ICSP'04 Proceedings

ROBUST AND FAST GLOBAL MOTION ESTIMATION FOR ARBITRARILY SHAPED VIDEO OBJECTS IN MPEG-4

Hoi-Kin Cheung, Yui-Lam Chan and Wan-Chi Siu
Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ABSTRACT

Global motion estimation is used for exploiting temporal redundancies in arbitrarily shaped video objects, which is computationally the most demanding part within the MPEG-4 standard. In this paper, we propose a robust and fast method for the estimation of global motion of video objects used in the MPEG-4. Predictors are widely used in block-based motion estimation and it is well proven that the use of good predictors can speed up the motion estimation process. The proposed algorithm makes use of the shape information (alpha-plane) of video objects to form a new predictor. From the experimental results, we conclude that the new predictor provides additional performance gains with reducing computational complexity of global motion estimation.

1. INTRODUCTION

MPEG-4 is an international standard which provides a coding scheme for arbitrarily shaped video objects (VOs) [1-3]. Each VO is composed of its temporal instances, video object planes (VOPs), which is the central concept of MPEG-4 video. A VOP can be fully described by texture variations and shape representation, as shown in Figure 1. In natural scenes, VOPs are obtained by a semiautomatic [4-5] or automatic segmentation [6], and the resulting shape information can be represented as a binary alpha-plane. The alpha-plane contains the information to identify the pixels which are inside of an object (value of alpha-plane = 1), and the pixels which are outside of the object (value of alpha-plane = 0), as depicted in Figure 1(b).

Coding of arbitrarily shaped video objects relies on the reduction of statistical redundancies in the data and on the exploitation of the human visual system limitations. As the luminance and color in dynamic images are most correlated in the direction of motion, any redundancy removal is highly dependent on motion information. For this purpose, a nonlinear prediction technique, called motion compensation (MC), has been developed and used with a remarkable success. It consists of two steps: estimating motion between successive video frames and then predicting the current frame from previously transmitted frames using the motion information [7-8]. Motion compensation is a core technology of all video compression standards developed to date, such as MPEG-1 [9] and MPEG-2 [10].

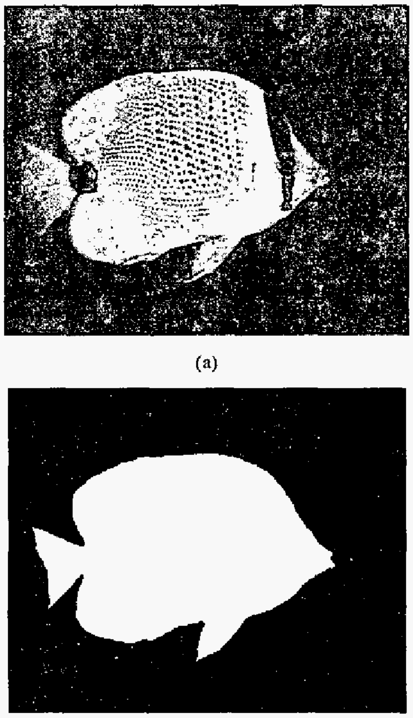

Figure 1. Representation of the VOP. (a) Image of original "Bream" VOP. (b) Binary alpha-plane of "Bream" VOP.

In general, motion in a VO results from entire motion of the VO and from displacements of individual parts composing the VO. The former is often referred to as global motion of the VO and the latter as local motion of the VO. Most motion estimation techniques ignore this aspect and make no distinction between the global and local motion; global motion is taken into account only implicitly through local estimates. For instance, in terms of motion compensation, MPEG-1 [9] and MPEG-2 [10] rely on local motion model of blocks under translation. However, it is usually advantageous to process global and

0-7803-8406-7/04/$20.00 © 2004 IEEE.

local motion separately. Not only does it result in more precise estimates, but it also leads to a more compact representation of the motion information.

Consequently, global motion estimation (GME) for an individual VO is a new coding technology for video compression in the recent MPEG-4 standard, as can be found in the MPEG-4 verification model (VM) [11]. However, techniques for global motion estimation of each foreground VO just directly employ the tools that are used in dynamic sprites generation [12]. In this paper, by considering the shape information of video objects, we propose a novel global motion estimation algorithm that is especially tailored for arbitrarily shaped VOs.

