Adaptive Pixel Difference Classification, an Efficient and Cost Effective Algorithm for Motion Estimation

H.R. POURREZA[†], M. RAHMATI[†] and F. BEHAZIN[‡]
[†] Computer Eng. Dept., Amirkabir University of Tech., Hafez Ave., Tehran
[‡] YMA College, P.O. Box 19585-774, Tehran
IRAN

Abstract: - In this paper we present a simple adaptive block-matching algorithm for video compression. Our method is an adaptive approach to pixel difference classification (PDC) matching criterion. A constant threshold is used in original PDC algorithm in which optimum threshold varies from sequence to sequence. In contrast to the original PDC algorithm, adaptive PDC (APDC) employs a time variant threshold, where the threshold is obtained from parametric classification of each frame. The proper threshold obtained for each frame is used for subsequent frame. Experimental results indicate that this method can yield further 99% of optimum performance.

Key-Words: - Block Matching, Motion Estimation, Video Compression.

1 Introduction

In video compression, the motion compensation is a traditional technique for reducing the temporal redundancy between consecutive frames of a video sequence [3]. Motion compensation mainly consists of two stages: motion estimation and prediction error coding [1]. There are various approaches for motion estimation. Block matching motion estimation technique is one of the approach that is widely used in various video compression applications such as MPEG1,2 and H.263 [3]. In the block matching method, the motion estimation is carried out on a block-by-block basis. For this purpose, the coding frame (current frame) is partitioned into non-overlapped blocks ($N \times N$ blocks). Assuming that all pixels within each block have a uniform motion, for finding corresponding motion vector of a block, we search for a block on the reference frame that has the best match to it (according to a given criterion). The search performs within a larger block; i.e. $(N+2w) \times (N+2w)$; (search area) centered at the same location on the current frame, where w denotes the maximum predicted displacement of any objects. The reference frame is defined either at past time, for forward motion estimation, or at future time, for backward motion estimation [3]. Whereas motion estimation process imposes the most hardware cost to the video encoder, several approaches have been proposed for hardware reduction by reducing the number of search points and simplification of criterion function [3,4]. The PDC[1] is one approach that targets the criterion simplification [2]. In PDC, each pair of pixels is classified as either matched or mismatched, depending on whether their difference exceeds a certain preset threshold.

Let us denote $C(x,y)$ and $R(x,y)$ as the gray-level of current and reference frame at location (x,y), respectively. The PDC criterion at (i,j) location of search area is defined by [1,2]:

