

A MUCOM TECHNICAL REPORT: TEMPORAL FILTER WITH BILINEAR INTERPOLATION FOR ROI VIDEO CODING

L. S. Karlsson, R. Olsson and M. Sjöström

Dept. of Information Technology and Media, Mid Sweden University, 851 70 Sundsvall, Sweden

ABSTRACT

In videoconferencing and video over the mobile phone, the main visual information is found within limited regions of the video. This enables improved perceived quality by region-of-interest coding. In this paper we introduce a temporal preprocessing filter that reuses values of the previous frame, by which changes in the background are only allowed for every second frame. This reduces the bit-rate by 10-25% or gives an increase in average PSNR of 0.29-0.98 dB. Further processing of the video sequence is necessary for an improved re-allocation of the resources. Motion of the ROI causes absence of necessary background data at the ROI border. We conceal this by using a bilinear interpolation between the current and previous frame at the transition from background to ROI. This results in an improvement in average PSNR of 0.44 – 1.05 dB in the transition area with a minor decrease in average PSNR within the ROI.

1. INTRODUCTION

In videoconferencing applications and video over mobile phones, the quality is affected by the limited bandwidth imposed by the channel. Most of the visual information is communicated by particular areas in the video sequence, such as the face. The overall perceived quality can in such cases be improved by focusing on improving quality in these regions-of-interest (ROI) at the expense of the background quality. How can resources be redistributed from the background to the ROI using temporal information independent of the codec?

To reallocate resources, previous research has mainly targeted spatial methods using two main approaches. One controls quantization parameters within the codec [1-3] and the other applies preprocessing using low-pass filtering to remove details [1], [4-5]. The former makes a direct integration with the rate-distortion function possible within the codec but can introduce “blockiness” due to coarse quantization parameters. The latter, on the other hand, avoids codec dependencies by applying pre-filtering, whose resulting error generally is less disturbing.

Temporal approaches include mainly those coding ROI and background in separate layers, enabling a lower frame-rate for the background than the ROI. These approaches are either dependent on using object-based coding within the MPEG-4 standard [6-7], or using special implementations whereby these layers are manually separated and transmitted as two separate sequences [8]. Normally, the latter requires an adaptation on the receiver side, but by ensuring that the bit stream stays conformed to the standard, no modifications are necessary at the decoder [9]. Blocks not affecting the ROI are then skipped by editing the compressed sequence. Earlier codec independent approaches include the temporal filter in [3], which averages out differences between the backgrounds of two frames.

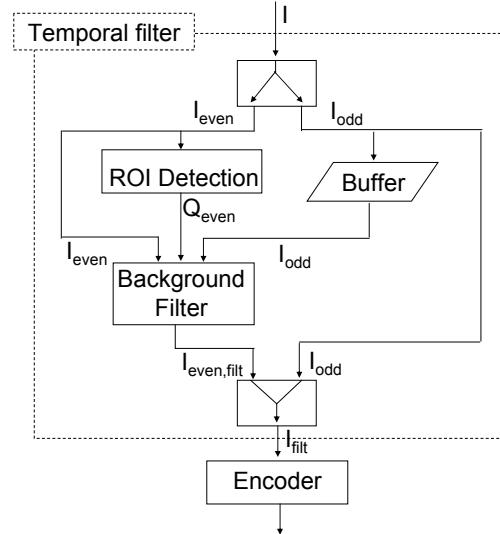


Figure 1: The block diagram of our proposed method.

The algorithm we propose in this paper extends the methods in [3] and [9] into a pre-filtering approach that codes the background with half the frame-rate of the ROI. (See figure 1.) The difference to the layer-based methods is that the background and the ROI are not divided into separate images. Therefore, the resulting sequence can be encoded by any standard using block-based coding. This is achieved by copying background blocks of the previous frame into the current frame for every second frame. The

filtered block is skipped in the predictive encoding since there is no change from the previous frame. A bilinear interpolation of the transition from ROI to background is also introduced in order to deal with problems caused by the non-stationary ROI border.

This paper is organized as follows. First the algorithm is presented in section 2, with a theoretical analysis of the temporal filter in section 3. The results of experiments is found in section 4 followed by the conclusions in section 5.

2. PROPOSED ALGORITHM

The proposed algorithm performs a temporal filtering to achieve a decrease in frame-rate of a factor two in the background without altering the shape of the sequence. An overview of the approach can be found in figure 1. Assume frame number $N = 1, 2, \dots, M$. Odd frames are saved in a buffer for later processing but also transmitted directly to the encoder. In the case of even frames, a quality map is determined based on a binary map of the detected ROI as in [4]. Low-pass filtering of the binary detection map results in a quality map Q that contains the location of the ROI and the distance to the border. In the next step, only the pixels (m, n) with $Q(m, n) < 1/A$ as defined in [4], are considered background and therefore filtered. The actual filtering implies that the background pixels of even frames are assigned values from the previous odd frame. The transition from ROI to background is made smoother by using the option to bilinearly interpolate pixels with $0.02 \leq Q(m, n) < 1/A$, using the value of pixel (m, n) in both the even and the previous odd frame.

