SPATIO-TEMPORAL FILTER FOR ROI VIDEO CODING

Linda S. Karlsson, Mårten Sjöström and Roger Olsson

Department of Information Technology and Media, Mid Sweden University, SE-851 70 Sundsvall, Sweden
phone: +46 60 148695, fax: +46 60 148830, email: linda.karlsson@miun.se, marten.sjostrom@miun.se, roger.olsson@miun.se,
web: www.miun.se/itm

ABSTRACT

Reallocating resources within a video sequence to the regions-of-interest increases the perceived quality at limited bandwidths. In this paper we combine a spatial filter with a temporal filter, which are both codec and standard independent. This spatio-temporal filter removes resources from both the motion vectors and the prediction error with a computational complexity lower than the spatial filter by itself. This decreases the bit rate by 30-50% compared to coding the original sequence using H.264. The released bits can be used by the codec to increase the PSNR of the ROI by 1.58 – 4.61 dB, which is larger than for the spatial and temporal filters by themselves.

1. INTRODUCTION

The quality of video in mobile phone applications or video-conferencing depends on the bandwidth of the transmission channel. In video sequences with focus on communication between humans, the visual information is primarily concentrated to the facial region. The perceived quality can be improved by reallocating coding resources within the video sequence to enhance the quality of the region-of-interest (ROI). How can codec-independent resource reallocation be improved without increasing the computational complexity?

The main research on reallocating coding resources is focused on codec dependent methods. The two primary approaches include spatial reallocation by controlling quantization parameters [1-2] and temporal approaches dividing the background and ROI into layers to introduce different frame-rates [3-5]. Alterations in the codec make adaptation of the resource reallocation to channel conditions easier because it can be directly integrated with the existing rate-distortion functions of the encoder. The approaches using layers are only supported by MPEG-4 unless alterations of both encoder and decoder are applied. The possibility of generalizing an approach is lost when it becomes dependent on standard or codec. Independent approaches include spatial low-pass filtering in the background [1],[6-7] to remove details and the temporal filter in [8]. The method in [8] releases resources from the background by only allowing changes in the background every second frame. Additional methods, such as low pass filtering of the background or controlling the quantization parameters in the codec, are necessary to redistribute the released resources to the ROI.

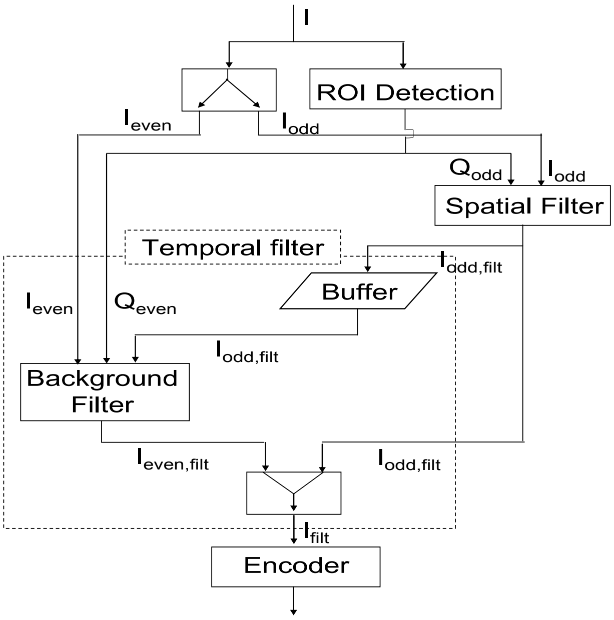

Figure 1 – The block diagram of the proposed method.

The algorithm in this paper combines the spatial approach in [6] with the temporal approach in [8], resulting in the spatio-temporal filter presented in figure 1. This gives a decreased prediction error of the motion compensation, which both releases extra resources and causes the released bits from the temporal approach to be distributed to the ROI. The spatio-temporal filter also has a lower computational complexity than the spatial filter.

This paper is organized as follows. The proposed approach is presented in section 2 followed by a theoretical analysis of the performance of the filter in section 3. This is followed by the presentation of experimental results in section 4 and a conclusion in section 5.

