Object-Oriented Video Coding Algorithm For Very Low Bit-Rate System

Liang-Gee Chen, You-Ming Chiu, Tzi-Dar Chiueh, Her-Ming Jong
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.

Abstract

In this paper, a video coding method based on region segmentation is proposed. It is designed for very low bit-rate systems such as video-phone and video-conferencing. Instead of block-based coding, we adopt an object-oriented approach, which first calculates the motion vector of each pixel using a modified optical flow approach, then segments the motion field into circumscribed rectangular regions that can be efficiently coded. Simulation results of several typical image sequences reveal that the proposed algorithm is both effective and of high performance.

1 Introduction

In many standard video coding systems, motion compensation (MC) with block-oriented motion estimation (ME) algorithms is widely used. However, disadvantages such as block effect degrade the subjective performance; and it's inefficient to encode at the moment of typical video-phone and video conference applications. To enhance coding efficiency, several approaches such as model-based coding[1] and analysis-synthesis coding[2] have been proposed. Nevertheless, these method require complex procedures in computation and system control. Therefore, an efficient and simpler algorithm is still lacking for the very low bit-rate application.

An algorithm proposed years ago, pel-recursive algorithm (PRA)[3], is a gradient-descent approach that minimizes $|E_x u + E_y v - E_t|$, where E indicates the luminance of picture, (u, v) is the motion vector viewed from the current frame to the previous frame, and the subscripts x and y represent partial differentiation; for example, E_x indicates $\frac{\partial E}{\partial x}$. PRA makes a very good estimation for each pixel, i.e. the residual error, or the displaced frame difference (DFD), is small.

However, because each vector is calculated locally, a densely and randomly distributed motion field is generated and it is impossible to transmit these motion vectors efficiently. Optical flow algorithm(*OFA*) [4] is similar to PRA except that the cost function to be minimized is now

$$\varepsilon = \int\int ((E_x u + E_y v - E_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2))dxdy. \quad (1)$$

The equation is composed of two terms: the former is the same as that in PRA, which is called constraint term; the latter is a smoothness term.

In this paper, an efficient video coding method based on *OFA* is proposed. By combining region segmentation, the proposed method can provide very low bit-rate coding and still preserve high subjective quality. The derivation of the algorithm from *OFA* is shown in Sec. 2. Details of the proposed algorithm is described in Sec. 3. Sec. 4 presents the simulation results and some discussions. The conclusion is made in Sec. 5.

2 Solution to Optical Flow

When the minimum of Eq. (1) is reached, $\frac{d\varepsilon}{du} = \frac{d\varepsilon}{dv} = 0$. That is, we have

$$\frac{\partial \varepsilon}{\partial u} - \frac{\partial}{\partial x}\frac{\partial \varepsilon}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial \varepsilon}{\partial u_y} = 0,$$

$$\frac{\partial \varepsilon}{\partial v} - \frac{\partial}{\partial x}\frac{\partial \varepsilon}{\partial v_x} - \frac{\partial}{\partial y}\frac{\partial \varepsilon}{\partial v_y} = 0.$$

Therefore,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{1}{\lambda}(E_x u + E_y v - E_t)E_x,$$

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = \frac{1}{\lambda}(E_x u + E_y v - E_t)E_y.$$

By replacing u_x and u_y with $u_{i,j} - u_{i-1,j}$ and u_y with $u_{i,j} - u_{i,j-1}$ respectively, we have

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) - 4u_{i,j} \\ &= 4\overline{u} - 4u.\end{aligned}$$

0-7803-2440-4/94/ $4.00

Similarly,

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = 4\overline{v} - 4v.$$

A more clear description of $\overline{u}$ and $\overline{v}$ is shown in Fig. 1.(a). Therefore, the motion vector at position (i,j) becomes

$$u_{i,j} = \overline{u} - \frac{E_x\overline{u} + E_y\overline{v} - E_t}{4\lambda + E_x^2 + E_y^2}E_x, \quad (2)$$

$$v_{i,j} = \overline{v} - \frac{E_x\overline{u} + E_y\overline{v} - E_t}{4\lambda + E_x^2 + E_y^2}E_y. \quad (3)$$

0	$\frac{1}{4}$	0
$\frac{1}{4}$	0	$\frac{1}{4}$
0	$\frac{1}{4}$	0

(a)

$\frac{-1}{4}$	$\frac{2}{4}$	$\frac{-1}{4}$
$\frac{2}{4}$	0	$\frac{2}{4}$
$\frac{-1}{4}$	$\frac{2}{4}$	$\frac{-1}{4}$

