A NOVEL CONTEXT MODELING SCHEME FOR MOTION VECTORS CONTEXT-BASED ARITHMETIC CODING

Mahmoud Ghandi
School of Electrical Engineering
Sharif University of Technology
Tehran, Iran
ghandi@ee.sharif.edu

M. Mahdi Ghandi
ESE Department
University of Essex
Colchester, UK
mahdi@essex.ac.uk

M. Bagher Shamsollahi
School of Electrical Engineering
Sharif University of Technology
Tehran, Iran
mbshams@sharif.edu

Abstract

In this paper a new context modelling scheme for arithmetic coding of Motion Vectors (MVs) is proposed. The model uses the correlation between the horizontal and vertical components of MVs to improve the probability estimation of symbols. The accurate probability estimation can therefore improve the efficiency of the context-based arithmetic coder. The proposed scheme has been adapted to the H.264 advanced video codec and the simulation results show that a considerable bit rate saving can be achieved in MV coding.

Keywords: *Video compression, context based arithmetic coding, motion vectors context modeling.*

1. INTRODUCTION

Temporal redundancy reduction is one of the fundamental principles of digital video compression algorithms. To remove temporal redundancy, motions of objects in a picture are modelled by Motion Vectors (MVs). In a typical video coding method, pictures are divided into rectangular blocks, each having a separate MV consisting of a horizontal and a vertical component. The bit rate used to represent MVs can therefore become a non-trivial portion of the bit budget. In order to remove the redundancy of the MVs, each MV is first predicted from the neighboring MVs (generating MVP). The difference between the MV and MVP (i.e. MVD) is then calculated and entropy coded.

Arithmetic coding is an efficient entropy coding method in which symbols are coded based on the probability models used by the encoder [1],[2]. The encoder asymptotically uses $-Log_2[p_i]$ bits to encode each symbol s_i, where p_i is the probability estimate of s_i determined by the encoder. If p_i is a good estimate for ρ_i (the real probability of the appearance of s_i) the total average bits per symbol used to encode the entire sequence would be:

$$-\sum p_i Log_2[p_i] \cong -\sum \rho_i Log_2[\rho_i] \qquad (1)$$

which is shown by Shannon equation to be a lower bound on the number of required bits[2]. Therefore, estimating p_i would be a critical part of an efficient arithmetic encoder to achieve a better compression.

Context-based Adaptive Binary Arithmetic Coding (CABAC) is an efficient entropy coding algorithm which is applied in the state of the art H.264 Advanced Video Codec (AVC) [3],[4],[5]. Context-modeling enables the CABAC to estimate the probability model of a symbol based on neighboring coded symbols. In adaptive arithmetic coding, probability models are adaptively modified. These characteristics make the CABAC efficiency superior to other entropy coders [4],[6].

In CABAC for coding the horizontal and vertical components of an MVD (*mvd-x* and *mvd-y* respectively), neighboring MVDs determine the context. This paper proposes a new context modeling scheme taking advantage of the inter-correlation between *mvd-x* and *mvd-y* for MV coding. This improves the probability estimation of the symbols and hence yields in bit rate saving in MV coding.

The paper is organized as follows: section 2 summarizes the MV coding method in video codecs, particularly H.264. The proposed context modelling scheme is described in section 3 followed by simulation results in section 4.

2. MOTION VECTOR CODING IN VIDEO CODECS

2.1 Motion Vector Prediction

MVs of neighboring blocks are mostly parts of a single object in the video and have strong correlations.

CCECE 2004- CCGEI 2004, Niagara Falls, May/mai 2004
0-7803-8253-6/04/$17.00 ©2004 IEEE

Therefore, in most video codecs [3],[7],[8], MVs are first predicted (MVP) typically using the three neighboring MVs as depicted in figure 1. In this method, for each component of MVP, the median of the neighboring MVs (MV1, MV2 and MV3) is selected.

