Adapting video delivery based on motion triggered visual attention

Velibor Adzic[a], Hari Kalva*[a], Lai-Tee Cheok[b]

[a]Florida Atlantic University, 777 Glades Road, Boca Raton, FL, USA 33431;
[b]Dallas Technology Lab, Samsung Telecommunications America, 1301 East Lookout Drive, Richardson, TX, USA 75082

ABSTRACT

Cues from human visual system (HVS) can be used for further optimization of compression in modern hybrid video coding platforms. We present work that explores and exploits motion related attentional limitations. Algorithms for exploiting motion triggered attention were developed and compared with MPEG AVC/H.264 encoder with various settings for different bitrate levels. For the sequences with high motion activity our algorithm provides up to 8% bitrate savings.

Keywords: Video coding, video delivery, video compression, visual attention, video quality, EEG, motion

1. INTRODUCTION

Our visual system has architecture that is efficient in acquiring and processing the information from the outside world. Different parts of HVS are tuned for different tasks and the amount of information at different stages of processing is significantly different. Reduction of information happens all the way from our eyes to our brain. Light enters our eyes and is focused on retina. The retina spatially encodes the image to fit the limited capacity of the optic nerve. Compression is necessary because there are 100 times more photoreceptor cells than ganglion cells that form the optic nerve carrying the information to the brain. Further reduction and compression of information happens after information is processed by visual cortex. Megabits of information that encompasses the viewed scene are crunched into couple dozen of bits that are used to represent the scene in our consciousness [1, 2].

Although some aspects of the information reduction are addressed by modern hybrid coding systems it is our opinion that there is more room for improvement. One of the primary goals in perceptually driven coding optimization is spending of available bits only on the parts of video sequence that are attended to and later processed by viewers' brain. In order to achieve this goal, first step is to determine regions in each frame of the video a viewer actually looks at and how much of the information in the video sequence is unattainable due to temporal limitations of the HVS.

2. RELATED WORK

A substantial body of previous work is present in the area of identifying attended areas of the image. Maybe the most developed is a model of building so-called "saliency" map that can be extracted in various ways. While there are well developed techniques for still image processing, moving picture domain is still not covered in great detail. Previous research includes work from Itti et al. [3,4], Bovik et al. [5] and Rensink [6] among others. However, in most of the cases in order to extract attention map, expensive computational methods are used which render these techniques impractical for real time scenarios and implementation on low power devices.

More efficient models for optimization use the information from the compressed domain [7]. This way, the comprehensive information that is already present after first encoding – frequency domain coefficients and motion vectors, provides parameters for further analysis. However, in [7] authors used edge detection on top of compressed domain information, which adds complexity.

*hkalva@fau.edu; phone 1 561 297-0511; fax 1 561 297-2800; mlab.fau.edu

Applications of Digital Image Processing XXXV, edited by Andrew G. Tescher, Proc. of SPIE
Vol. 8499, 84991L · © 2012 SPIE · CCC code: 0277-786/12/$18 · doi: 10.1117/12.930527

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 08/04/2013 Terms of Use: http://spiedl.org/terms

3. MOTION SUPPRESSION MODEL

Since our ultimate goal is to discard all the content of the original source sequence that is not attended to, we are seeking models that relate content data to appropriate characteristic of HVS that can be exploited. Our experiments are focused on implementation of the visual saccadic suppression phenomenon [8]. A fast eye movement in pursuit of a target is called saccade.

Because saccade is a motion that is optimized for speed, there is inevitable blurring of the image on the retina, as the retina is sweeping the visual field. Since blurred image is not useful for our cognitive processes, our brain practically replaces all intermediate "images" with the mask – the next perceived sharp scene. Humans are effectively blind during a saccade. The effects are similar to temporal masking that occurs at scene changes. However, detection of scene changes is somewhat easier task. The challenge here is to identify parts of scenes where there is significant movement of pursued objects we can discard much of the information in these scenes, because it is not going to be perceived.

Furthermore, we seek methods that are computationally feasible and hence are "recycling" data from compressed domain. Motion can be interpreted on a frame level or from more localized regions in the frame. We focused on frame-level quantization decisions based on motion activity information extracted from encoded sequences as a logical starting point. We used first pass stage of encoding to gather data and then used an algorithm to modify QP values that were applied in second pass to match developed model. On top of that we extracted motion vector (MV) information that we parsed and passed to an algorithm that made decisions on motion thresholds.

