AN ADAPTIVE CODING SCHEME USING AFFINE MOTION MODEL FOR MPEG P-VOP

Xiaohuan Li, Joel R. Jackson*, Aggelos K. Katsaggelos** and Russell M. Mersereau**

*Center for Image and Signal Processing
Georgia Institute of Technology, Atlanta, GA U.S.A.
**Image and Video Processing Lab
Northwestern University, Evanston, IL U.S.A.

ABSTRACT

Block matching has been used for motion estimation and motion compensation in MPEG standards for years. While it has an acceptable performance in describing motion between frames, it requires quite a few bits to represent the motion vectors. In certain circumstances, the use of whole frame affine motion models would perform equally well or even better than block matching in terms of motion accuracy, while it results in the coding of only 6 parameters. In this paper, we modify an MPEG-4 codec by adding (1) 6 affine model parameters to the frame header, (2) mode selection among INTRA, SKIP, INTER-16x16, INTER-8x8, and GLOBAL-AFFINE modes by Lagrange optimal rate-distortion criteria. Simulation results demonstrate 10-20% decrease in bit-rate, compared to the MMS' codec for an average coded P-frame with the same reconstruction PSNR.

1. INTRODUCTION

Existing MPEG standards [1] have mainly used the approach of block matching for motion estimation. In many cases, the estimated motion vectors(MV's) are very similar in a neighborhood or even over the whole frame. While predictive coding of motion vectors, as specified by the MPEG standards, partially reduces the motion information's temporal redundancy by decreasing their amplitudes, it does not affect the number of motion vectors. Spatial redundancy of the motion vectors remains to be exploited.

2. AFFINE MOTION MODEL

When the motion between two frames is predominantly global, it can be reliably described by an affine model as (1). Here (x, y) is the position of the sample point in the reference frame; a_1, a_2, a_4 and a_5 for the dilation, rotation and shear components of motion; and a_3 and a_6 account for translational displacement in the two directions.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \end{bmatrix} \quad (1)$$

The advantage of an affine motion model for coding motion information is apparent: it represents motion of the whole frame by only six parameters, instead of two times the number of blocks (in the order of hundreds). Moreover, they will hopefully decrease the compensation error when global rotation, divergence and shear occur, since none of these can be accurately expressed by simpler models.

MPEG-4 introduced the concept of a Video Object Plane (VOP), defined by its shape and texture. Masked by its shape (the α-plane), the VOP is often a contoured independent object, and so is more likely to move rigidly from frame to frame. In this scenario, it makes a great amount of sense to adopt an affine motion model in place of the conventional block-wise translational motion models.

An affine motion model has been utilized with the MPEG standards, but only for coding the background sprite [1], not for an ordinary foreground video object. [2, 3] are two efforts to introduce an affine motion model for coding motion of consecutive frames, however, they both apply the affine motion model on each block (locally) instead of on the whole frame (globally), so they result in an even larger number of motion parameters, as a tradeoff against motion accuracy in cases of rotation, divergence and shear. Zhang et al. designed a multiple-level video coder with global affine motion estimation for the whole frame in order to segment the background and foreground motion objects, as well as to provide a prediction for background blocks' block-matching motion estimation in [5]. Their codec is based on a variable block size system, which is totally distinct from MPEG standard codecs. Wiegand et al. proposed a multiple frame motion estimation video coding scheme using affine model as a refinement to the BM compensated blocks in [6]. Since the affine model they use is local (the warping block is of size 22x36 pixels), they need to transmit 32 sets of affine parameters (6 numbers per set) for one QCIF frame, with enormous computational cost.

In this paper, we propose a MPEG-4-based codec featuring (1) slight modification and easy plug into an MPEG-4 standard codec; (2) modest additional computation com-

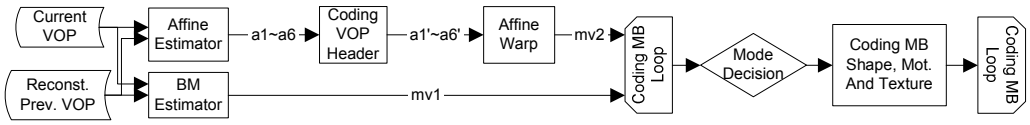

Fig. 1. MB-wise multiple motion model encoder structure.

plexity; (3) almost negligible number of bits added to the original bit-stream.

