A High Quality Adjustable Complexity Motion Estimation Algorithm For Video Encoders

Muhammad Shahid, Andreas Rossholm, Benny Lövström

International Congress on Image and Signal Processing (CISP)

2011 Shanghai

A HIGH QUALITY ADJUSTABLE COMPLEXITY MOTION ESTIMATION ALGORITHM FOR VIDEO ENCODERS

Muhammad Shahid, Andreas Rossholm and Benny Lövström
School of Engineering, Department of Electrical Engineering
SE-37179 Karlskrona, Sweden
Corresponding author email: muhammad.shahid@ieee.org

Abstract—**In the video encoding process, the motion estimation usually consumes a large part of the encoder computations. This paper presents motion estimation techniques, targeted mainly for MPEG-4 video encoding but also applicable for other video codecs e.g. H.264. A high quality adaptive algorithm with adjustable complexity, based on partially blind prediction for motion estimation, is proposed.The computational complexity of motion estimation is reduced with minor loss in the video quality. In the paper, the quality metrics PSNR, BD PSNR and PEVQ are used, and the possible trade off between complexity and visual quality is studied.**

I. Introduction

The use of video in portable electronic devices like mobile phones has seen tremendous progress recently. Inter frame predictive coding is implemented to decrease temporal redundancies in compression of video sequences and hence it makes videos compact enough to be used in such devices after transmission. This predictive coding employs motion estimation to predict motion between the video frames. Block matching motion estimation is one of the most used techniques in this area, found in video encoders like MPEG-4 [1] and H.264 [2]. However, motion estimation is quite computationally complex [3] and hence there is a need of computationally simpler methods to complete this task. Starting from the exhaustive search methods called Full Search used for motion estimation, there have been many algorithms introduced so far, which make the process faster and hence called fast search methods. Considering the limitations like power usage in a mobile device, it is believed that a sub optimal and a computationally less intensive algorithm is preferable over a more complex and optimal performance algorithm [4]. This fact advocates the need of low complexity video encoders which are fast and consume less power but offer fair level of visual quality. Full Search algorithm is not considered fit for implementation in hardware due to the huge computational complexity it requires while performing the motion estimation. To circumvent this complexity problem, many algorithms like Three Step Search, Four Step Search [5] and Diamond Search [6] have been introduced. These algorithms have a rather fixed pattern of search and have reasonable accuracy only for videos with low amount of motion contents. In real life applications, we usually have a lot of activities in videos and hence such algorithms may get trapped in local minimas [7]. UMHEX [2], EPZS [8] and PMVFAST [9] are some motion estimation algorithms which also employ predictor motion vectors while searching for best match from the search center. However, the need of a mechanism of early termination and the usage of mixed templates in accordance with motion type makes these algorithms more complex. An implementation of ”Substantially Light Motion Picture Estimator for mpeG” (SLIMPEG) has been described by Rovati et al. in [10] for MPEG-2 and later by Alfonso et al. in [3] for MPEG-4. Given that SLIMPEG [10] performs similar or better than optimal Full Search in terms of picture quality and it is being employed in industry [11], it has been used as the reference motion estimation algorithm in this paper. We suggest two techniques of partially blind prediction for motion estimation for reducing complexity in motion estimation process. An adjustable complexity motion estimation algorithm is proposed which has demonstrated high reduction in computational complexity with minor loss in visual quality. The quality metrics used to evaluate the proposed method's performance are PSNR, BD PSNR [12] and PEVQ [13], the experiments were performed on QCIF and CIF sized videos.

Rest of this paper is organized as follows. Section 2 provides a brief description of motion estimation algorithms. The proposed algorithm is described in section 3 and an introduction to the video quality metrics used is presented in section 4. The results of this work are presented in section 5. Some conclusive remarks have been drawn in section 6 along with some insight into the future work in this area.