For saccadic suppression detection in the frames we used thresholds of 48 to 60 degrees per second that are suggested by Mizukoshi et al. [9] to be the limit over which eye starts to lag behind the speed of the optokinetic stimulus. To detect frames with motion over this threshold we calculate angular velocity in the visual field using following formula for visual angle:

$$V = 2\ arctan\ \frac{S}{2D}\ . \tag{1}$$

Here, V is visual angle, S is size of the monitor display and D is distance between display and observer. For the size S we used the diagonal measure of the monitor display. The calculations for angular velocity are based on the particular monitor size and viewer distance from display. Formula (1) with different parameters should be used for any experimental setup.

Motion vector lengths are calculated using following formula:

$$MV_L = \sqrt{MV_x^2 + MV_y^2}\ . \tag{2}$$

All frames that contained more than 60% of macroblocks with motion vectors above threshold of 48 deg/s are marked as motion masked frames. The figure of 60% is used as rough estimate for frame area dominated by motion. Although this is not supported by any scientific study we think it is reasonable estimate. We suggest further investigation that would take into account content of the video sequence. This percentage might as well be much lower depending on the content and allow for more frames to be identified as motion-masked and hence further savings in bitrate.

Final restriction that was imposed on lower threshold frames was that motion vectors had to have dominant orientation. Vector angles were sorted into 8 bins representing 45 degree partition of the 360 angle. All bins were populated comparing ratio MVy / MVx to $\tan(k \times \pi/8)$ for $k = \{\pm1, \pm3, \pm5, \pm7\}$.

In general, values for ΔQP can be calculated based on target bitrate reduction depending on the application scenario. If the maximal gain is required with perceptually lossless optimization, all parameters should be taken into account. In our experiments we used naïve approach, which can be regarded as somewhat aggressive. Our ΔQP values ranged from 8 to 12 steps since we calculated it using formula:

$$\Delta QP = \frac{QP_{MAX} - QP_i}{k}\ . \tag{3}$$

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 08/04/2013 Terms of Use: http://spiedl.org/terms

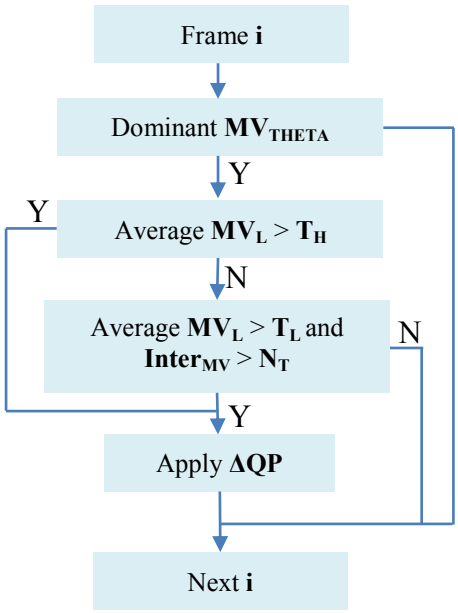

Figure 1. Flow diagram of the proposed motion optimization algorithm. All steps are applied on frame-by-frame basis.

Steps in the algorithm are shown in Figure 1. MV_{THETA} denotes dominant bin for orientation angle, T_H and T_L are high and low thresholds calculated for 60 and 48 degrees/s motion respectively, $Inter_{MV}$ is number of inter-coded macro-blocks in current frame and N_T is threshold for minimum number of such macro-blocks (60% in our case).

QP_{MAX} is maximal QP step allowed in the encoder (for X264 it is 51), QP_i is original QP step of the current frame (calculated by encoder) and k is coefficient that determines how much bitrate we want to save. It is inversely proportional to saved bitrate and has minimum value of 1. We used k = 2 which is in the middle of the possible range, but for the optimal results k should be calculated using compressed domain parameters for a specific scenario.

4. EXPERIMENTS AND RESULTS

Experimental setup was done according to ITU-R BT.500-11 recommendation with some reasonable modifications. Experiments are conducted in our lab, on a PC machines with 20'' LCD monitors. Participants were lab members (5 of them) with normal or corrected to normal vision. Video clips dataset is composed using DVD source movies. All clips are 30 seconds long and are extracted from random parts of the original movie sequences. Dataset contained 10 YUV sequences with 750 frames each in standard definition resolution (720x480p). Source movies were "Iron Man", "Star Trek" and "Faster". We selected the movies with significant amount of motion activity, because large number of motion vectors with high magnitude is required because of the threshold. Movies that don't have rapid movement in the motion sequences are not going to yield any savings after algorithm is applied.