(b)

Figure 1: (a) Weights to compute $\overline{u}$ and $\overline{v}$ (b) Weights to compute $\hat{u}$ and $\hat{v}$

In order to code the motion vectors further efficiently, we have to come up with motion field with less spatial variation. Instead of using extra estimation filtering scheme[5], we proposed a modified cost function which is shown as follows:

$$\varepsilon = \int\int ((E_x u + E_y v - E_t)^2 + \alpha(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \beta(u_{xy}^2 + v_{xy}^2))dxdy. \quad (4)$$

The additional third term indicates the penalty on convoluted edges. In other word, the cost function favors motion fields with smoother edges. To minimize the cost function, the following relation must be satisfied

$$\frac{\partial \varepsilon}{\partial u} - \frac{\partial}{\partial x}\frac{\partial \varepsilon}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial \varepsilon}{\partial u_y} + \frac{\partial^2}{\partial x \partial y}\frac{\partial \varepsilon}{\partial u_{xy}} = 0.$$

By similar derivation, we have

$$4(\alpha\overline{u} + \beta\hat{u} - (\alpha+\beta)u) = \frac{1}{\lambda}(E_x u + E_y v - E_t)E_x,$$

$$4(\alpha\overline{v} + \beta\hat{v} - (\alpha+\beta)v) = \frac{1}{\lambda}(E_x u + E_y v - E_t)E_y.$$

where $\hat{u}$, $\hat{v}$ indicates as shown in Fig. 1.(b). Let

$$\alpha\overline{u} + \beta\hat{u} = (\alpha+\beta)u^*,$$

$$\alpha\overline{v} + \beta\hat{v} = (\alpha+\beta)v^*,$$

then

$$u_{i,j} = u^* - \frac{E_x u^* + E_y v^* - E_t}{4(\alpha+\beta) + E_x^2 + E_y^2}E_x, \quad (5)$$

$$v_{i,j} = v^* - \frac{E_x u^* + E_y v^* - E_t}{4(\alpha+\beta) + E_x^2 + E_y^2}E_y. \quad (6)$$

We can find Eq. (5)(6) has a similar form as that of Eq. (2)(3).

3 The Proposed Coding Algorithm

The proposed algorithm, which is called as *modified optical flow algorithm(MOFA)*, consists of two phases: the first step is to find the interframe motion flow (u, v); the other is to segment the flow and to code the information.

To obtain the motion flow, we can iteratively use the Eq. (5)(6). It should be noticed that in each iteration, the motion vector of each pixel is calculated independently and parallelly.

After the motion field is obtained, we segment these motion vectors and record them into *coding linked list (CLL)*. Each element in the linked list represents one "object" in the frame. An object is defined as a region that contains the same nonzero motion vector. To encode an object, we have to record its shape and the motion vector. At first, determine the circumscribed rectangle of this "object" by traversing its contour, as shown in Fig. 2.(a), and identify the pixels in the rectangle with the nonzero vector as "1" and the others as "0". Next, transform the bit-map into a bit-stream by a spiral-like scanning (Fig. 2.(b)) which may generate as many contiguous 1 or 0 as possible. For example, the bit-stream in Fig. 2.(a) is
0000111000010000111100111111111111011101111111111.
Therefore, the information of an object contains the location and size of the rectangle, the vector, and the bit-stream, as shown in Fig. 3. The whole encoding algorithm is listed below:

1. Clear the *CLL* and initialize all (u, v) to zero.
2. For k=1 to *number of iterations*
   for i, j in raster scanning
   find $(u_{i,j}, v_{i,j})$ by Eq. (5)(6).
3. Check the pel. If its motion vector is nonzero and if it doesn't belong to any traversed objects, then goto step 4, else check the next pel.
4. Find the circumscribed rectangle of this "object" and append the information of the object to the *CLL*.
5. Goto step 3 until all pixels in current frame are scanned.
6. Transmit the whole *CLL* for further possible lossless scheme, such as VLC and RLC.

In the proposed algorithm, residual error will not be sent. Consequently, to enhance the performance we employ an extra *stationary region test*. If the DFD with MC is larger than non-compensated difference, then the motion vector of the pixel is set to be zero. In other words, if $DFD(u, v) > DFD(0, 0)$, let $(u, v) = (0, 0)$.