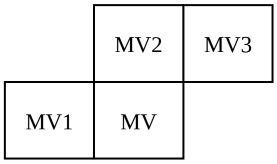


Fig. 1. Neighboring MVs for MV prediction

The distributions of MVs and MVDs absolute values (of both components) for a selected MPEG-2 sample [1] are shown in Figure 2 and 3. It can be seen that the MVDs distribution is denser at small values. Therefore, MVDs have smaller entropy than MVs and hence could be coded with smaller number of bits. In this example, the entropy for the MVs is 3.91 bits per symbol, while it is 2.89 bits per symbol for MVDs. Therefore, MVDs could be coded with an average of 1 bit per symbol less than MVs [1].

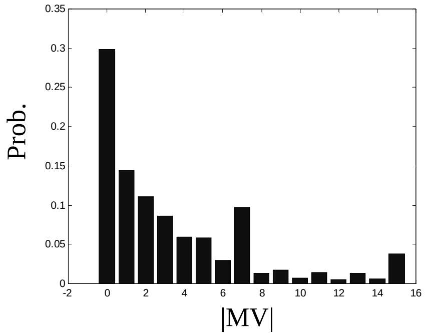


Fig. 2. MV value distribution

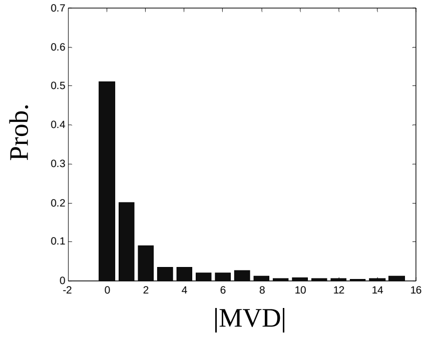
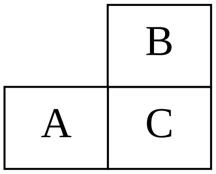


Fig. 3. MVD value distribution

2.2 MVD Probability Modeling

In H.264, the *mvd-x* and *mvd-y* are first binerized and converted to a sequence of bins as described in [4]. Each bin is then coded using CABAC. A total number of 7 different models are used for each vector component. The sum of absolute MVD values of the upper and the left blocks (Figure 4) are used as a measure to estimate the accuracy of MV prediction for each block. This value selects the probability model for the first bin among three different available models (i.e. model number 0, 1 and 2). The second bin, the third one, the rest of the bins and the sign bin are consecutively coded by separate models (model number 3, 4, 5 and 6 respectively).



$$a = |MVD\_A| + |MVD\_B|$$

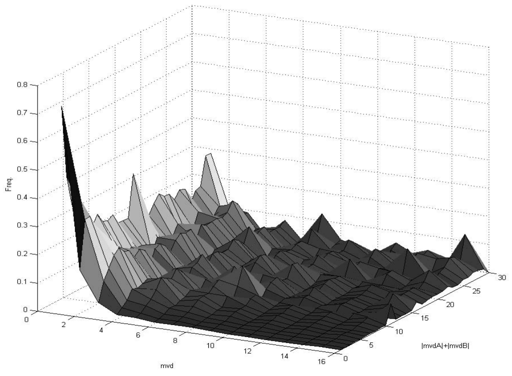
$$\text{ctx\_MVD(C)} = \begin{cases} 0, & \text{for } a < 3, \\ 1, & \text{for } a > 15, \\ 2, & \text{otherwise} \end{cases}$$

Fig. 4. Context selection rule for coding of the first bin in H.264/AVC

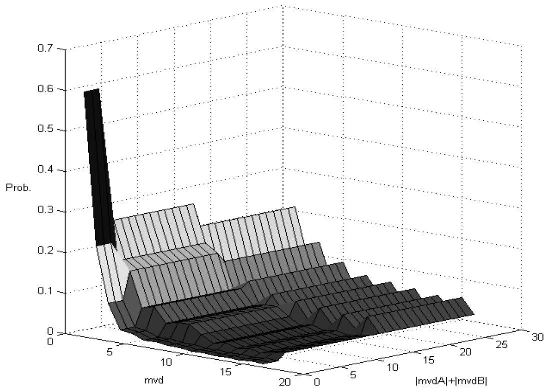
Figure 5.a shows the distribution of MVDs for different values of the neighboring MVDs of the same sample sequence used in Figures 2 and 3. It can be observed that when neighboring blocks have smaller MVDs, the distribution of the current MVD is denser (the probability of MVD = 0 is high). Figure 5.b shows the quantized form of figure 5.a. The number of probability models is limited to 3 as used in coding of the first bin of MVD in H.264.