2.1 THE FILTERING OF THE BACKGROUND

The filtering of the background is performed on a block basis in every even frame. The values of the resulting filtered frame are determined by combining the values from even frames I_{even} with the previous odd frame I_{odd} , allowing only the ROI to contain new information. Thus for every block (p, q) in I_{even} the corresponding block in the filtered frame is

$$I_{filt}(p, q) = \begin{cases} I_{even}(p, q), & \text{if } \max(Q(p, q)) \geq 1/A \\ I_{odd}(p, q), & \text{otherwise} \end{cases} \quad (1)$$

where blocks containing ROI pixels are determined by $\max(Q(p, q)) \geq 1/A$.

The border between the ROI and background is not stationary. This leads to problems similar to those when combining layers in an MPEG-4 decoder [6]. Large movements of the ROI from frame to frame will cause the background in even frames to be assigned values from the

ROI in the previous odd frame. (See figure 2.) Artifacts also occur if the current ROI covers the previous background.

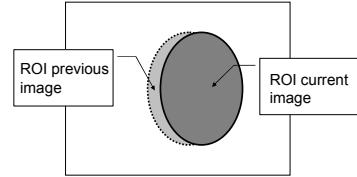


Figure 2: Position of the ROI from frame to frame.

These artifacts are compensated by applying a gradual transition of quality from the ROI to the background. The transition region contains all blocks (p, q) , where $0.02 \leq \max(Q(p, q)) < 1/A$ is true, since the value of Q indicates distance to the ROI border. Adding bilinear interpolation of this region gives the following modification to the background filter. For every block (p, q) in I_{even} , the filtered frame is

$$I_{filt}(p, q) = \begin{cases} I_{even}(p, q), & \text{if } \max(Q(p, q)) \geq 1/A \\ f_{bl}(I_{even}, I_{odd}), & \text{if } 0.02 \leq \max(Q(p, q)) < 1/A \\ I_{even}(p, q), & \text{otherwise} \end{cases} \quad (2)$$

where for each pixel (m, n) belonging to block (p, q) the bilinear interpolation of the transition region $Q \in [0.02, 1/A]$, if $\alpha = A \cdot Q(m, n)$, becomes $f_{bl}(B, C) = \alpha \cdot B(m, n) + (1 - \alpha) \cdot C(m, n)$. Bilinear interpolation is chosen based on its simplicity and results in a blurred transition region without sharp artifacts.

3. ANALYSIS

The following assumptions are made for a qualitative analysis of the temporal filter. The bilinear interpolation is not considered. The sequences are encoded using the JM 10.1 H.264 codec for the High Profile [10], including the adaptive context-based arithmetic coding (CABAC) [11]. The sequence consists of only one I-frame followed by $M - 1$ P-frames. The motion of the background is assumed to be uniform. For each background macro-block in an even frame $N = 2k$ that is used as a reference to a macro-block in frame $N = 2k + 1$, there exists a corresponding reference macro-block in the previous odd frame $N = 2k - 1$. Furthermore, the number of motion vectors assigned by the codec to each macro-block in frame $2k + 1$ is equal to the number of motion vectors in the referenced macro-blocks in frames $2k$ and $2k - 1$.

Under these assumptions, the codec skips all background macro-blocks in frame $2k$ because there is no

change from frame $2k - 1$. Thus no bits are assigned by the codec to describe the type of prediction, motion vectors or prediction error.

The non-filtered frame $2k + 1$ has approximately the same number of bits describing the type of the prediction within each macro-block as when coding the original sequence. If there are T macro-blocks in one frame, the total length of all background motion vectors in filtered frame $2k + 1$ becomes

$$\sum_{l=0, MB_l \notin ROI}^{T-1} \left\| mv_{2k+1,l,filt} \right\| \leq \sum_{l=0, MB_l \notin ROI}^{T-1} \left(\left\| mv_{2k,l} \right\| + \left\| mv_{2k+1,l} \right\| \right), \quad (3)$$

where $mv_{k+1,l,filt}$ is the total of motion vectors in each block of the filtered sequence of frame $2k+1$ and $mv_{N,l}$ is the corresponding motion vectors of the original sequence of frame N . The assumption of uniform motion implies approximately equal motion vectors in frame $2k$ and $2k+1$ of the original sequence, which leads to

$$\sum_{l=0, MB_l \notin ROI}^{T-1} \left\| mv_{2k+1,l,filt} \right\| \approx 2 \cdot \sum_{l=0, MB_l \notin ROI}^{T-1} \left\| mv_{2k,l} \right\|. \quad (4)$$