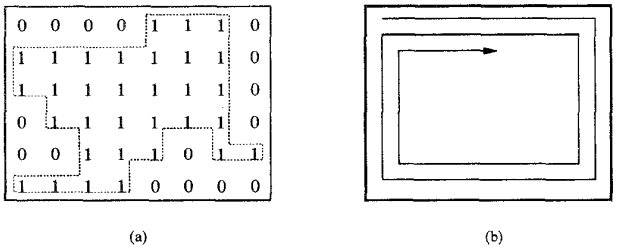

Figure 2: (a) Circumscribed rectangular region (b) Spiral-like scanning

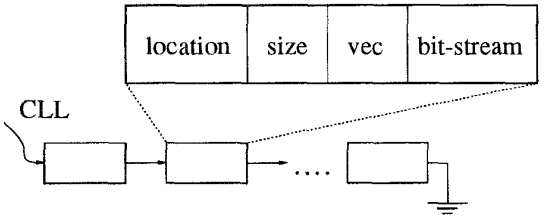

Figure 3: Data structure of coding linked list

The parameters α and β must be chosen properly. If α is too large, the motion vectors tend to group together, which degrades the performance badly. On the other hand, if α is too small, motion vectors distribute randomly and are difficult to code. The parameter β should be chosen much less than α; if it is larger than α, an oscillated motion field may be generated.

4 Simulation Result and Discussion

Similar vectors in the motion field tend to group together because of the smoothness term in Eq. (5)(6). Fig. 4 shows the amplitude of motion vectors between 2 successive frames in the sequence *claire*. We find most portion of the frame is stationary and other parts sensitive to human eyes, say, the mouth and the eyes, will be coded accurately because their vectors are calculated precisely. The picture reconstructed with our approach, therefore, is of high quality.

Performances and bit-rates of the sequence *claire* using different approaches are shown in Fig. 5 and Fig. 6, respectively. *OFA16* indicates that utilizing *OFA* with 16 iterations; in the same manner, *MOFA16* and *MOFA8* indicate utilizing *MOFA* with 16 and 8 iterations, respectively. According to the results, *MOFA16* and *OFA16* are almost the same in PSNR, while the former requires less bandwidth when the bit-rate is relatively higher. It is worth noticing that *MOFA8* degrades a little in PSNR but requires much lower bit-rate. Therefore, we choose this approach as the final proposed algorithm.

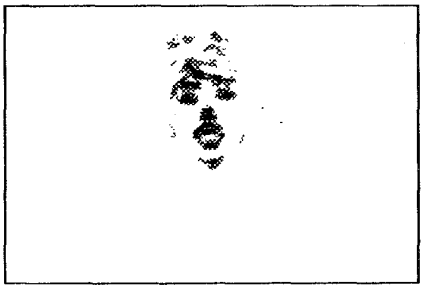

Figure 4: The amplitude of motion flow of *claire*

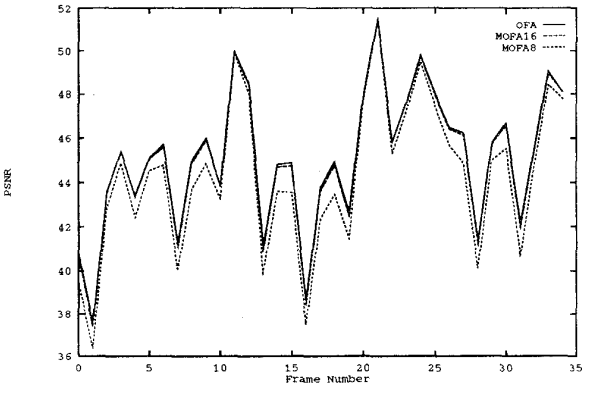

Figure 5: The performance of sequence *claire*

We list the average PSNRs of 4 sequences in Table 1 with the proposed algorithm in comparison with 16×16 *full-search block matching algorithm (FSBMA)*. We use the *MOFA* with 8 iterations, ($\alpha = 1024, \beta = 64$). From the simulation result we find that the performances of *MOFA* are better than or close to FSBMA in PSNR except for the case of ***susie***, which is a fast and global moving sequence. Nevertheless, our reconstructed images may look better than that of *FSBMA* subjectively because of the object-oriented characteristic of the proposed approach. To enhance the PSNR, more iterations are needed but it may entail a higher bit-rate.