3. PROPOSED CODEC STRUCTURE

Our work is based on the software of the MoMuSys-OM-1.0-000706 coder-decoder. The adaptive coding scheme with multiple-motion models is created by modifying to the MMS system as described in the following section. Figure 2 shows a flow chart of the core parts of the modified encoder. .

3.1. Affine Motion Estimator

For each predicted-VOP(PVOP), an affine motion estimator ((A) in Figure 2) is implemented as well as the original block-matching searching algorithm ((B) in Figure 2), prior to the macroblock(MB) coding loop. The affine motion parameters are estimated by solving

$$HA = B \tag{2}$$

$$H = \begin{bmatrix} \sum xI_xxI_x & \sum yI_xxI_x & \sum I_xxI_x & \sum xI_yxI_x & \sum yI_yxI_x & \sum I_yxI_x \\ \sum xI_xyI_x & \sum yI_xyI_x & \sum I_xyI_x & \sum xI_yyI_x & \sum yI_yyI_x & \sum I_yyI_x \\ \sum xI_xI_x & \sum yI_xI_x & \sum I_xI_x & \sum xI_yI_x & \sum yI_yI_x & \sum I_yI_x \\ \sum xI_xxI_y & \sum yI_xxI_y & \sum I_xxI_y & \sum xI_yxI_y & \sum yI_yxI_y & \sum I_yxI_y \\ \sum xI_xyI_y & \sum yI_xyI_y & \sum I_xyI_y & \sum xI_yyI_y & \sum yI_yyI_y & \sum I_yyI_y \\ \sum xI_xI_y & \sum yI_xI_y & \sum I_xI_y & \sum xI_yI_y & \sum yI_yI_y & \sum I_yI_y \end{bmatrix}$$

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{bmatrix}^T$$

$$B = \begin{bmatrix} \sum -xI_xI_t & \sum -yI_xI_t & \sum -I_xI_t & \sum -xI_yI_t & \sum -yI_yI_t & \sum -I_yI_t \end{bmatrix}^T$$

Here I_x and I_y are gradients of the current frame in x and y directions, I_t is the difference between the previous reconstructed and the current frames. All sums in H are carried out over the union of the VO's (defined by the binary α-plane) in the previous and the current frames. For better accuracy, the motion estimator follows a low-pass pre-filter for noise reduction, uses a three-level hierarchical structure and runs iteratively in each level until convergence.

It outputs the six affine parameters, interpreted in terms of block-wise optical flow vectors ($\overrightarrow{mv}_2$) by (3). Note that here i and j are indexes of the MB, instead of the pixel.

$$\begin{aligned} \overrightarrow{mv}_2 &= [mvx_2, mvy_2]^T \\ &= \begin{bmatrix} a'_1 - 1 & a'_2 \\ a'_4 & a'_5 - 1 \end{bmatrix} \times \begin{bmatrix} (i-1) \times 16 + 8 \\ (j-1) \times 16 + 8 \end{bmatrix} + \begin{bmatrix} a'_3 \\ a'_6 \end{bmatrix} \end{aligned} \tag{3}$$

$a'_1 \sim a'_6$ in (3) are decoded affine parameters from the variable length codes(VLC) of $a_1 \sim a_6$,which will eventually be used by the decoder for further computation. The MB loop creates two compensated MB's for each MB to be coded – one from $\overrightarrow{mv}_1$ via block-matching; another trimmed from the affine-warped VOP, which is generated before the loop starts. The only use of $\overrightarrow{mv}_2$ is to provide prediction for shape motion estimation. They are used neither for texture compensation, nor coded to the bit-stream, as $\overrightarrow{mv}_1$ is in the MB coding loop.