3. USAGE OF MVD COMPONENTS CORRELATION IN CONTEXT MODELING

Figure 6 shows the distribution of the *mvd-y* value relating to each value of *mvd-x*. It can be seen that similar to figure 5.a the *mvd-y* value has a denser distribution for small values of *mvd-x*. This could be used in context modeling for coding of motion vectors. The basic assumption is that if a block and its neighbors are parts of a single object in the video, the calculated MVP from neighboring MVs would be a good prediction for MV and hence, both components of MVD would be close to zero. However, if they belong to different moving objects in the video, MVP would not be a good prediction anymore and both components of MVD are more likely to take larger values. Therefore, we can use the *mvd-x*



(a)



(b)

Fig. 5. MVD conditional distribution: a) MVD distribution for each value of |mvdA|+|mvdB|, b) MVD distribution quantized to 3 models based on the value of |mvdA|+|mvdB|

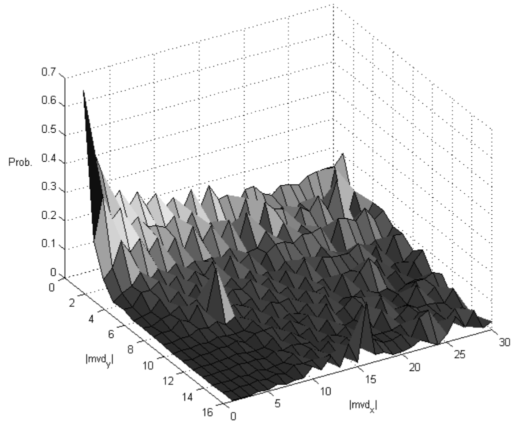


Fig. 6. MVD_y conditional distribution vs MVD_x.

value as a measure for the accuracy of motion vector prediction and as an estimate for the distribution of *mvd-y*.

In order to implement this idea and to fairly compare it with the method which is currently used in the H.264/AVC codec, we applied the same MV prediction and the same binarization schemes. We also limited the number of models to 3 for the first bin of *mvd-y*. The proper probability model among these three models is selected using Equation (2), where the value of a determines the context index by the method shown in Figure 4.

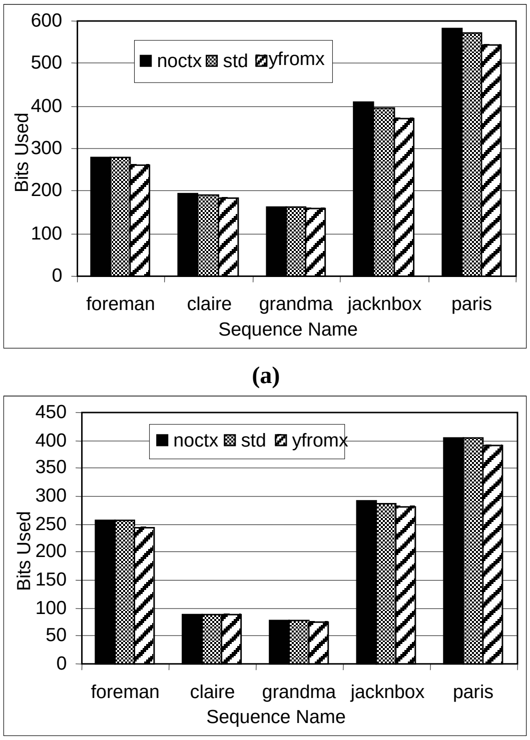
$$a=2*|mvd\text{-}x\text{-}curr| \quad (2)$$

Using the proposed method, the distribution of the *mvd-y* value can be estimated more accurately compared to the method currently used in H.264. Our model uses the information of the same block rather than neighboring blocks which may belong to different objects. Particularly in larger block sizes (such as 16x16), the neighboring MVs are spatially more apart from each other and hence have less correlation. Therefore, the proposed method is independent of the selected block size and efficiently models the MVD context in all MB division modes. Furthermore, in some situations neighboring blocks might have been coded using different types of ME or different reference frames, so they would contain less information about the accuracy of the MVP of the current block.