2. THE SPATIO-TEMPORAL FILTER

In the proposed algorithm, the spatial and temporal filters are combined as presented in figure 1. A quality map Q, which contains information about the location and distance to the ROI, is created by low pass filtering of the binary map given by an arbitrary ROI detection. The background ($Q^{m,n} < 1/A$ for all pixels as defined in [6]) of the original frame is low pass filtered. Several spatial low pass filters are used based on the distance to the ROI as in [6]. The transition region from ROI to background ($0.02 \le Q^{m,n} < 1/A$) is, however, processed differently from the rest of the background to ensure a smooth transition in quality. The temporal filter in [8] uses values from the previous frame to remove changes in the background and bilinear interpolation to compensate for large movement of the ROI. This filter is controlled by the same quality map as for the spatial filter. Therefore the filters can be combined. The spatial filter is applied to every odd frame, assuming frame number $N = 1,2,\ldots,M$ with the total number of frames M. Even frames are temporally filtered using the previously buffered spatially filtered odd frame.

2.1 Spatial filter

The spatial filter in [6] can be combined with the temporal filter without any alterations, since only information within the same frame is used in the calculations. The filter uses a limited number of low-pass filters. The standard deviations of the filters have a uniform distribution between a minimum and a maximum. The quality map Q is used to determine which filter is applied for each pixel in the transition region from the ROI border to the background. The closer to the border the pixel is located, the smaller standard deviation of the applied low-pass filter. The usage of several filters based on the distance to the ROI enables strong smoothing (large standard deviation) of the main part of the background without causing disturbing border artefacts.

2.2 Temporal filter

A temporal filtering approach is described in [8] that reduces the background frame-rate to half. This is achieved by reusing blocks of the background in the previous odd frame in the current even frame. The size of the blocks is arbitrary and only affects to which region the included pixels belong. However, smaller sized blocks are to be favoured to ensure codec independency. Non-covered areas at the border, caused by movements of the ROI from frame to frame, are masked by applying an additional bilinear filter based on the distance to the ROI. The distance is determined by the quality map Q. Alterations to the background filter in [8] are necessary because the present spatio-temporal uses filter information from the odd spatially filtered frame. The background pixels of even frames used in the bilinear interpolation must also be spatially low-pass filtered. Another technicality is that the determination of the ROI is block-based for the temporal filter while pixel-based for the spatial filter. Hence, there exist pixels classified as background by the spatial filter that are simultaneously part of a block classified as ROI by the temporal filter. Therefore, these pixels must be detected and spatially low-pass filtered to ensure that the border stays as smooth as possible in all frames.

These alterations result in the following background filter using values from the even frame I_{even} and the previous spatially filtered odd frame $I_{odd,filt}$. First it is decided whether a block (p,q) of size $B \times B$ belongs to the ROI, the transition area or the background :

$$I^{p,q}_{even,filt} = \begin{cases} f_{ROI}(I^{p,q}_{even}, I^{p,q}_{odd,filt}), & if\ Q^{p,q}_b \ge \dfrac{1}{A} \\ f_{Trans}(I^{p,q}_{even}, I^{p,q}_{odd,filt}), & if\ 0.02 \le Q^{p,q}_b < \dfrac{1}{A} \\ I^{p,q}_{odd,filt}, & otherwise \end{cases}$$

where $Q^{p,q}_b$ is the maximum value of the quality map $Q^{m,n}$ within the block. Then each pixel (m,n) within the block is treated according to the following functions:

$$f_{ROI}(I^{m,n}_{even}, I^{m,n}_{odd,filt}) = \begin{cases} f_{SP}(I^{m,n}_{even}), & if\ Q^{m,n} < \dfrac{1}{A} \\ I^{m,n}_{even}, & otherwise \end{cases}$$

$$f_{Trans}(I^{m,n}_{even}, I^{m,n}_{odd,filt}) = \alpha \cdot f_{SP}(I^{m,n}_{even}) + (1-\alpha) \cdot I^{m,n}_{odd,filt}$$

$$\alpha = A \cdot Q^{m,n}$$

The function $f_{SP}(I^{m,n}_{even})$ determines the low-pass filtered value for that pixel using the same filters as for the spatial filtering the odd frames, based on $Q^{m,n}$.