3.2. Header Modification

The affine parameters $a_1 \sim a_6$ are predicatively coded by VLC to the VOP header, as part of the syntax parameters for the whole frame, also to maintain the bitstream for the content of the VOP as close to MPEG-confined as possible.

In parallel with the MPEG-4 coder counterpart, a step size of 0.5 and a search range of $[-16, +16]$ are chosen for a_3 and a_6 , which represent the shift components of the motion. The original VLC for motion vector coding(MVD) in the MPEG standards is also used for a_3 and a_6.

The rotation parameters a_1, a_2, a_4 and a_5 are more subtle and need greater accuracy. With the assumption that smaller amplitude motions occur with larger probability, we create a VLC for the magnitude of a_1, a_2, a_4 and a_5 (an extra bit is allocated for the sign) as in Table 1. For the test sequences, the magnitude of a_1, a_2, a_4 and a_5 rarely exceeds 0.1, with a substantial redundancy we set the range to be $[-0.25, +0.25]$.

The choice of accuracy is more important than the range, for it will affect the length of the VLC, and hence the coding efficiency. We experimented with accuracies uniformly separated on a logarithmic scale of $[0.0005, 0.01]$, to obtain an optimal accuracy that trades off the number of bits used for the affine parameters and the PSNR of affine warped VOP. In Figure 1, we see that as the accuracy increases in magnitude, the PSNR of the warped VOP decreases. However, in the lower range of accuracy, the number of header bits doesn't change as drastically as in the higher range, while the PSNR drops almost linearly over the whole range. Therefore we choose the 4th point 0.001 as the accuracy, which achieves the lowest number of bits for the header, with one of the highest affine compensation PSNR's.

3.3. Shape Motion Coding

Binary shape motion estimation is carried out as in the original MPEG-4 coder, in which shape motion vectors are coded to the bit stream, no matter whether the affine or the block-matching model is used. Although no affine model is utilized for the shape motion estimation, the predicted shape

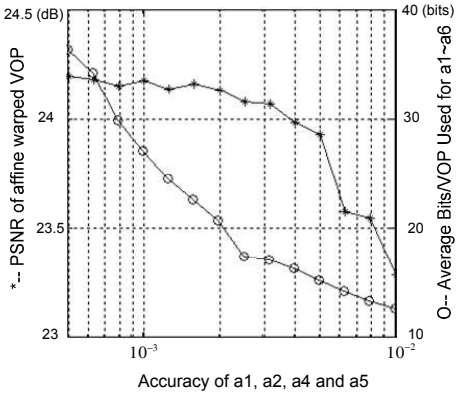

Fig. 2. Decision of accuracy of a_1, a_2, a_4 and a_5.

Variable Length Code	Magnitude of Affine Parameter
1	0
01	1
001	2
⋮	⋮
$\underbrace{000\ldots0001}$	500

Table 1. VLC table for a_1, a_2, a_4 and a_5. The magnitude is in units of accuracy, e.g. 0.0005. The last VLC contains 500 0's in a row.

motion vector is obtained from $\overrightarrow{mv}_2$, which are results from the affine motion estimation.

3.4. Block Mode Selection with Rate-Distortion Control

The choice between affine and BM motion models should take two factors into consideration: the coded bit-stream length and the reconstructed PSNR. The BM model needs some bits for motion vectors, which are totally avoided by the affine model; while the BM compensated MB might be smaller in magnitude than the affine warped MB, and thus result in fewer bits for the residual DCT, which is always the case, because of the BM motion compensation's MSE optimization target. Chances are (also proven by simulation in next section) that the affine model can beat the BM method for large error MBs, due to the absence of motion vectors. The motion mode selection between affine and BM models shares the same philosophy as the mode selection among different block sizes in MPEG-4/H.263 (INTER8x8 and INTER16x16) and H.264 (INTER16x16, INTER16x8, INTER8x16 and INTER8x8) standards: the optimal coding mode finds a best tradeoff between motion vector bit-rate and residual error bit-rate.