II. Motion Estimation Algorithms

In block matching motion estimation, a video frame is usually divided into non overlapping blocks of pixels, termed as *macroblock* in literature. Some commonly used macroblock sizes are 8x8 and 16x16 but they may vary depending upon the encoding standard under consideration. The macroblocks in a video frame are searched for similarity in neighboring frame(s) to get their relative motion in the form of dimension arrays called *motion vectors*. In order to find the best match of a macroblock in the current frame by doing the similarity search in the reference frame, a lot of computations may be required. This process is simplified by limiting the search area inside a search window. Among the various algorithms developed for motion estimation, the Full Search is considered to be optimal in encoding quality and hence usually it is taken as reference for comparison with other algorithms. The Sum of Absolute Difference (SAD) is a commonly used criteria to find the similarity between the successive frames of a video in the process of motion estimation [14]. The expression used for SAD calculation for a NxN region is given by:

$$SAD = \sum_{i=1}^{N} \sum_{j=1}^{N} |c(i,j) - p(i,j)| \qquad (1)$$

where c(i,j) is the pixel value at position (i,j) in the current frame and p(i,j) denotes the same in the reference frame. Full Search does this SAD calculation on all the N points of a frame but fast search search methods do it on selected points only, hence the reason of being simpler. In real time implementation of video coders, SAD computational processing limits the performance of the system [15]. To solve this problem, SAD reuse has been a popular approach in some encoders like H.264 [16]. SLIMPEG works on the idea of reusability of refined motion vectors to avoid unnecessary SAD computations. Fig. 1 depicts the usage of previously calculated motion vectors in two steps. The algorithm starts with applying initial predictor motion vectors to the projected position of the current macroblock in the reference frame. The initial predictor motion vectors come from neighboring macroblocks and their positions are calculated dynamically, depending upon the motion content of the video sequence. These predictors are then refined to achieve more accuracy by searching around in an area, and the dimensions of this area are dictated by motion contents of the video sequence. The final motion vector points to the macroblock in the reference frame which is possibly the most similar to the macroblock under consideration in the current frame. This algorithm is 99 % less complex in computations than Full Search and its encoded picture quality has been found better than Full Search [10]. Even state-of-the-art research shows that SLIMPEG performs better than its competitors because of its sub-pixel accuracy [17]. Following the SLIMPEG implementation given in [3], the load is approximately distributed as given in the table 1. The values are obtained after the whole video encoder has been optimized, and in this case it has been done for an ARM cortex A9 with Neon support (which is a co-processor that performs vectorization). In table 1, ”SAD” refers to the computations involved in calculating the SAD value at all the search points for finding the best match and it requires 30% of the total coding workload, which is more than half of the motion estimation workload. Hence, we chose SAD computations complexity as a metric of comparison of complexity between motion estimation algorithms. The other subtasks such as *pre work* mainly consists of the process of gearing up the encoder like fetching data (frames and related information) out of the input video sequence.

TABLE I
COMPUTATIONAL WORKLOAD DISTRIBUTION FOR SLIMPEG

Task	Workload
Motion Estimation	50 %
SAD 30%, Pre work 15%, Other 5%	
Motion Compensation	6%
Write	12%
Encode	30%
Other	2%

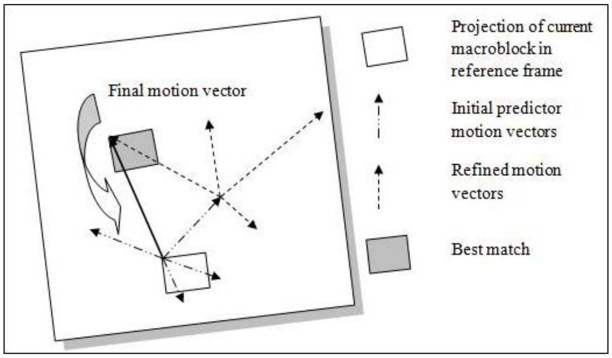
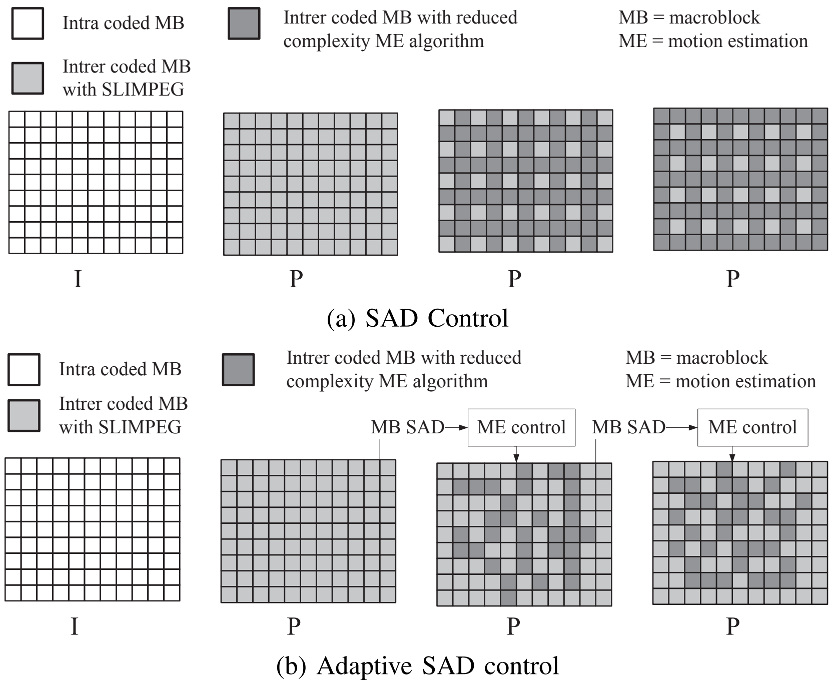