As a benchmark encoder we used open source H.264 encoder "X264". This encoder is widely used and regarded as one of the best H.264 encoder implementations available today. Also, X264 allows explicit frame-level QP change via external list file. First, we encoded original YUV sequence to H.264 bitstream using encoder's default settings. Then we parsed encoder's log file in order to obtain QP values used in the first pass. After this we update the QP using the proposed algorithm and round QP to integer values (because QP list file can only contain integer values). Parser also collects all motion vector and macroblock data. Using this data, algorithm detects frames that are above the set motion threshold and calculates ΔQP increase that is applied in final pass. Resulting H.264 bitstream is optimized.

The requirement for the final sequence is to be perceptually lossless. In order to identify perceptually lossless sequences we asked subjects to report any perceived distortions in modified sequences. Subjects were not informed which sequences were encoded by default parameters and which were modified. The resulting savings in bitrate were calculated only for the perceptually lossless sequences. Out of 10 sequences, 7 were identified as perceptually lossless.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 08/04/2013 Terms of Use: http://spiedl.org/terms

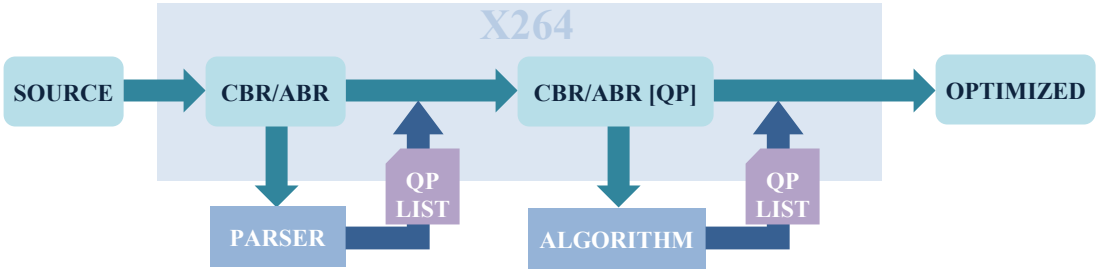

Figure 2. Experimental setup for the motion optimized encoding. Parser and algorithm are implemented outside of X264.

The reason that some sequences contained perceivable artifacts can be found in rough estimation of the parameter k. Less aggressive value of k would allow for sequences to be perceptually lossless but with smaller savings. This is why we calculated savings only for sequences that were confirmed to be perceptually lossless. We noticed that majority of sequences that were identified as perceptually lossless had motion parameters well above set thresholds. However, this is not the only factor that characterizes such sequences. Further investigation into content related parameters, such as illumination and texture would make identification of perceptually lossless sequences more precise and easier.

Table 1. Savings achieved using algorithm implementation as compared to default X264's CBR (with baseline profile) and ABR (with high profile).

	1.2 Mbps	**1.5 Mbps**	**2.1 Mbps**
CBR	4.16%	4.54%	5.24%
ABR	6.86%	7.25%	8.45%

The whole process flow is illustrated in Figure 2. Savings calculated for all 7 videos as average percentage are shown in Table 1 and Figure 3. We compared our implementation with both constant bitrate (CBR) with baseline profile and adaptive bitrate (ABR) with high profile.

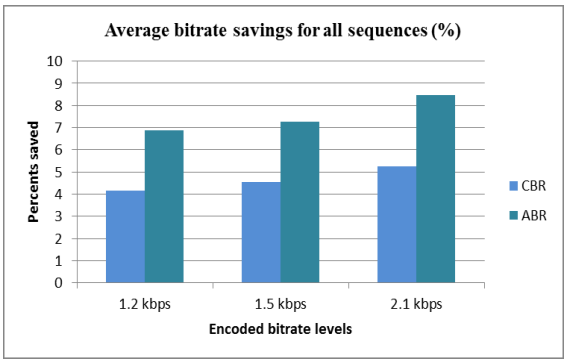

Figure 3. Bitrate savings for 3 bitrate levels compared to CBR and ABR.

We chose CBR baseline profile because it is a popular choice for adaptive streaming over HTTP, especially for mobile devices. In order to test algorithm with other encoder settings we also used adaptive bitrate (ABR) with high profile. ABR is X264's version of variable bitrate coding, done in one pass. The results are even better in this setting, because originally encoder is spending more bits on frames that contain high motion activity and there is more room for savings with our algorithm.

Figure 3 shows the savings range from couple of percent to over 8% on average for all sequences. The best results were achieved for sequence "Star Trek". The reason is that the sequence is almost whole composed of frames with very high

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 08/04/2013 Terms of Use: http://spiedl.org/terms

motion activity – more than one quarter of the frames were well above our motion threshold. Savings for this sequence are over 10% for higher bitrates. Savings are plotted in Figure 4.