With the BM estimated motion vectors and the affine warp parameters, the MB modes are decided based on a Lagrangian cost function [7], which is minimized when the optimal rate-distortion combination is achieved.

$$J_{MODE}(\mathbf{S}_k, I_k|Q, \lambda_{MODE}) = D_{REC}(\mathbf{S}_k, I_k|Q) + \lambda_{MODE} R_{REC}(\mathbf{S}_k, I_k|Q) \quad (4)$$

where the MB mode I_k is varied over the set of possible MB modes $\boldsymbol{I}$={INTRA, SKIP, INTER-16x16, INTER-8x8, AFFINE}. The distortion D_{REC} is measured as the sum of square differences between the reconstructed (s') and the original frame(s)

$$D_{REC} = \sum_{(x,y)\in A} |s[x,y,t] - s'[x,y,t]|^2 \quad (5)$$

where A is the MB to be coded. The rate R_{REC} is the sum of the bits for syntax, motion vectors, residual errors and shape information. Coefficient λ_{MODE} is chosen according to [7] as

$$\lambda_{MODE} = 0.85 \times Q^2 \quad (6)$$

where Q is the quantization step size for the current frame.

3.5. Complexity Analysis

The computational load of the proposed coding scheme consists of (1)affine model estimation (the warped VOP is generated herein) and (2)calculation of affine mode cost in BM mode selection.

The temporal gradient I_t used in (2), is obtained by simple pixel-wise deduction, and costs about $6 \times 9 \times 16 \times 16 = 13824$ additions, in the case of the frame0 of bream.qcif, whose VO is defined on a 6×9 rectangle. I_x and I_y need 12 additions and 7 multiplications for filtering and deduction with a 9×9 kernel at each pixel, so the total cost for getting the spatial gradients is $12 \times 13824 \times 2$ additions and $7 \times 13824 \times 2$ multiplications. Getting xI_x, xI_y, yI_x and yI_y costs 4 multiplications and filling the elements above the diagonal of H requires another 21 multiplications at each pixel. The load for constructing H with the given gradients is 25×13824 multiplications and 21×13823 additions. Likewise, B needs 6×13824 multiplications and 6×13823 additions. Adding the 301 multiplications and 250 additions for 6×6 equation solving, and taking the 3-layer structure and an average of 3 iterations in each layer into account the overall complexity for affine motion estimation is about $2.8M$ multiplications and $2.5M$ additions.

For a half-pixel MSE BM scheme, with search range of $[-16, 16]$, the maximum case takes $(16 \times 16 + 16 \times 16 - 1) \times 64 \times 64$ additions and $16 \times 16 \times 64 \times 64$ multiplications to find the motion vector for one of the 6×9 MB's and $4 \times (8 \times 8 + 8 \times 8 - 1) \times 64 \times 64$ additions and $4 \times 8 \times 8 \times 64 \times 64$ multiplications for the MV's for the corresponding 4 blocks. This comes up to $225M$ additions and $113M$ multiplications for one frame. The computation load for the global affine motion estimation is approximately 1%

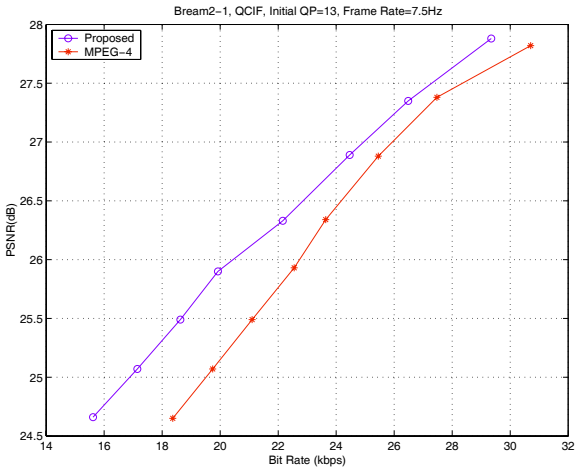

Fig. 3. Rate-PSNR curves of BREAM sequence encoded with MPEG-4 and proposed encoders.