Final motion vector
Projection of current macroblock in reference frame
Initial predictor motion vectors
Refined motion vectors
Best match

Fig. 1. Two steps of using previous motion vectors in SLIMPEG

III. THE PROPOSED ALGORITHM

The relative motion of objects in a video frame is usually correlated in temporal domain. For reducing complexity, we propose to use SLIMPEG algorithm only for the parts of a video frame which have high motion content, and for low amount of motion the previously calculated motion vectors of a macroblock can be reused without applying any refinement. Hence, SAD computations required to perform refinement can be avoided. This is called here as *temporal correlation technique* and its a partially blind method of estimating motion and hence has reduced complexity.

Fig. 2 shows the schematics of the proposed algorithm. In a typical encoding scenario where after intra coding the first frame ’I’, the remaining frames are inter coded ’P’, occasional occurrences of ’I’ frames may be present later on. In inter coding frames ’P’, the approach here is to control the error (SAD) measure to keep it minimal while we apply the temporal correlation technique. We present two ways of implementing this concept. One is the SAD Control and its scheme is shown in Fig. 2a, the other is the Adaptive SAD Control as shown in Fig. 2b. As shown in Fig. 2a for SAD Control, the two gray levels of a macroblock represent two different motion estimation algorithms being used inside that particular frame. The selection of a particular algorithm for a certain macroblock position changes along the way between consecutive frames. One of them is SLIMPEG and the other is the temporal correlation technique, described earlier. The error incurred due to blind motion prediction for one macroblock in one frame is thus compensated in the next frame. SAD control technique gives considerable reduction in complexity and it has been enhanced further forming an algorithm called Adaptive SAD Control. This algorithm adaptively chooses one of the two available methods of motion estimation to minimize the error in motion estimation. It adapts with respect to the amount of motion present between successive frames of a video and this amount is estimated in terms of the previous



(a) SAD Control

(b) Adaptive SAD control

Fig. 2. Schematic of SAD control algorithms.

SAD value of the corresponding macroblock in the previous frame. In a sample execution, let us say that the algorithm is set to use SLIMPEG only for a set of macroblocks which had SAD values falling in the range of the highest n% values, in the previous frame, then the following describes our algorithm in simple wording.

- Sort the SAD values for all the macroblocks of previous frame and find the index *i* of the macroblock that has the lowest SAD value out of the top n% SAD values.
- For all the positions above index *i*, encode the corresponding macroblocks in the current frame by SLIMPEG. The rest of the macroblocks in the current frame, which had SAD values less than the n%, are encoded using temporal correlation technique.

IV. Quality Metrics

Peak signal to noise ratio (PSNR) has been a widely used metric for objective test of video quality. The encoding quality of two codecs can also be compared by using their rate distortion (RD) curves and Bjontegaard proposed a method of calculation of average PSNR differences between RD curves in [12] called BD PSNR. A curve is fitted through four data points consisting of PSNR/bitrate obtained by encoding the video sequence at four different quantization parameter (QP) values. An expression for the integral of such curve is formulated for two different codecs. The average difference is then calculated by taking difference between the integrals, divided by the integration interval. The resulting value thus obtained is called BD PSNR. An interpolation curve through four data values of a normal RD curve is obtained as following:

$$PSNR = a + b * bit + c * bit^2 + d * bit^3 \qquad (2)$$

where a, b, c and d are determined such that the curve passes through all 4 data points and *bit* means bit rate expressed in logarithmic scale.