The paper is organized as follows. Section 2 gives an overview of global motion estimation. Section 3 describes the proposed algorithm. Section 4 demonstrates experimental results. Finally, section 5 provides the conclusions.

2. GLOBAL MOTION ESTIMATION

The GME technique is designed to minimize the sum of squared difference between the current VOP I and the motion-compensated previous VOP I'.

$$E=\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} w(i,j)[e(i,j)]^2 \tag{1}$$

$$\text{with } e(i,j)=I(i,j)-I'(i',j')$$

where w is the alpha-plane of the current VOP, (i,j) denotes the spatial coordinates of the pixel in the current VOP. (i',j') denotes the coordinates of the corresponding pixel in the previous VOP, $M\times N$ is the size of the bounding box that is a box surrounding the current VOP with the minimum number of macroblocks, and the summation is carried out over all corresponding pairs of pixels inside the current VOP, that is, the value of $w(i,j)$ is equal to 1.

To describe the global motion field, a 6-parameter affine motion model is typically used, which can be written as

$$\begin{aligned} i' &= m_0 i + m_1 j + m_2 \\ j' &= m_3 i + m_4 j + m_5 \end{aligned} \tag{2}$$

where $\boldsymbol{m} = [m_0, m_1, m_2, m_3, m_4, m_5]$ are the global motion parameters. Basically, m_2 and m_5 describe the VO under translation while m_0, m_1, m_3 and m_4 describe rotation and deformation of the VO. This affine model can deal with the majority of motion types encountered in video coding.

To compute model parameters, implicit in (i',j'), we use a gradient descent method to minimize E in equation (1). In the MPEG-4 VM, the Levenberg-Marquardt iterative nonlinear (LM) algorithm [13] is employed to perform the object-based minimization in order to get affine parameters $[m_0, m_1, m_2, m_3, m_4, m_5]$. Since the dependence of E on $\boldsymbol{m}$ is nonlinear, LM algorithm updates motion parameters $\boldsymbol{m}$ iteratively [13] as follows:

$$m^{(t+1)} = m^{(t)} + H^{-1}b \tag{3}$$

where $\boldsymbol{m}^{(t+1)}$ and $\boldsymbol{m}^{(t)}$ are the motion parameters at iteration $t+1$ and t, respectively, $\boldsymbol{H}$ is the Hessian matrix and $\boldsymbol{b}$ is the weighted gradient vector. More specifically, the coefficients of matrix $\boldsymbol{H}$ and vector $\boldsymbol{b}$ are given by

$$H_{kl}=\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} w(i,j)\frac{\partial e(i,j)}{\partial m_k}\frac{\partial e(i,j)}{\partial m_l} \tag{4}$$

and

$$b_k=-\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} w(i,j)e(i,j)\frac{\partial e(i,j)}{\partial m_k} \tag{5}$$

We are now to describe the flow of the LM algorithm. The motion parameters $\boldsymbol{m}_{init}$ are first initialized with the translational vector. The initialization will be discussed in Section 3. Each iteration of the LM algorithm consists of the following steps:

1. For each pixel at location (i,j) which is inside the VO, compute its corresponding position in the position (i',j') according to equation (2);
2. Compute the error $e(i,j)$ and the partial derivative of $e(i,j)$ with respect to the m_k; add the pixel's contribution to matrix $\boldsymbol{H}$ and vector $\boldsymbol{b}$ as equations (4) and (5).
3. Solve the system of equations and update the motion parameters defined in equation (3).
4. Continue iterating until the error starts to increase or N_{max} iterations are carried out.

3. ROBUST ESTIMATION OF INITIAL MOTION PARAMETERS

However, GME is a computation-intensive, time consuming task. In order to reduce the computational complexity, the estimation of initial motion parameters – the predictor is desired. This predictor is also to assure the convergence of the subsequent LM algorithm. Note that this predictor does not need to accurate. However, it must be robust enough to make the starting point of the LM algorithm lie within a convergence "basin" of the global minimum of E. For this purpose, an initial estimation of the translational components (m_2, m_5) of the displacement is computed in the MPEG-4 VM [11]. This predictor is obtained by a matching technique, which minimizes E using a n-step hierarchical search (n-SHS) [14].