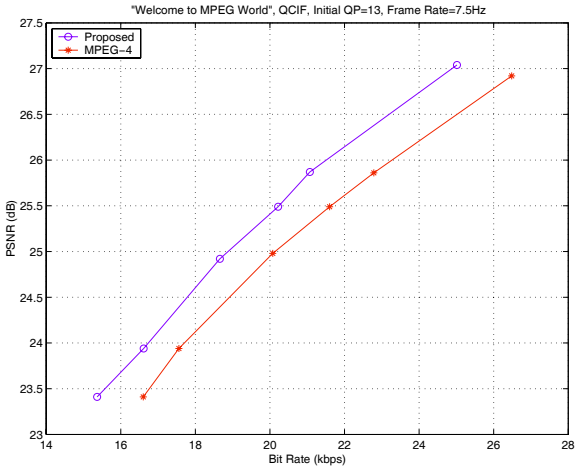

Fig. 4. Rate-PSNR curves of "WELCOME TO MPEG4 WORLD" sequence encoded with MPEG-4 and proposed encoders.

of the full search BM case, and is comparable to most of the realistic partial search cases. The computation complexity for affine mode in mode selection is of the same level as that for the BM mode.

4. SIMULATION RESULTS

Simulations are carried out on the sequences of bream.qcif and mpeg_world.qcif, under various target bit rates, with the original MMS' codec and the proposed multiple motion model code. We observe a notable decrease in the number of bits used for an average P-frame, especially when the transmission bit rate is very low. The system's relative improvement at low bit rates can be explained by the observation that affine model, in most cases, loses to the BM model in terms of compensation error, as a tradeoff of compact motion representation. Also, due to BM model's MSE optimization target, even if it doesn't give as accurate a motion vector as the affine model does, it still might well generate a smaller residual error block. When the target bit rate for video communication is low, the quantization of DCT is forced to be crude, and the subtle difference of the error block produced by the two motion models is diminished. In all cases, 50% to 70% of the inter-coded MB choose the affine mode over other modes.

5. CONCLUSION

The proposed codec system can decrease the number of bits used for an average P-frame by 10 to 20% for a range of data transmission bit rates, compared to the existing MPEG codec. The practical significance of this system is that it requires only moderate modification to the standard, to achieve its coding gain, and therefore may well be added to an original MPEG-4 codec. Due to its low bit-rate adaptability, the proposed scheme is best utilized in video conference scenarios. Although simulation of this paper is carried out on sequences with pre-defined VOP's, the alpha channel is not necessarily a pre-requisite for using the affine model. Application of the proposed coding scheme for more general sequences is under way.

6. REFERENCES

[1] Text for ISO/IEC FDIS 14496-2 Visual, ISO/IEC JTC1/ SC29/ WG11 N2502, Vancouver, Canada, Nov., 1998.

[2] N. Grammalidis, D. Beletsiotis, and M. G. Strintzis, "Sprite Generation and Coding in Multiview Image Sequences", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 10, No. 2, Mar., 2000.

[3] M.C. Lee, W.G. Chen, C. B. Lin, C. Gu, T. Markoc, S. I. Zabinsky, and R. Szeliski, "A Layered Video Object Coding System Using Sprite and Affine Motion Model", *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 130-145, Vol. 7, No. 1, Feb., 1997.

[4] H. Jozawa, K. Kamikura, A. Sagata, H. Kotera, and H. Watanabe, "Two-stage motion compensation using adaptive global MC and local affine MC", *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 75-85, Vol. 7, No. 1 , Feb. 1997.

[5] K. Zhang, M. Bober, and J. Kitter, "Image sequence coding using multiple-level segmentation and affine motion estimation", *IEEE Journal on Selected Areas in Comm.*, pp. 1704-1713, Vol. 15 No. 9, Dec. 1997.

[6] T. Wiegand, E. Steinbach, and B. Girod, "Long-Term Memory Prediction Using Affine Motion Compensation", *ICIP99*, Kobe,Japan, Oct.1999, vol.2, pp.56-60.

[7] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. J. Sullivan, "Rate-Constrained coder Control and Comparison of Video Coding Standard", *IEEE Trans. Circuit Syst. Video Technol.*, vol. 13, pp.688-703, Jul. 2003.