A metric close to perceptual evaluation of video quality is PEVQ, offered by OPTICOM [13], and it has been bench marked by the Video Quality Experts Group (VQEG) to be part of ITU-T Recommendation J.247 (2008). This quality measurement algorithm can be used to analyze visible artifacts caused by a digital video encoding/decoding process and it provides picture quality of a video sequence by means of a 5-point mean opinion score (MOS).

V. Results

The proposed algorithm is tested for six video sequences with high, moderate and low motion contents. An implementation of MPEG-4 advanced simple profile encoder has been used for simulations. The first 100 frames of each of the video sequences Football @15 fps, Foreman @15fps, Clair @30fps in QCIF size and Football @15fps, Foreman @15fps, Akiyo @30fps in CIF size have been used to test visual quality outcome of the proposed motion estimation algorithm. These videos were in YUV format with 4:2:0 chrominance subsampling, and one frame from each of these videos is shown in Fig. 3. Full Search is taken as reference for calculating the BD PSNR values. The computational complexity of the proposed algorithm is compared against SLIMPEG, which is already 99% less complex than Full Search. Fig. 4 shows the trend how the decrease in computational complexity falls down to

Fig. 3. Snapshots of the video sequences used for experiments

TABLE II
Adjustability of Adaptive SAD Control Algorithm

SAD Threshold	30%			70%		
Complexity Reduction	62-65%			34-36%		
Quality metric	PSNR [dB]	BD PSNR [dB]	PEVQ Score	PSNR [dB]	BD PSNR [dB]	PEVQ Score
Football[QCIF]	28.6	-1.62	1.26	28.97	-0.41	1.27
Foreman[QCIF]	31.06	-3.03	2.97	31.67	-0.65	3.06
Clair [QCIF]	43.1	-1.13	4.24	43.37	-0.19	4.26
Football [CIF]	30.11	-2.14	2.28	30.55	-0.70	2.48
Foreman [CIF]	32.45	-3.15	3.11	32.99	-1.24	3.15
Akiyo [CIF]	42.82	-1.35	4.25	43.01	-0.31	4.25

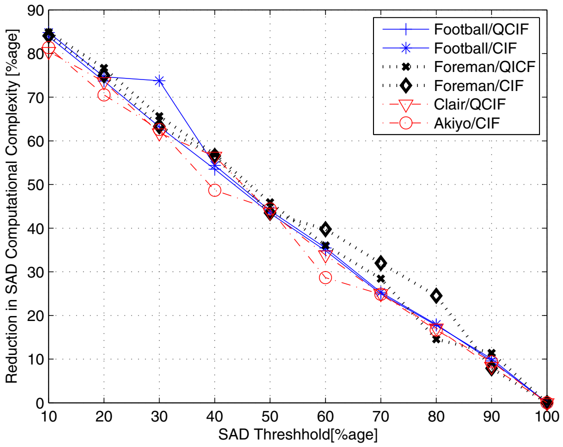


Fig. 4. Adjustability of Computational Complexity.

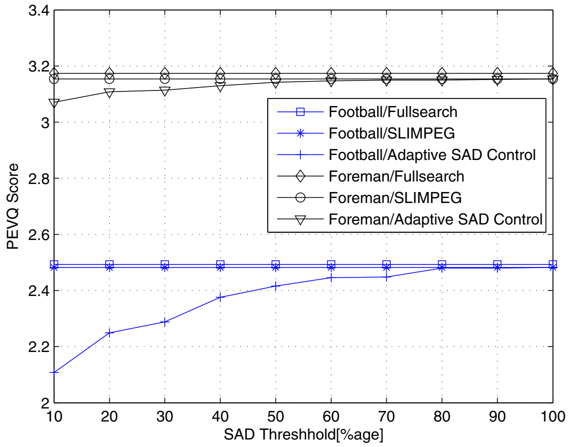


Fig. 5. PEVQ Score with Adjustable Complexity for CIF Size

zero level while the SAD threshold is increased upto 100%. The adjustability of the computational complexity with minor loss in visual quality is depicted in table II, which shows the complexity reduction together with the quality measures of all sequences for two selected SAD threshold levels. It can e.g. be seen that a complexity reduction of 62-65% can be achieved with a loss of only 1.1 dB in BD PSNR for a low motion content video like Clair. Its worth mentioning that PSNR has been widely criticized in literature [18] due to its poor correlation with perceptual quality. It operates solely on a pixel-by-pixel basis and fails to incorporate other image features like the contents. The proposed algorithm performs considerably well for the quality metric PEVQ which has quite close correlation to the human perceptual assessment [13]. Some PEVQ scores are shown in Fig. 5 for CIF size, where only the Foreman sequence shows little degradation even when maximum reduction in SAD calculations is achieved. The QCIF size videos follow almost the same trend. The PEVQ results for Clair and Akiyo sequences are not shown in the figure since they don't undergo any noticeable degradation in PEVQ when SAD threshold is reduced to its lowest level where about 84% reduction in SAD calculations is achieved.