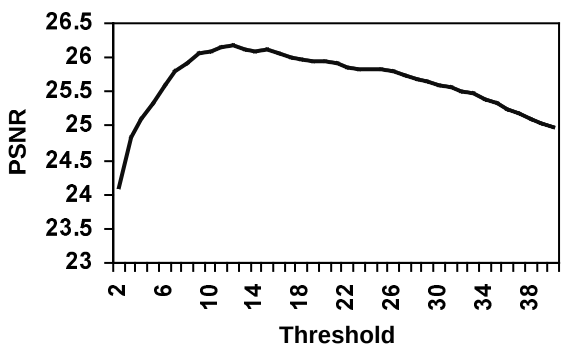

Fig. 1. PDC threshold versus prediction noise of Foreman Sequence

[1] Pixel Difference Classification

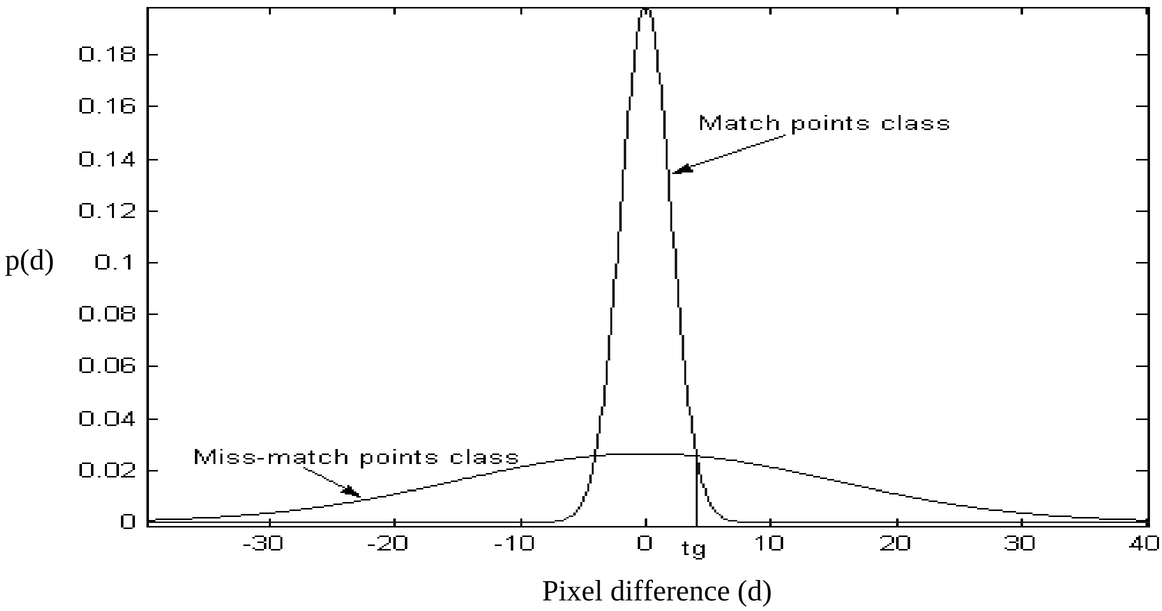

Fig. 2. Assumed distribution of pixel differences for matched and missed-match points

$$PDC(i,j) = \sum_k \sum_l G_{i,j}(k,l) \qquad (1)$$

where, $(l,k) \in$ {block area} and given a threshold t_g, the $G_{i,j}$ is defined as:

$$G_{i,j}(k,l) = \begin{cases} 1, if \ \left|C(x+k,y+l) - R(x+i+k,y+j+l)\right| \le t_g \\ 0, otherwise. \end{cases} \qquad (2)$$

Motion vector *(u , v)* is defined by the *(i , j)* values for which *PDC(i , j)* is maximized.

The PDC presents a reasonable performance at low hardware cost, but one drawback of PDC is that the optimum threshold varies from sequence to sequence [4]. Choosing proper threshold has a great impact on the effectiveness of the PDC distortion function, but the best threshold for one sequence is not necessarily optimal for another sequence. Nelson [5] tested a variety of thresholds on three sequences and found threshold of 5, 15, and 25 were optimal depending on the sequence. Chan et al. [6] applied PDC and found its performance varied depending on the type of motion in the video sequence. They suggested that smaller threshold values perform better than larger ones where the motion in the sequence is slow and that the inverse is also true.

In Table 1, 4th column indicates the optimum threshold for 20 different sequences that we used for our experiments. The PSNR[1], that employed the prediction mean squared error (MSE), is used for extracting optimum threshold. The Fig. 1 shows the results for all the thresholds tested with PDC for Foreman sequence. Variation of optimum threshold motivated us to propose an adaptive technique for threshold selection in PDC criterion. We try to achieve nearest PSNR to the optimum PDC while the computation burden don't increased considerably.

The organization of paper is as follows. In the next section we explain our proposed algorithm (APDC). Some of the experimental results and conclusions are presented in section 3 and 4, consequently.

2 Adaptive PDC

2.1 Main Idea

Adaptive selection of a threshold for PDC helps us obtain proper results in pixel difference classification for various image sequences.

We can distinguish two different classes for pixel differences; matched points and missed-match points. By assuming zero mean normal distribution for these two classes, as shown in Fig. 2, we used parametric classification to classify pixel differences. Therefore we must estimate variance of these two classes. We select crossover point of two distributions as threshold for classification. This certain threshold, t_g, can be used for Eq. 2.

[1] Peak Signal-to-Noise Ratio

Fig. 3. Original and motion compensated frame 17 of the Foreman sequence and frame 16 of the Silent sequence produced by full search with *N*=8, *w*=7. From left to right: Original, SSD Criterion and APDC criterion.