3. ANALYSIS

The benefits in the terms of coding efficiency, redistribution of resources to the ROI and computational complexity must be considered when combining the spatial filter and the temporal filter

3.1 Coding efficiency

Assume a video sequence consisting of an I-frame followed by P-frames. The removal of details by the spatial low-pass filter will result in increased similarities between blocks both within the frame and between adjacent frames. Thus, the prediction error will decrease resulting in less information to send. Other reference blocks may be chosen for the prediction than for the original sequence as a result of the spatial filtering. The reference blocks are chosen based on length of motion vector and prediction error. When the prediction error decreases because of the filtering, a block with a smaller motion vector may be the better choice. The decrease in prediction error within the transition region is smaller, when several filters are compared to using only one for the complete non-ROI. However, the resulting reduction of the edges, created by the spatial filter at the ROI border, gives a small improvement in coding efficiency For the temporal filtering, we assume approximately uniform motion; the same number of motion vectors in a frame as in the unfiltered case; and the same or smaller total prediction error of two adjacent frames. Under these assumptions the analysis in [8] indicates that the coding efficiency is improved mainly by utilizing fewer bits for the motion vectors. Some bits are also saved because information on used prediction mode is not transmitted when skipping a block. Bilinearly interpolated blocks are not skipped by the codec, but choice of motion vectors and the decrease in prediction error gives a improved coding efficiency compared to the original sequence.

Hence, the temporal filter saves resources in coding of motion vectors, and the spatial filter in coding of the prediction error and when choosing motion vectors.

3.2 Re-allocation of resources to the ROI

The temporal filter releases only resources from the background. If the temporal filtered sequence is encoded using a target bit rate, the released bits is used to decrease prediction error independent of the errors' location within the frame. The prediction error in the background may however be decreased as a result of removing details by combining the temporal and the spatial filter. The encoder then reallocates the released bits to the most likely larger prediction errors within the ROI instead.

3.3 Computational complexity

The quality map Q is computed for all frames independent of the investigated temporal or spatial filters. The decision, based on the value of Q for a particular pixel, also has the same computational complexity for all considered filters. Thus, the complexity of computing Q is exempted from the comparison.

3.3.1 Spatial filter

The spatial filter has a computational complexity of S^2 for each pixel, where $S \times S$ is the size of the filter kernel. By using the separability property of the two dimensional Gaussian filters it is possible to reduce the computational complexity to $2S$ per pixel by filtering with two separate one-dimensional Gaussians. This results in a computational complexity of $2SN_B$ per frame, where N_B is the total number of pixels in the background.

3.3.2 Temporal filter

Solely the bilinear interpolation of the transition region needs consideration when determining the computational complexity of the temporal filter. The bilinear interpolation costs two multiplications for each considered pixel. Thus, the complexity of the temporal filter is $2N_T$, where N_T is the number of pixels in the transition area. No calculations are necessary to determine filtered background pixels.

3.3.3 Spatio-temporal filter

The unmodified spatial filter is here used for odd frames, which results in a computational complexity of $2SN_B$ per odd frame. The complexity of the bilinear interpolation in even frames increases with $2S$ for each interpolated pixel, since the background filtering is applied on spatially filtered data. The computational complexity of each even frame is $2(S+1)N_T$ based on the reasoning in section 3.3.2. Thus, the average computational complexity per frame becomes

$$SN_B + (S+1)N_T$$

3.3.4 Comparison of filters

Only the spatial and the spatio-temporal filters are considered in the comparison, since the temporal filter needs additional processing to reallocate resources. (See section 3. 2.)

The spatio-temporal filter gives a lower computational complexity than the spatial filter if

$$SN_B + (S+1)N_T < 2SN_B$$
$$\Rightarrow \frac{N_T}{N_B} < \frac{S}{S+1}$$

Hence, the spatio-temporal filtering has a lower computational complexity than the spatial filter if the transition region from ROI to background is less than 75 % of the background, since $S \geq 3$. The percentage increases for larger S.Thus as long as the ROI detection gives a limited number of smaller misdetections, the transition region will only occupy a small part of the background. This implies that the computational complexity of the combined spatio-temporal filter is much smaller than of the spatial filter alone. The additional method to reduce computational complexity using intensity variance in [6] would reduce the difference in computational complexity between the spatial and the spatio-temporal filters. However, it can be assumed that the spatio-temporal filter would still have a lower computational complexity than the spatial filter.