4. SIMULATION RESULTS

We have implemented the proposed method on JM7.5 H.264 test model software and compared the results. The simple case with no context modeling (using only 1 probability model instead of 3 for the first bin) has also been checked to estimate the efficacy of the current context modeling scheme used in H.264/AVC.

Figure 7.a shows the simulation results for P-pictures and figure 7.b shows the results for B-Pictures. In these simulations, standard QCIF test sequences are used, with frame rates equal to 10 f/s and GOP=IPBPBPB. 16x16 ME mode is used with the RD-Optimization disabled and Quantizer Parameters (QP) set to its smallest value in order to fairly compare the two methods with the same set of MVs. From the figures, it can be seen that the proposed method yields to less bit usage for coding of *mvd-y* symbols. In P-pictures more bit rate saving is achieved. This is explained by the fact that the temporal distance of a B-picture and its reference pictures are half the distance for a P-picture.



(a)

(b)

Fig. 7. Bitrate comparison for coding of MVD_y: a) in P-pictures, b) in B-pictures.

In another simulation, we evaluated the relative bit rate saving of the proposed method compared to the current method of H.264/AVC for different ME block sizes. Figure 8 shows the results for both B-pictures and P-pictures. The test sequence used in this test is foremen, QCIF, 10Hz, with GOP=IPBPB. It can be seen that the proposed method has an improved relative gain for larger block sizes. This could be explained by the fact that for larger block sizes neighboring blocks have less correlation, since they are more probable to be parts of different objects.

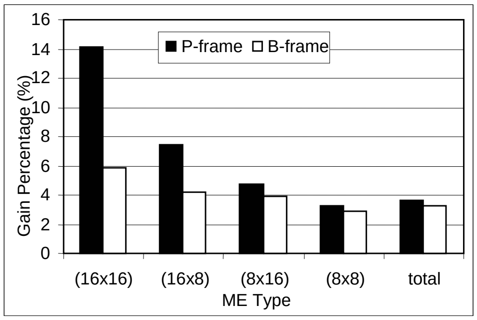


Fig. 8. Bitrate comparison for coding of MVD_y for different ME block sizes.

5. CONCLUSIONS

We have shown that in a video coding process, there is a correlation between the horizontal and vertical components of each Motion Vector (MV). This correlation is used in context modeling of MV coding to improve the coding efficiency. In the proposed method, the context modeling scheme for coding the vertical component of MV is modified based on the coded data of the horizontal component, and a considerable bit rate saving is achieved. As well as context modeling, this correlation could also be used to predict the MV vertical value from the horizontal one and to code the MV components jointly. Furthermore, a context modeling scheme that considers both current block and neighboring blocks information could be designed to achieve more bit rate saving. These aspects are the subjects of the further research.

References

[1] Ghandi Mahmoud, “Methods of Motion Vectors Coding in Video Codecs”, B.Sc. thesis (in Farsi), Sharif University of Technology, 2003.

[2] Ghanbari Mohammad, “Video Coding, an introduction to standard codecs”, IEE, 1999.

[3] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, “Final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC),” JVT-G050, 2003.

[4] Marpe D., Schwarz H., and Wiegand T., “Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard”, IEEE Trans. On Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003, pp. 620-636.

[5] Thomas Wiegand et al, “Overview of the H.264/AVC video coding standard,” IEEE Trans. on circuits and systems for video technology, Vol. 13, No. 7, pp. 560-576, July 2003.

[6] Marpe D., Blattermann G., and Wiegand T., “Video Compression Using Context-Based Adaptive Arithmetic Coding”, 13th IEEE International Conference on Image Processing(8:2001:Thessaloniki), Image processing: Proceedings. Vol.1 of 3, IEEE, 2001.

[7] International Telecommunication Union (ITU), “Recommendation H.263: video coding for low bit-rate communications,” Feb 1998.

[8] ISO/IEC MPEG, “Coding of audio-visual objects – Part 2: Visual,” ISO/IEC 14 496-2 (MPEG-4 Visual Version 1) Apr. 1999.