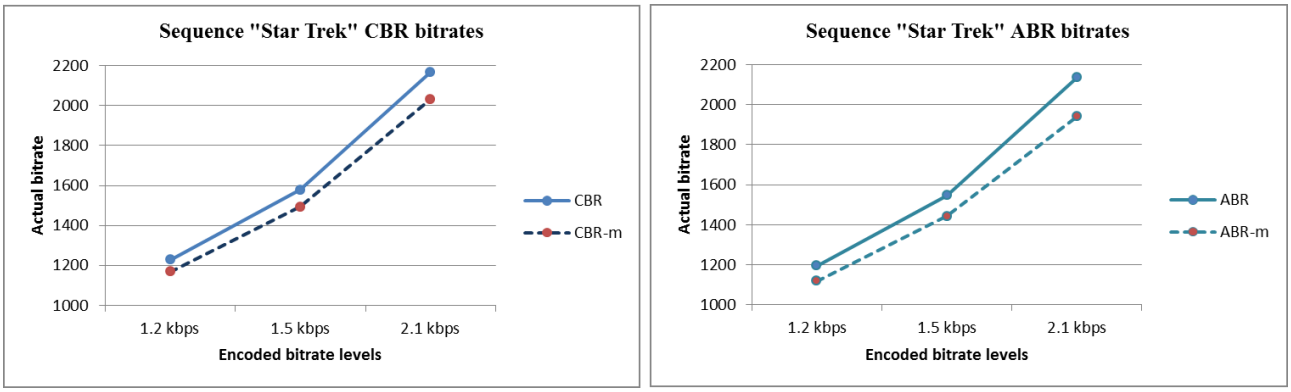

Figure 4. Savings for sequence "Star Trek" at three bitrate levels for both CBR and ABR.

Just as an illustration of the introduced artifacts that went unnoticed because of the HVS limitations we show two screenshots from sequences "Iron Man" and "Faster" in Figure 5.