4. EXPERIMENTAL RESULTS

The QCIF sequences *carphone*, *foreman*, *closeup* and *outdoor* for 10 fps and 15 fps were used in the tests. The sequences *closeup* and *outdoor* were created by the authors of the paper and consist of different sizes of ROI and panning backgrounds. The parametric model presented in [9] with experimentally determined thresholds at 30 % (carphone), 32 % (foreman, closeup) and 29 % (outdoor) gave a binary detection map for each frame. This was used as a base for the quality map Q. Using the parameter A = 3 (describes the position of the ROI border as defined in [6]) the average sizes of the ROI are 32 % (carphone), 25 % (foreman), 49 % (closeup) and 12 % (outdoor) in percent of the frame. The transition region is smaller than 75 % for the investigated sequences. Hence, there is a gain in computational complexity using the spatio-temporal filter. The JM 10.1 H.264 codec for the High Profile [10] was used to encode the filtered sequences.

The spatial filter was first tested separately using 9 filters of size 5x5. The same setup as in [6] was used, apart from excluding complexity reduction and using H.264 instead of MPEG-2 for compression. When testing the temporal filter, blocks of size 8x8 were used. The results of the spatial filter and the proposed spatio-temporal filter were compared to the results in [8], in the performance analysis of the temporal filter.

As performance measures, we used bit rate of the encoded sequence and the average PSNR of the image intensity, or luminance:

$$PSNR = \frac{1}{M}\sum_{j=1}^{M} 10\log_{10}\frac{255^2}{\sigma_{e,j}^2}$$

where $\sigma_{e,j}^2$ is the error variance of the j:th frame for blocks within the ROI. In the tests either a target bit rate was used, or the quantization parameter $Qp = 28$ was kept fixed. The latter determines the precision when quantizing the prediction error.

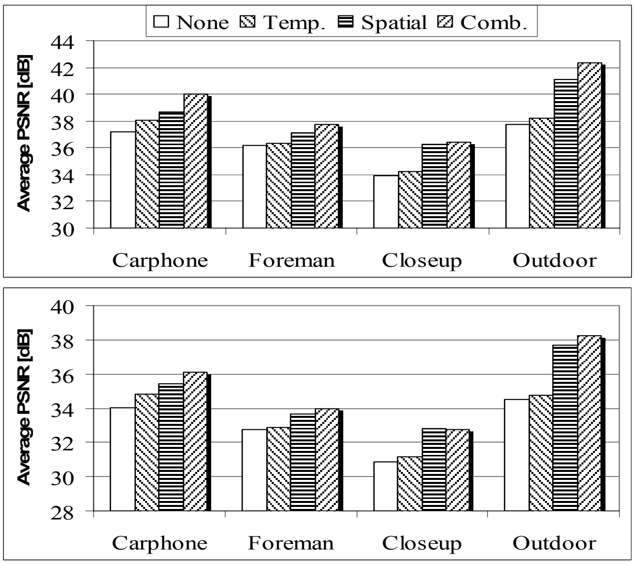

Figure 2 – Average PSNR (dB) for 10 fps of the ROI at bit rates of 64 kbps (top) and 32 kbps (bottom).

The spatio-temporal (combined) filter gave an 1.58 – 4.61 dB increase in PSNR of the ROI with respect to using no filter, for the target bit rate of 64 kbps (See fig. 2.) This increase is more than 60 % larger than the increase achieved by only using the temporal filter. A fixed quantization parameter instead results in a decrease in bit rate of 30% - 58% (See fig. 3). This is more than twice the reduction in bit rate achieved by the temporal filter alone. The improvement in PSNR of the ROI for target bit rate of 32 kbps is 1.19 – 3.70 dB. Thus the spatio-temporal filter gave an improvement within the ROI for all tested target bit rates.

Compared to the spatial filter, the combined filter improved the PSNR of the ROI by 23 % - 46 % and 13 % - 38 % for 64 kbps and 32 kps, respectively (See fig. 2). The exception is the closeup sequence, which only gave an improvement of 4 % for 64 kbps and a decrease of 3 % for 32 kbps. In most cases, the average improvement in PSNR of the ROI by the combined filter was larger than the sum of the average improvements for the spatial and temporal filters by themselves. In figure 4 (carphone) and figure 5 (foreman) it can be seen that this is also true for most frames in the sequence. Considering fixed quantization parameters, the spatial filter decreased the bit rate by 30 % - 56 % by itself. This was further decreased by the combined filter by 10% (carphone), 0% (foreman) and 6% (outdoor). The closeup sequence gave an increase in bit rate of 4 %.