Depending on the previously encoded motion vectors, which include those in the ROI, CABAC adapts to probabilities in the filtered sequence. For motion vectors where $\|mv_{k,l}\| > 9$, the prefix “9” is encoded using the adaptive context-based arithmetic coding, as all motion vectors with $\|mv_{k,l}\| \leq 9$. The suffix consisting of $\|mv_{k,l}\| - 9$ is encoded by applying a 4th order Exponential Golomb code, which has an exponential growth of number of code words with increasing length. Coding two equal length motion vectors gives a larger total codeword length than when only one twice as long motion vector is coded. This means that the cost in bits to encode the motion vectors of frame $2k + 1$ in the filtered sequence is less than total cost in frame $2k$ and $2k + 1$ for the original sequence, since there are no motion vectors in frame $2k$.

The coding of the prediction error does not decrease the coding efficiency, since the prediction error of frame $2k + 1$ is smaller or equal to the total prediction error of frame $2k$ and $2k + 1$.

4. EXPERIMENTAL RESULTS

The QCIF sequences *carphone*, *foreman* and *closeup* for 10 fps and 15 fps were used in the tests. The sequence *closeup* was created by the authors of the paper and consists of a close-up of a face with a panning outdoor background. The parametric model presented in [12] with experimentally

determined thresholds at 30 % (carphone), 32 % (foreman) and 34 % (closeup) gave a binary detection map. This was used as a base for Q . In percent of the frame the average sizes of the ROI are 32% (carphone), 25% (foreman) and 33% (closeup). Two versions of the background filtering were tested, method 1, utilizing the background filtering in eq. (1) and method 2, including the bilinear interpolation as in eq. (2). The used quality measures for the compressed sequences were either bit-rate or average PSNR in ROI and transition region, respectively, calculated from the intensity component:

$$PSNR = \frac{1}{M} \sum_{j=1}^M 10 \log_{10} \frac{255^2}{\sigma_{e,j,area}^2}, \quad (5)$$

where $\sigma_{e,j,roi}^2$ and $\sigma_{e,i,trans}^2$ are the variance of errors for blocks containing pixels with $Q \in [1/A, 1]$ (ROI), $A = 3$, and $Q \in [0.01, 0.5]$ (transition region), respectively.

For the bilinear interpolation the linear function $\alpha = A \cdot Q(m,n)$ was compared to two alternatives, $\alpha = \sqrt{A \cdot Q(m,n)}$ and $\alpha = (A \cdot Q(m,n))^2$. Tests showed that $\alpha = A \cdot Q(m,n)$ is a good compromise, which gives both an increase of quality in the transition region and a limited decrease of the quality in the ROI. The alternative methods favor either the transition region or the ROI and thus $\alpha = A \cdot Q(m,n)$ is applied in the tests.

Table 1: Bitrate (kbps) for $Qp = 28$

Framerate	Method	Carphone	Foreman	Closeup
10 fps	Original	74,18	70,69	147,18
	1	56,67	63,38	110,58
	2	58,94	66,06	123,60
15 fps	Original	99,53	87,79	186,50
	1	75,19	79,69	141,42
	2	77,99	82,68	157,68

Temporal filtering alone (method 1) saves about 25 % in bit-rate when using a fixed quantization parameter $Qp = 28$. (See table 1.) The exception is for the foreman sequence where the decrease is 10 %, mainly because of the low motion content in the background compared to the ROI. Introducing bilinear filtering (method 2) causes a decrease in bit-rate of 21 % for carphone, 15 % for closeup and 6 % for foreman. The large decrease for the closeup sequence is partially because of misdetections in the skin color detection.

Tests were also performed with a fixed bit-rate by choosing the rate control option of the codec. At 64 bps and 10 fps an improvement within the ROI of 0.98 dB (carphone) and 0.29 dB (foreman) is obtained for method 1.

(See table 2.) A similar improvement was achieved for 15 fps. (See table 3) This is a moderate improvement compared to the decrease in bit-rate when the quantization parameter was fixed. The resources released by the temporal filtering when employing a fixed bit-rate are used by the codec to decrease the error in each frame and particularly the coding of the error after the motion compensation. The released resources are therefore used in the complete image unless some additional control is added.