2.2 The Threshold Selection

We need to estimate the variances of both classes for determination of t_g. The variances calculated from each frame are used for next frame. But calculation of variances require some mathematical operations, which are time consuming, therefore we must calculate variances by minimum operations. For this purpose, we used only one point per block for calculation of variances, one pixel difference from best-match point and one pixel difference for other test points in search area. After calculation of variances, we must obtain crossover point of two distributions. By normal distribution assumption, t_g calculated as follows:

$$\frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{t_g^2}{2\sigma_1^2}} = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{t_g^2}{2\sigma_2^2}} \tag{3}$$

$$\Rightarrow \quad t_g = \left(2\log(\frac{\sigma_1}{\sigma_2}) / (\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}) \right)^{\frac{1}{2}} \tag{4}$$

where σ_1 and σ_2 are standard deviation of matched points class and missed-match points class, respectively. Various factors can cause the standard deviation of matched points class to be increased. One of the most important factors is motion in the scene. We have restricted the motion vector estimation to integer pixel grid. The true frame-to-frame displacements are unrelated to the sampling grid and thus, this restriction can increase the matched points class's standard deviation [7] and this can causes increasing the t_g. Table (1) shows that t_g is proportion with motions in the scene. As the motion in the scene is higher, so the t_g will increase. If $\sigma_1^2 << \sigma_2^2$, the equation (4) can be simplified as following:

$$t_g \approx \sigma_1 \left(2\log(\frac{\sigma_2}{\sigma_1}) \right)^{\frac{1}{2}} \tag{5}$$

This threshold is used for PDC calculation of next frame. Variances can calculate one time per some of frames for further calculation burden reduction.

3 Experimental Results

Our experiments are performed on twenty image sequences which each frame is 176x144, and *N=8, w=7*. We used PSNR as the performance measure. First, for different image sequences we used various threshold values from 2 to 40 and selected value that

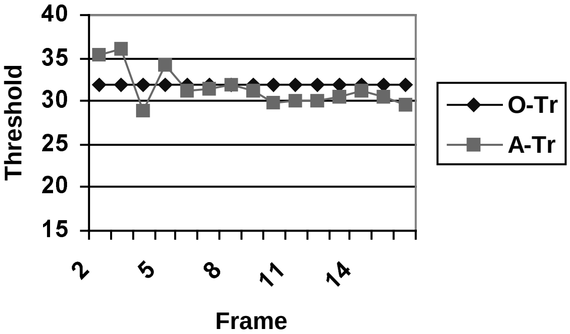

Fig. 4. Comparison of optimum threshold (O-Tr) and adaptive threshold (A-Tr) for Flower Sequence

maximized PSNR as optimum threshold. The 3rd and 4th columns of Table (1) show the optimum PSNR and optimum threshold for image sequences, respectively. Then, for adaptive PDC our operations explain as following: assuming that the mean values of matched and missed-match classes are zero, we calculate the variance of these two classes. For any test block in the search area, we calculate squared error for one of pixels of block (center pixel). We used only one pixel to reduced calculation burden. The squared error at maximum PDC value location for any search area used for calculation of matched point class's variance. Other squared errors used for calculation of missed-match point class's variance. Then the threshold value, calculated using Eq. 5. This threshold value is used for next frame as threshold for PDC calculations. The PSNR obtained for this method for different sequences presented in 5th column of Table (1). In this column we present the percentage of PSNR by using SSD's PSNR as reference. The 6th column of Table (1) presents the ratio of PSNR for APDC and optimum PDC, in percentage. This column indicates the APDC can outperform 99% of optimum performance. Fig. 3 shows the motion compensated frame obtained from SSD and APDC criteria.

Fig 4 shows the optimum threshold and adaptive threshold for Flower sequence.

In order to evaluate the adaptability of APDC, we constructed a sequence with different motions. For this purpose, we used Flower after Missamerica in our sequence. Fig. 5 shows the threshold variation throughout the sequence.

4 Conclusions

An adaptive approach to pixel difference classification was presented in this paper. In this method we calculated a threshold by using a normal distribution assumption for matched and missed-match classes from a frame and then used this threshold for next frame.