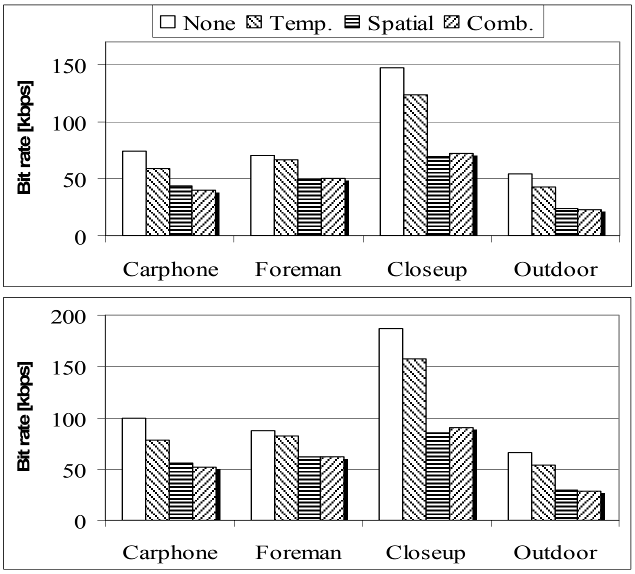

Figure 3 – Bit rate (kbps) for Qp = 28 at frame rates of 10 fps (top) and 15 fps (bottom).

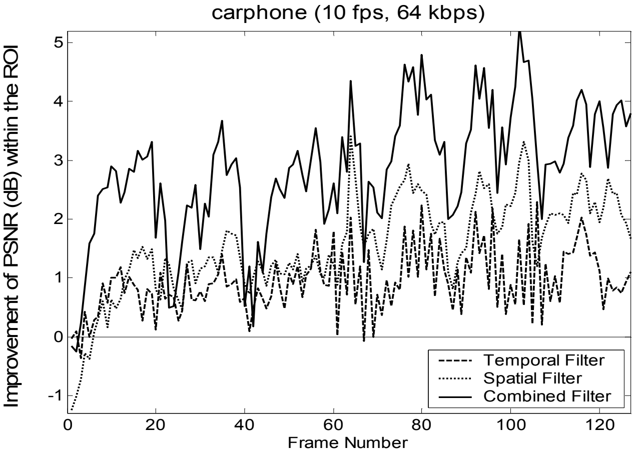

Figure 4 – The improvement of PSNR of the ROI for each frame of carphone compared using no filters.

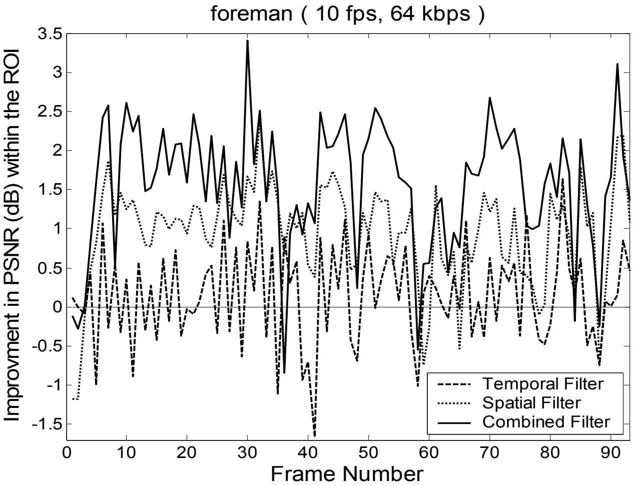

Figure 5 – The improvement of PSNR of the ROI for each frame of foreman compared to using no filters.

The improvement when applying the temporal filter to a spatial filtered sequence (i.e the combined filter) was substantially smaller compared to when the original sequence was temporally filtered. This is likely explained by the fact that the spatial filter decreases the average length of the background motion vectors. Low average lengths of the background motion vectors reduces the improvement in bitrate of the temporal filter.

The main part of the motion vectors remains unchanged in the foreman sequence and especially in the closeup sequence, where most of the signification motion is concentrated to the ROI,. Thus, the adaptive context-based probabilities in CABAC will not adapt as well to the new motion vector lengths of the background. In these cases, the bit rate is not likely to be decreased by using the combined filter compared to the spatial filter. This also indicates that a large ROI might actually increase the bit rate. The lower computational complexity of the combined filter compared to the spatial filter may motivate the usage of the combined filter even with a small increase in bit rate.

An improved reallocation of bits from background to ROI is achieved by using the rate-distortion option and a target bit rate, rather than using a fixed quantization parameter.