However, it is well known in block-based motion estimation that n-SHS can give reasonable translational motion within a search area of ±7. But when the global translational motion of a VO is larger than ±7, n-SHS would typtically give very unreliable values resulting in many iterations in LM. The worst case performance of GME using n-SHS as initialization is poor in our experiments. And even if the global translational motion of a VO is within ±7, it is well known that the motion parameters from n-SHS can be chaotic. This gives rise to many iterations in the LM algorithm resulting in poor performance.

Our improvement over GME using n-SHS is achieved by incorporating the binary alpha-plane to accurately predict the initial motion parameters such that a good initial condition is applied in the LM algorithm to reduce the number of iterations. Since global translational motion is mostly the entire translation between two consecutive VOPs, we can approximate the global translational motion as the displacement of centroids between the current and previous alpha-planes. The coordinates of the centroid of the current alpha-plane w, (c_x, c_y), are determined using the moment, as defined in following equations:

$$c_x = \frac{\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} i \times w(i,j)}{\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} w(i,j)} \qquad (6)$$

and

$$c_y = \frac{\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} j \times w(i,j)}{\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} w(i,j)} \qquad (7)$$

Similarly, the coordinates of the centroid of the previous alpha-plane w', (c'_x, c'_y),, are then computed as

$$c'_x = \frac{\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} i \times w'(i,j)}{\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} w'(i,j)} \qquad (8)$$

and

$$c'_y = \frac{\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} j \times w'(i,j)}{\sum_{j=0}^{M-1}\sum_{i=0}^{N-1} w'(i,j)} \qquad (9)$$

Thus, the proposed initial predictor $\boldsymbol{m}_{init}$ is $[1,\ 0,\ c'_x\text{-}c_x,\ 0,\ 1,\ c'_y\text{-}c_y]$.

4. SIMULATION RESULTS

A series of computer simulations have been conducted to evaluate the performance of the proposed algorithm. Simulations have been carried out using arbitrarily shaped video objects of “Bream” and “Aki2” sequences. The mean square error (MSE) is used to compare the performance of the proposed algorithm with some related techniques. For the stopping criterion of GME, N_{max} was set to 32.

Table 1 compares the performance of GME without predictor (GME+no_Pred) versus the cases when the predictor is computed using the displacement of centroids (GME+DoC_Pred) or the conventional n-SHS (GME+nSHS_Pred). In Table 1, the performance of GME using both predictors (GME+Both_Pred) is also shown. For GME+Both_Pred, E as defined in equation (1) of both predictors are calculated and the predictor with the minimum E is chosen as the starting point of the LM algorithm.

The detailed comparisons of the average MSE and the number of iterations of the LM algorithm between GME+no_Pred, GME+nSHS_Pred and the proposed GME+DoC_Pred are tabulated in Table 1 in which the frames are temporally dropped by a factor of 0 (no dropping), 1, 2 and 3. Note that the number of iterations can reflect the computational complexity of GME since the initialization process is simple as compared with each iteration step of the LM algorithm. We show that GME+DoC_Pred outperforms GME+no_Pred and GME+nSHS_Pred. The results are more significant for the larger skipping factors. In those cases, since the motion between two consecutive VOPs is increasing, the initial predictor becomes critical for GME. Figure 2 shows that the proposed GME+DoC_Pred usually gives the minimum initial MSE for all of the frames. Thus, our proposed estimation of predictor can provide a good initial condition for the LM algorithm such that the number of iterations can be reduced and it is also robust enough to make the starting point of the LM algorithm lie within a convergence- “basin” of the global minimum of E. Besides, since GME+Both_Pred has the advantage of selecting the best predictor, it can further improves the performance.

5. CONCLUSIONS

In this paper, a robust predictor for global motion estimation of arbitrarily shaped video objects in MPEG-4 has been proposed. The binary alpha-plane is used for the adjustment of the starting point of the LM algorithm such that a number of iterations can be significantly reduced Experimental results show that our proposed algorithm can also improve the reconstruction quality of arbitrarily shaped video objects. Thus, this algorithm has the advantage of being simultaneously robust and fast.

6. ACKNOWLEDGMENTS

This work is supported by the Centre for Multimedia Signal Processing, HKPolyU (A-PD64).

7. REFERENCES

[1] ISO/IEC 14496-2, "Information tenhnology – Coding of audio-visual objects: visual", 1998.

[2] ISO/IEC, "MPEG-4 video verification model version 11.0," ISO/IEC JTC1/SC29/WG11 N2172, March 1998.