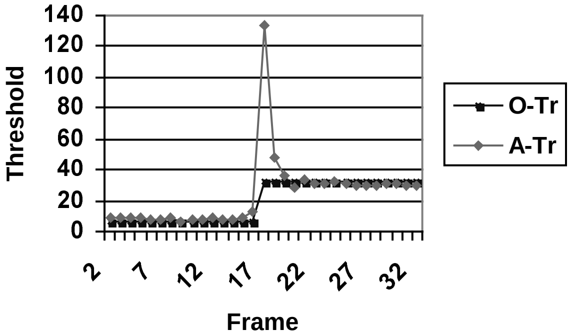

Fig. 5. Threshold Variation of APDC throughout a sequence with different motion. This sequence is constructed by using Flower sequence after Missamerica sequence.

Advantage of our method verified by conducting tests on 20 image sequences. Experimental results shown that this method can outperform 99% of optimum performance.

References

[1] H.R. Pourreza, M.Rahmati and F. Behazin, " Simple and Efficient Bit-Plane Matching Algorithms for Video Compression", *Proc. Of Workshop on Real-Time Image Sequence Analysis,* pp. 33-42, Aug.-Sept. 2000.

[2] H. Gharavi and M. Mills, "Block Matching Motion Estimation Algorithms; New Results", *IEEE Trans. On Circuits and Systems*, Vol. 37, No. 5, pp. 649-651, May 1990.

[3] V. Bhaskaran and K. Konstantinides, "**Image and Video Compression Standards**", *Kluwer Academic Publishers*, 1997.

[4] Y. Chan, S.Y. Kung, "Multi-Level Pixel Difference Classification Methods", *Proc. Of IEEE Int. Conf. On Image Processing*, pp. 252-255, 1995.

[5] G.J. Nelson "The Block Matching Approach to Motion Estimation", *M.Sc. Dissertation, Department of Computer Science, University of Warwick, UK.*, September 1993.

[6] E. Chan, A.A. Rodriguez, R. Ghandi, S. Panchanathan, "Experiments on block-matching techniques for video coding", *Multimedia Systems*, Volume 2, Number 5, pp. 228 - 241, Dec. 1994.

[7] B. Girod, "Motion-Compensation Prediction with Fractional-Pel Accuracy", *IEEE Trans. on Communications*, Vol. 41, No. 4, pp. 604-612, 1993.

Table 1

Sequence	SSD's PSNR	Opt. PDC's PSNR	Opt. Tr.	Adap. PDC's PSNR	PSNR APDC/OPDC
Akiyo	38.14	36.95	4	36.24 (95.0%)	98.1%
Bus	24.19	23.20	28	23.09 (95.4%)	99.5%
Carphone	32.74	30.80	12	30.69 (93.7%)	99.6%
Claire	37.07	35.40	4	34.91 (94.2%)	98.6%
Coastguard	28.96	28.15	16	28.09 (97.0%)	99.8%
Container	34.93	34.26	4	33.30 (95.3%)	97.2%
Flower	22.93	22.42	32	22.41 (97.7%)	100%
Football	23.64	22.69	31	22.66 (95.9%)	99.9%
Foreman	30.69	29.19	15	29.03 (94.6%)	99.5%
Grandma	36.60	35.79	7	35.79 (97.8%)	100%
Hallmonitor	37.83	37.06	5	36.23 (95.8%)	97.8%
Missamerica	38.45	37.04	6	36.72 (95.5%)	99.1%
Mobilecal.	24.72	24.23	25	24.16 (97.7%)	99.7%
News	32.71	30.95	15	30.83 (94.3%)	99.6%
Salesman	34.30	32.95	14	32.75 (95.5%)	99.4%
Silent	33.01	31.48	12	31.18 (94.5%)	99.0%
Stefan	25.77	25.17	23	25.07 (97.3%)	99.6%
Suzie	34.06	32.29	8	32.29 (94.8%)	100%
Tennis	28.01	26.18	12	25.89 (92.4%)	98.9%
Trevor	31.61	29.42	13	29.25 (92.5%)	99.4%
Average				95.4%	99.2%