Table 2: Average PSNR (dB) for 10 fps					
Framerate	Method	Carphone		Foreman	
		ROI	Border	ROI	Border
64 fps	Original	36.16	37.11	36.18	36.34
	1	38.14	34.27	36.47	31.70
	2	38.08	35.02	36.32	32.75
32 fps	Original	34.10	33.83	32.72	33.08
	1	34.88	32.23	32.93	29.82
	2	34.84	32.78	32.85	30.60

Table 3: Average PSNR (dB) for 15 fps					
Framerate	Method	Carphone		Foreman	
		ROI	Border	ROI	Border
64 fps	Original	35.89	35.70	35.09	35.26
	1	36.86	35.02	35.43	31.94
	2	36.80	34.32	35.35	32.80
32 fps	Original	33.00	32.1	31.68	31.99
	1	33.74	31.70	31.99	30.00
	2	33.77	32.14	31.92	30.53

Table 2 and table 3 also show that adding bilinear interpolation to the temporal filter improves the average border PSNR with 0.44 – 0.75 dB (carphone) and 0.53-1.05 dB (foreman). This is achieved with only a minor decrease in average PSNR of the ROI.

The temporal filtering may cause jerkiness in the background for low frame-rates. This could be improved by post-processing, but it is out of the scope of this paper.

5. CONCLUSION

We have presented a temporal pre-processing approach that filters a video sequence temporally by using values from previous frames to cause the encoder to skip macro-blocks in the background for every second frame. This decreases the bit-rate of 10-25 %, assuming fixed quantization parameters, compared to the original sequence. At a fixed bit-rates of 32 kbps and 64 kbps, this results in an increase of average PSNR of 0.29-0.98 dB. Additional methods, such as low-pass filtering of the background, are required to optimize the reallocation to the ROI. Bilinear interpolation of the transition area is also applied to reduce border problems due to large movement of the ROI. This gives an improvement in average PSNR of 0.44 – 1.05 dB of the

transition area without a noticeable quality reduction in the ROI.

6. ACKNOWLEDGEMENT

This work is supported by the Swedish Graduate School of Telecommunications and by the EU Objective 1-programme Södra Skogslän region.

7. REFERENCES

- [1] M.-J. Chen, M.-C. Chi, C.-T. Hsu and J.-W. Chen, "ROI Video Coding Based on H.263+ with Robust Skin-Color Detection Technique," *IEEE Trans. Consumer Electronics*, Vol. 49, pp. 724-730, Aug 2003
- [2] S. Sengupta, S. K. Gupta and J. M. Hannah, "Perceptually Motivated Bit-Allocation for H.264 Encoded Video Sequences," *IEEE ICIP*, Vol. 3, pp. III - 797-800, Sept. 2003
- [3] T. Adiono, T Isshiki, K. Ito, and T. Ohtsuka, "Face Focus under H.263+ Video Coding Standard," *IEEE APCCAS*, pp. 461-464, 2000
- [4] L. S. Karlsson and M. Sjöström, " Improved ROI Video Coding using Variable Gaussian Pre-Filters and Variance in Intensity, " *IEEE ICIP*, vol. 2, pp. 313-316, Sept 2005
- [5] Laurent Itti, " Automatic Foveation for Video Compression Using a Neurobiological Model for Visual Attention, " *IEEE Trans. Image Processing*, Vol. 13, pp. 1304-1318, Oct. 2004
- [6] J.-W. Lee, A. Vetro, Y. Wang and Y.-S. Ho, "Bit Allocation for MPEG-4 Video Coding With Spatio-Temporal Tradeoffs, " *IEEE Trans. Circuits Syst. Video Techn.*, Vol. 13, pp. 488-502, June 2003
- [7] W. Lei, X.-D. Gu, R.-H. Wang, L.-R. Dai and H.-J. Zhang, " A Region Based Multiple Frame-Rate Tradeoff of Video Streaming, " *IEEE ICIP*, pp. 2067-2070, 2004
- [8] J. Meessen, C. Parisot, X. Desurmont and J.-F. Delaigle, "Scene Analysis for Reducing Motion JPEG 2000 Video Surveillance Delivery Bandwidth and Complexity, " *IEEE ICIP*, vol. 1, pp. 577-580, Sept. 2005
- [9] J. Augustine, S. K. Rao, N. P. Jouppi and S. Iyer, " Region of Interest Editing of MPEG-2 Video Streams in the Compressed Domain, " *IEEE ICME*, pp. 559-562, 2004
- [10] H.264/AVC, JM 10.1, <http://iphom.hhi.de/suehring/tm/>
- [11] D. Marpe, H. Schwarz and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *IEEE Trans. Circuits Syst. Video Techn.*, Vol 13, pp. 620-636, July 2003.
- [12] Y.-X. Lv, Z.-Q. Liu, and X.-H. Zhu, "Real-time face detection based on skin-color model and morphology filters, " *Int. Conf. Machine Learning and Cybernetics*, Vol. 5, pp. 3203-3207, Nov. 2003