In some cases, artefacts occur due to the movement of the ROI border. This arises when the identified ROI contains parts of the background to ensure coverage of the complete interesting region. If the spatial filter causes border artefact they are increased by the spatio-temporal filter.

Examples from a frame in the compressed and filtered carphone sequence at bit rate 32 kbps can be found in fig. 6 for the different filters.

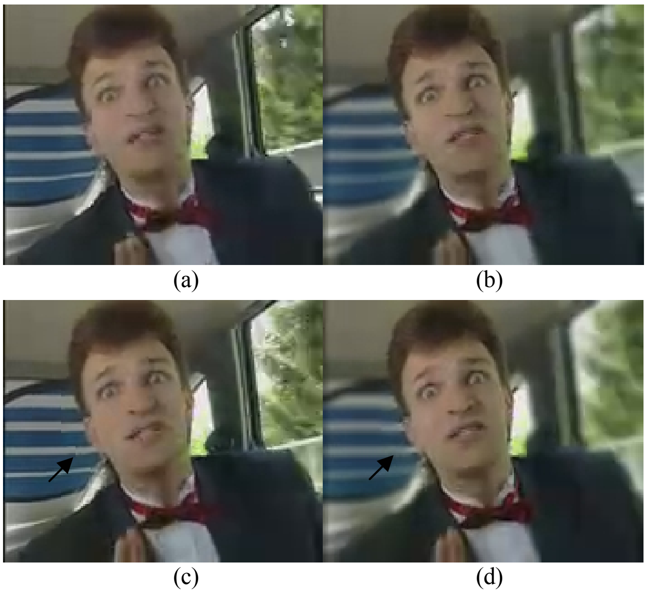
(a) (b) (c) (d)

Figure 6 – Frame 64 of the compressed carphone sequence for 10 fps and 32 kbps with (a) no prefiltering, (b) spatial filter only, (c) temporal filter only and (d) spatio-temporal filter. The visual quality within the facial region is improved by using the spatio-temporal filter. In the temporally filtered cases (c,d) the trees outside the car window are positioned as in the previous frame. At the ROI border (see arrows) still some artefacts in c) due to the large movement of the head in, but these are reduced in (d).

5. CONCLUSIONS

Two approaches to reallocate resources to the ROI by prefiltering are combined into one spatio-temporal filter. This combination gives an improvement in coding efficiency by both reducing the resources necessary for the prediction error and the motion vectors of the background.

The spatial filter performs the reallocation of these resources to the ROI, while the temporal filter ensures that the computational complexity of the spatio-temporal filter is lower than for the spatial filter by itself. The spatio-temporal filter gives an improvement in average PSNR of the ROI of 1.58 – 4.61 dB compared to no filters, which in most cases corresponds to the combined improvement in PSNR for the spatial and temporal filters by themselves.

6. ACKNOWLEDGEMENT

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

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, Aug 2003, pp. 724-730

[2] S. Sengupta, S. K. Gupta and J. M. Hannah, "Perceptually Motivated Bit-Allocation for H.264 Encoded Video Sequences, ", *IEEE ICIP*, vol.2, Sept. 2003, pp. III - 797-800

[3] 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, June 2003, pp. 488-502

[4] 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* , 2004, pp. 2067-2070

[5] J. Meessen, C. Parisot, X. Desurmont and J.-F. Delaigle, "Scene Analysis for Reducing Motion JPEG 2000 Video Surveillance Delivery Bandwidth and Complexity*,* " *IEEE ICI*P, vol. 1, Sept. 2005, pp. 577-580

[6] L. S. Karlsson and M. Sjöström, " Improved ROI Video Coding using Variable Gaussian Pre-Filters and Variance in Intensity, ", *IEEE ICIP*, vol. 2, Sept 2005, pp. 313-316

[7] Laurent Itti, "Automatic Foveation for Video Compression Using a Neurobiological Model for Visual Attention, " *IEEE Trans. Image Processing*, Vol. 13, Oct. 2004, pp. 1304-1318

[8] L. S. Karlsson, R. Olsson and M. Sjöström, "Temporal Filter with Bilinear Interpolation for ROI Video Coding", *MUCOM Technical Report*, May 2005, Avaliable at: http://www.miun.se/itm/mucom

[9] 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, Nov. 2003, pp. 3203-3207

[10] H.264/AVC,JM10.1, http://iphome.hhi.de/suehring/tml/