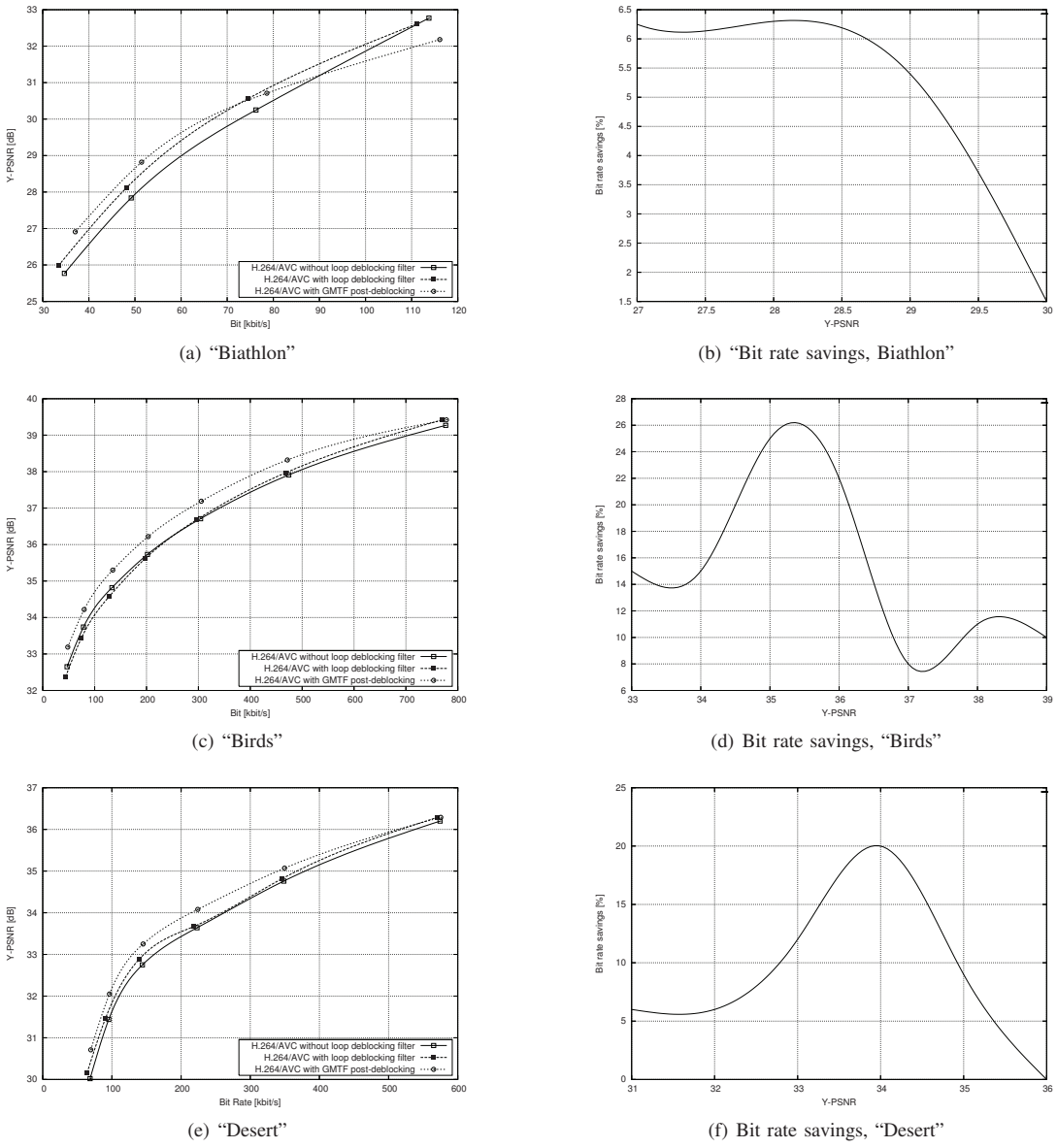
## V. Summary

We have shown that our new deblocking method outperforms the state-of-the-art approach significantly. It is possible to set up a coding scheme to transmit the input video with a lower bit rate and enhance the quality at the decoder using the post-processing step proposed. Both objective and subjective quality are improved. An open issue is to find the optimal number of frames taken into account for GMTF post-processing.

## References

[1] T. Sikora, "The mpeg-4 video standard verification model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 19–31, February 1997.

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




(a) "Biathlon"

(b) "Bit rate savings, Biathlon"

(c) "Birds"

(d) Bit rate savings, "Birds"

(e) "Desert"

(f) Bit rate savings, "Desert"

Fig. 3. Comparison of H.264/AVC loop deblocking filter and a post-processing deblocking filter using GMTF


[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] Y. Lu, W. Gao, and F. Wu, "Fast and robust sprite generation for MPEG-4 video coding," in *IEEE Pacific Rim Conference on Multimedia (PCM'01)*, Bejing, China, Oct. 2001.

[5] F. Pereira and T. Ebrahimi, *The MPEG-4 Book*. Springer, 2002.

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

[7] 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.

[8] G. Ye, J. Xu, G. Herman, and B. Zhang, "A practical approach to multiple super-resolution sprite generation," in *8th Workshop on Multimedia Signal Processing (MMSP'08)*, Cairns, Australia, Oct. 2008.

[9] A. Krutz, A. Glantz, M. Haller, M. Droese, and T. Sikora, "Multiple background sprite generation using camera motion characterization for object-based video coding," in *3DTV Conference 2008, The True Vision Capture, Transmission and Display of 3D Video, May 2008, Istanbul, Turkey*, Istanbul, Turkey, May 2008.

[10] A. Krutz, A. Glantz, M. Frater, and T. Sikora, "Local background sprite generation," in *2008 International Workshop on Local and Non-Local Approximation in Image Processing, A satellite event of the 16th European Signal Processing Conference (EUSIPCO 2008)*, Lausanne, Switzerland, Aug. 2008.

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

[12] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 614–619, July 2003.