[3] T. Sikora, "The MPEG-4 video standard verification model," IEEE Trans. Circuits Systems Video Technol., vol. 7, no. 1, pp. 19-31, Feb. 1997.

[4] Roberto Castagno, Touradj Ebrahimi, and Murat Kunt, "Video segmentation based on multiple features for interactive multimedia applications," IEEE Trans. Circuits Systems Video Technol., vol. 7, no. 5, pp. 562-571, Sept. 1998.

[5] Chuang Gu and Ming-Chieh Lee, "Semiautomatic segmentation and tracking of semantic video objects," IEEE Trans. Circuits Systems Video Technol., vol. 7, no. 5, pp. 572-584, Sept. 1998.

[6] Thomas Meier and King N. Ngan, "Automatic segmentation of moving objects for video object plane generation," IEEE Trans. Circuits Systems Video Technol., vol. 7, no. 5, pp. 525-538Sept. 1998.

[7] Yui-Lam Chan and Wan-Chi Siu, "An Efficient Search Strategy for Block Motion Estimation using Image Features", IEEE Transactions on Image Processing, vol. 10, no. 8, pp. 1223-1238, August 2001.

[8] Yui-Lam Chan and Wan-Chi Siu, "New Adaptive Pixel Decimation for Block Motion Vector Estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 1, pp.113-118, February 1996.

[9] ISO/IEC 11172-2, "Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 2: Video," 1993.

[10] ISO/IEC 13818-2, "Information technology – Generic coding of moving pictures and associated audio information: Video," 1996.

[11] ISO/IEC, "MPEG-4 video verification model version 18.0," ISO/IEC JTC1/SC29/WG11 N3908, January 2001.

[12] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," IEEE Trans. on Image Processing, vol. 9, no. 3, pp. 497-501, March 2000.

[13] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing. Cambridge, U.K.: Cambridge Univ. Press, 1988.

[14] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding of video conferencing," in Proc. Nat. Telecommun. Conf., New Orleans, LA, pp. G.5.3.1-G.5.3.5, Dec. 1981.

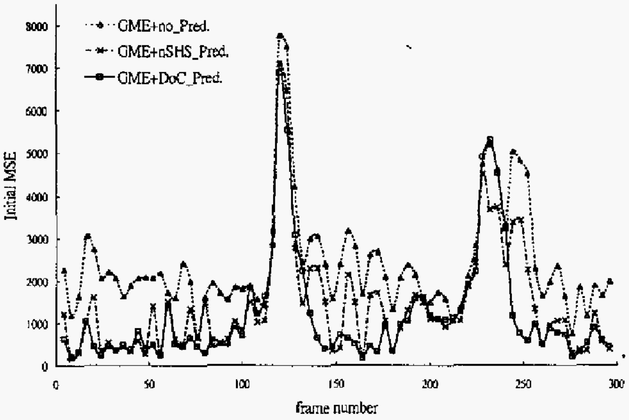

Figure 2. Initial MSE for different predictors.

Table 1. Comparison of GME without and with predictor(s).

Coding initialization predictor	GME+ no_Pred	GME+ nSHS_Pred	GME+ DoC_Pred	GME+ Both_Pred
Sequence	no. of iterations (MSE)	no. of iterations (MSE)	no. of iterations (MSE)	no. of iterations (MSE)
Skipping factor = 0				
Bream	6.43 (120.7)	4.82 (151.7)	6.22 (123.8)	4.85 (121.04)
Aki2	4.80 (67.4)	3.58 (65.2)	4.35 (66.4)	3.62 (64.9)
Skipping factor = 1				
Bream	13.3 (243.5)	11.9 (378.0)	9.17 (241.5)	8.52 (226.9)
Aki2	8.60 (134.8)	7.47 (127.9)	7.21 (128.3)	6.60 (125.7)
Skipping factor = 2				
Bream	19.2 (481.3)	15.1 (480.7)	9.96 (328.7)	10.5 (332.6)
Akis	10.4 (206.6)	10.0 (191.6)	8.9 (192.6)	9.42 (190.1)
Skipping factor = 3				
Bream	22.4 (741.4)	16.1 (546.7)	11.9 (421.2)	12.0 (385.7)
Aki2	13.8 (282.3)	13.6 (261.1)	12.3 (242.9)	12.6 (240.3)