The average bit-rates (bits per pixel) are also listed in Table 1. The results of ***miss*** and *claire* are very good, while that of ***salesman*** is moderate. However, the result for the sequence ***susie*** is not satisfactory. It is because the distribution of motion flow is sparse due to the zooming and global movement in this sequence. Furthermore, because the initial guesses are zero, and 8 iterations is not enough to find the true motion vectors. Besides, our algorithm is vulnerable for high-detailed-moving or fast-moving sequence. High-detailed-moving implies that there're so many local moving with different directions and speeds. This dense motion field needs a

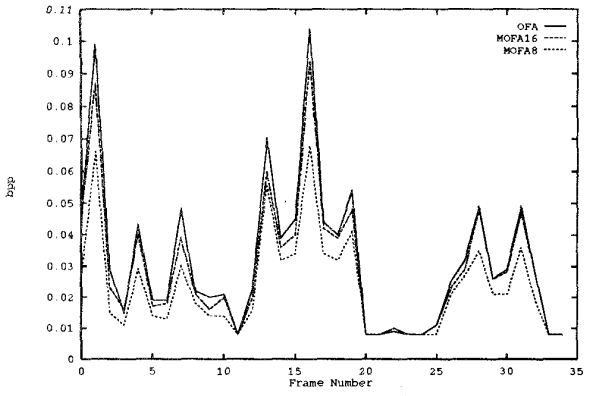

Figure 6: The bit-rate of sequence *claire*

large amount of side information due to lots of segmented regions. On the other hand, motion of fast-moving objects is difficult to calculate through limited number of iterations, and this causes a poor performance.

Sequence	susie	salesman	miss	claire
FSBMA(PSNR)	35.38	34.72	38.63	42.46
MOFA8(PSNR)	34.15	34.87	38.41	44.24
MOFA8(bpp)	0.133	0.040	0.014	0.023

Table 1: Average performance and bit-rate for different sequences

Figure 7: The reconstructed image of *susie* with PSNR= 28.5 dB

Due to the smoothness term, the motion vectors distribute much more regularly than that in *PRA*. So we can segment them into "objects" and code them efficiently. The additional penalty term results in fewer and more regular regions without degrading the performance because a little shape distortion won't be noticed by human eyes. In addition, the larger residual error occurs

Figure 8: The original image of *susie*

on the edges of the moving object, where is neglected by human eyes. Therefore, the subjective performance of the reconstructed frames using the proposed algorithm must be better than those using block-based methods even if the PSNRs are the same. Fig. 7 shows the reconstructed image of *susie* using *MOFA*. Although the PSNR is as low as 28.5 dB, it looks almost as good as the original image, as shown in Fig. 8.

5 Conclusion

Optical flow is originally applied in image analysis because it makes a precise estimation of object motion. The proposed algorithm is inspired from its feasibility of segmentation of motion vectors. Because the high quality of motion compensated pictures, there are no residual errors to transmit. To code these motion vectors efficiently, we also propose a "circumscribed rectangular" segmentation method, and the required bit-rate is quite low.

The proposed approach also provide a multifunctional system. The transmitted information can be used not only for image reconstruction but also for image analysis because it is coded with object orientation. Some traditional coding schemes such as the block-based method provide much limited information for analysis. Therefore, for different requirement, all we have to do at the transmitter is to adjust some parameters with the same architecture.

When applying the proposed algorithm to videophone of 180×120 and 10 frames per second, the bit-rate of 0.04 bpp results in a transmission rate of 8.64 Kbps, which is feasible for current telephone network. Considering wider applications in real-world systems, there are some improvements for further study. One is the adaptability: to provide a more uniform bit-rate that doesn't severely depend on the content of frames, a bit-rate control is needed; that is, the system should be able

to adjust the amount of coded information. The other is the complexity: although the proposed algorithm is much simpler than other previously reported object-oriented approaches, the computation is still too heavy for practical applications. To reduce the system cost, a simplified algorithm, which is feasible for hardware implementation while preserving the good performance, is being developed.

References

[1] Y. Nakaya, Y.C. Chuah, and H. Harashima, "Model-based/waveform hybrid coding for videophone images",*ICASSP*, Vol. 4, pp. 2741-2744, 1991.

[2] Harald Schiller, Michael Hötter, "Investigations on color coding in an object-oriented analysis-synthesis coder",*Signal Processing: Image Communication*, Vol. 5, pp. 319-326, 1993.

[3] A. Netravali, J. Robbis, "Motion compensated television coding part 1", *Bell Syst: Tech. J.*, Vol. 58, pp. 631-670, March. 1979.

[4] B. K.P. Horn, B. G. Schunck, "Determining optical flow", *Artificial Intelligence*, Vol. 17, pp. 185-203, 1981.

[5] A.V. Brandt, "Object tracking and motion estimation with a moving camera", *Proc. ASST*, 1990, Aachen, pp. 186-191.