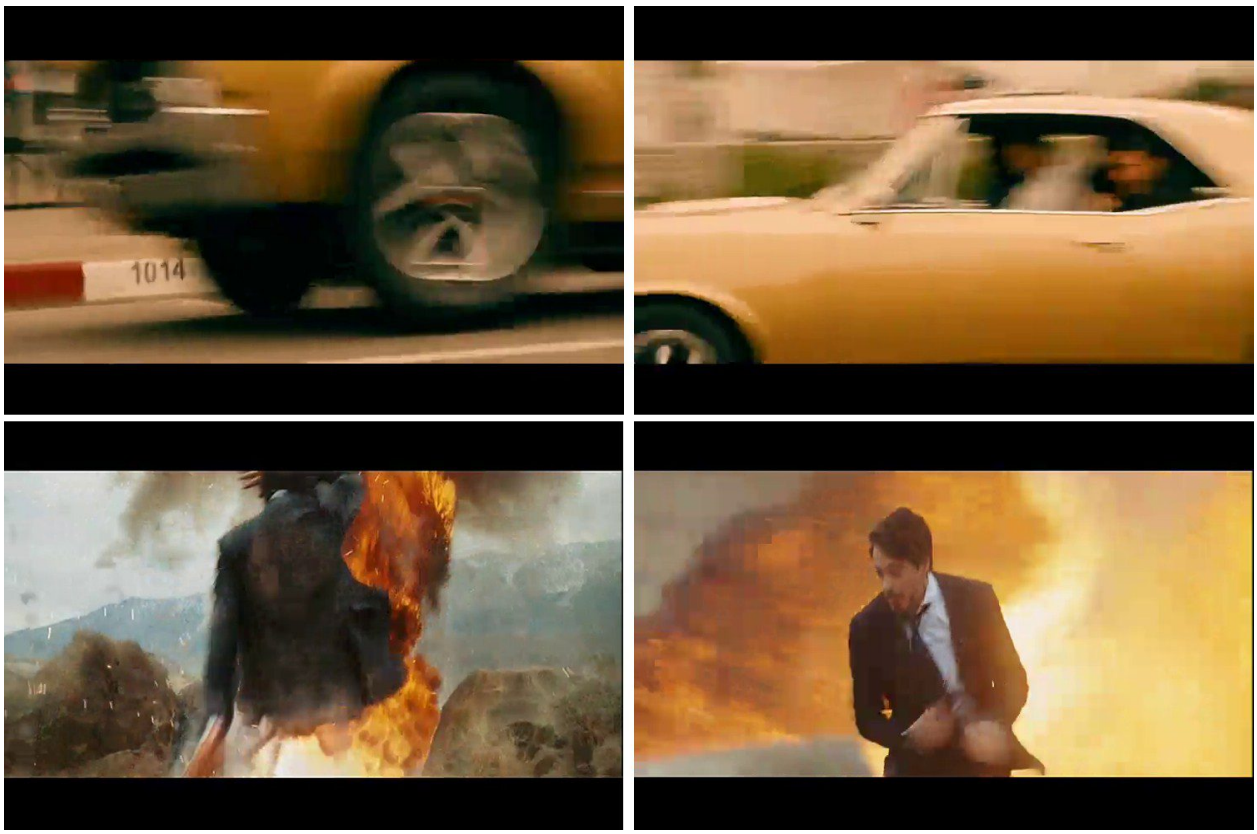

Figure 5. Frames from sequences "Faster (top row) and "Iron Man" (bottom row) that were altered with high QP increase by our algorithm, but significant artifacts were not perceived by viewers because of the rapid motion in the scenes.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 08/04/2013 Terms of Use: http://spiedl.org/terms

Clearly, by exploiting limitations of the HVS we can push the bitrate savings above limits that are currently considered as not recommended. However, this is just initial exploratory study that leaves a lot of room for improvement. Larger dataset and more detailed study should show the exact characteristics of the sequences that make them good candidates for perceptually lossless optimization. In our experiments, impairments in sequences that have motion parameters that are close to the threshold for both angular velocity and frame motion area were noticed by subjects. On top of this, contrast in luminosity between regions of interest and surroundings seems to play important role in determining potentially masked frames. Significant impairments on the edges tend to be more perceivable. For some sequences all subjects noticed impairments while for others less than half subjects reported reduced quality. Clearly, our goal is to optimize only sequences for which all subjects fail to notice impairments. Using more detailed analysis and applying impairments on the lower level of coding seems like reasonable path for improvement of current algorithm.

5. CONCLUSIONS AND FUTURE WORK

Our experiments and results show that there is room for further improvement of video coding efficiency by using cues from HVS. While the algorithm presented here works best on certain content-related subset of video sequences it still presents important first step. Savings introduced through perceptually lossless means are universal for any modern video coder. Our future work will include improvements to motion optimized model and investigation of other aspects of HVS application, such as temporal masking and further analysis of possible implementation of new findings from psychological studies such as [10]. It is our belief that potential savings introduced through combination of all phenomena can be significant.

ACKNOWLEDGEMENTS

This paper is based upon work supported by the National Science Foundation under Grant No. OISE-0730065.

REFERENCES

[1] Miller, G. A., “The magical number seven plus or minus two: Some limits on our capacity for processing information,” Psychology Review, 63: 81-97 (1956).
[2] Rensink, R. A., O. Regan, J. K., and Clark, J. J., “To see or not to see: The need for attention to perceive changes in scenes,” Psychol. Sci., vol. 8, pp. 368-373 (1997).
[3] Itti, L., “Automatic foveation for video compression using a neurobiological model of visual attention,” IEEE Trans. Image Process., vol. 13, no. 10, pp. 1304-1318 (2004).
[4] Itti, L., and Baldi, P., “A principled approach to detecting surprising events in video,” in Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, San Diego, CA, (2005).
[5] Ha, H., Park, J., Lee, S., and Bovik, A.C., “Perceptually Scalable Extension of H.264,” IEEE Transactions on Circuits and Systems for Video Technology, vol.21, no.11, pp.1667-1678 (2011).
[6] Rensink, R. A., “A model of saliency-based visual attention for rapid science analysis,” in ACM Proc. 2nd Int. Symp. Smart Graphics, New York, pp. 63-70 (2002).
[7] Tang, C.-W., Chen, C.-H., Yu, Y.-H., and Tsai, C.-J., “Visual sensitivity guided bit allocation for video coding,” IEEE Transactions on Multimedia, vol.8, no.1, pp. 11-18 (2006).
[8] Bridgeman, G., Hendry, D., and Stark, L., “Failure to detect displacement of visual world during saccadic eye movements,” Vision Research, 15, 719-722 (1975).
[9] Mizukoshi, K., Fabian, P., and Stahle, J., “Optokinetic Test Comprising Both Acceleration and Constant Velocity Stimulation (Acv-Okn Test),” Acta Oto-laryngologica, Vol. 84, No. 1-6 : Pages 155-165 (1977).
[10] Suchow, J., and Alvarez, G., “Motion Silences Awareness of Visual Change,” Current Biology volume 21 issue 2 pp.140-143 (2010).

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 08/04/2013 Terms of Use: http://spiedl.org/terms