VI. Conclusion

The proposed Adaptive SAD Control algorithm offers a system where complexity and visual quality can be traded off efficiently. Some quality is lost for high motion content video, in a controlled way, with reduction of complexity. For the two video sequences with low motion content, the PEVQ quality metric which is based on an approach to model human visual system, changes only in third decimal place even if the encoder is set to save upto 84% SAD computations. The results altogether show that the proposed algorithm has a great potential for reducing encoder complexity with minor and controllable loss in the video quality, especially regarding the perceptually close PEVQ metric. The future work should focus on conducting a subjective survey to get human scores for so processed videos.

References

[1] "MPEG-4 visual fixed draft international standard," iSO/IEC 14496-2, Oct.1998.

[2] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for h.264/avc," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 264 – 290, 2006, introduction: Special Issue on emerging H.264/AVC video coding standard.

[3] D. Alfonso, A. Artieri, A. Capra, M. Mancuso, F. Pappalardo, F. Rovati, and R. Zafalon, "Ultra low-power multimedia processor for mobile multimedia applications," in *Solid-State Circuits Conference, 2002. ESSCIRC 2002. Proceedings of the 28th European*, 2002, pp. 63 – 69.

[4] K. Namuduri and A. Ji, "Computation and performance trade-offs in motion estimation algorithms," in *Information Technology: Coding and Computing, 2001. Proceedings. International Conference on*, Apr. 2001, pp. 263 –267.

[5] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 6, no. 3, pp. 313 –317, jun. 1996.

[6] C.-W. Lam, L.-M. Po, and C. H. Cheung, "A new cross-diamond search algorithm for fast block matching motion estimation," in *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, vol. 2, dec. 2003, pp. 1262 – 1265 Vol.2.

[7] X. Wu, W. Xu, N. Zhu, and Z. Yang, "A fast motion estimation algorithm for h.264," in *Signal Acquisition and Processing, 2010. ICSAP '10. International Conference on*, feb. 2010, pp. 112 –116.

[8] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Visual Communications and Image Processing*, 2002, pp. 1069–1079.

[9] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive motion vector field adaptive search technique (pmvfast): enhancing block-based motion estimation," in *Visual Communications and Image Processing*, 2001, pp. 883–892.

[10] F. Rovati, D. Pau, E. Piccinelli, L. Pezzoni, and J. Bard, "An innovative, high quality and search window independent motion estimation algorithm and architecture for mpeg-2 encoding," in *Consumer Electronics, 2000. ICCE. 2000 Digest of Technical Papers. International Conference on*, 2000, pp. 310 –311.

[11] A. R. Bruna, A. Capra, S. Battiato, and G. Puglisi, "Digital video stabilisation in modern and next generation imaging systems," in *Proceedings of NEM Summit, Saint-Malo, France*, 2008.

[12] G. Bjontegaard, "Calculation of average psnr differences between rd curves." iTU-T SC16/Q6, 13th VCEG Meeting, Austin, Texas, USA., April 2001, Document VCEG-M33.

[13] I.-T. R. J.247, "Objective perceptual multimedia video quality measurement in the presence of a full reference," 2008.

[14] I. E. Richardson, *Video Codec Design*. John Wiley & Sons, 2002.

[15] H. Ates and Y. Altunbasak, "Sad reuse in hierarchical motion estimation for the h.264 encoder," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 2, march 2005, pp. ii/905 – ii/908 Vol. 2.

[16] A. Saha, J. Mukherjee, and S. Sural, "Approximate sad computation for real-time low power video encoders," *IET Conference Publications*, vol. 2006, no. CP522, pp. 207–212, 2006.

[17] S. Battiato, A. Bruna, and G. Puglisi, "A robust block-based image/video registration approach for mobile imaging devices," *Multimedia, IEEE Transactions on*, vol. 12, no. 7, pp. 622 –635, nov. 2010.

[18] S. Winkler, *Digital Video Quality - Vision Models and Metrics*. John Wiley and Sons, 2005.