NEW ADAPTIVE PARTIAL DISTORTION SEARCH USING CLUSTERED PIXEL MATCHING ERROR CHARACTERISTIC

Ko-Cheung Hui, Wan-Chi Siu and Yui-Lam Chan*

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
Tel: (852) 27666229 Fax: (852) 23624741 email: enwcsiu@polyu.edu.hk

ABSTRACT

The partial distortion search is a particular attractive fast block-matching algorithm, because it introduces no prediction error as compared with the full-search algorithm. It reduces the number of necessary matching evaluations for every searching point to save computation. In the literature, many researches have tried to improve block-matching algorithms by making use of an assumption that pixels with larger gradient magnitudes have larger matching errors on average. Based on a simple analysis, we have found that on average, pixel matching errors with similar magnitudes tend to appear in clusters for natural video sequences. By using this clustering characteristic, we propose an adaptive partial distortion search algorithm which significantly improves the computational efficiency of the original PDS. This approach is much better than other pixel gradient based adaptive PDS algorithms. In addition, our proposed algorithm is suitable for motion estimation of both opaque and boundary macroblocks of an arbitrary shaped object in MPEG-4 coding.

1. INTRODUCTION

Block-based motion compensation technique has been widely used to reduce the redundancy between successive frames in many modern video coding standards [1,2]. By the block-based motion compensation technique, the values of pixels in a macroblock (MB) in the current frame are predicted from another MB of pixels in a reference frame. The displacement between these two MBs is defined as a motion vector (u,v). The motion vector is obtained by minimizing a measure of matching distortion between these two MBs. One of the most frequently used criteria to measure a matching distortion is the sum of absolute difference (SAD).

The traditional full search algorithm (FSA) can give an optimal solution by exhaustively searching all possible locations within the search window. However, this algorithm suffers from heavy computational load.

The partial distortion search (PDS) [3,4] is a fast algorithm which has identical quality as that of the FSA. The PDS [3] reduces the computation complexity by terminating the SAD calculation early when it finds that a partial SAD is already greater than the minimum SAD encountered so far in the searching.

Let us define a generalized form of the partial SAD of a MB at position (x,y) as shown below,

$$SAD_p(x,y;u,v)=\sum_{j=0}^{p}\sum_{i=0}^{15}\left|I_t(x+k_n,y+l_n)-I_{t-1}(x+k_n+u,y+l_n+v)\right| \quad (1)$$

where $\{(k_n,l_n)\mid n=0,\ldots,j\times16+i,\ldots,256\}$ is an index set of all pixels in a MB and p specifies a stopping position of a partial SAD. The index set determines the coordinates and orders of the pixel matching errors accumulated to the SAD_p. In this paper, we propose an adaptive PDS algorithm by using a characteristic of clustered pixel matching errors. It can be shown that pixel matching errors with similar magnitude tend to appear in a cluster in natural video sequences. An adaptive index set is formed based on this characteristic, thus a pixel with greater matching error is accumulated to the SAD_p earlier than other pixels. As a result, the SAD calculation can be terminated earlier.

Experimental results show that our proposed algorithm has a significant speed-up when compared to the conventional PDS, and other PDS algorithms which make use of pixel gradient to predict pixel matching errors, such as the PDS algorithm using representative pixels and adaptive matching scan (AMS-PDS) [5].

2. THE CHARACTERISTIC OF CLUSTERED PIXEL MATCHING ERROR

In order to accumulate a pixel with greater matching error to the SAD_p earlier than other pixels according to the order indicated by an index set, it is necessary for us to investigate possible spatial distributions of pixel matching errors in a MB. We have found that errors with similar magnitude tend to appear together in clusters. It is because natural images are dominated by low frequency components. The matching errors of low frequency regions between a target MB and a candidate MB have similar magnitudes and are partitioned by edge pixels of these two MBs. This phenomenon is demonstrated in Fig. 1. Fig. 1(a) depicts the matching of a one dimensional (1-D) target MB (thick continuous line) within a 1-D search window (thin dotted line). The corresponding pixel matching errors appear in a cluster form as shown in Fig. 1(b).

Edges are the most prominent feature in image processing. They are also frequently used to predict pixel matching errors in motion estimation. The prediction is accurate especially near a minimum distortion position. Fig. 2 shows that locations with large pixel matching errors (the hatched region) can be detected by using pixel gradients when the target MB is located near a good candidate MB. However, the result is not good enough in general. In Fig. 1(b), only the pixel matching errors in the hatched region are found, while matching errors outside the hatched region are underestimated.

According to the above analysis, we can forecast that clustered pixel matching error characteristic can be used to achieve greater advantage in an adaptive partial distortion search.

3. PROPOSED ALGORITHM

3.1 Determination of an adaptive index set

For a target MB, the positions of its pixels are represented by an index set, $S = \{(k_n, l_n) \mid n = 0,..., N-1\}$, where N is the number of pixels in a MB. For a single pixel at $s_n = (k_n, l_n)$, $s_n \in S$, its matching error is $e(s_n) = I_t(s_n) - R(s_n)$, where $R(s_n)$ is a random variable which represents the pixel value at s_n of a candidate MB. For the sake of simplicity, in the following discussion, s_n is replaced by n, and both MB location (x,y) and motion vector (u,v) are dropped. To improve the saving in computation of a PDS, pixel matching errors with an ideal index set must have the following relation,

$$e(0)^2 \geq ... \geq e(n)^2 \geq ... \geq e(N-1)^2 \quad (2)$$

Let us define $p(n)$ as the predicted pixel matching errors, $p(n) = I_t(n) - m$, where m is a reference value to be used to obtain the prediction. One possible solution of m is to minimize the expected value of the sum of squares of the difference between $e(n)^2$ and $p(n)^2$,

$$\text{i.e.} \quad m = \arg\min_m \left\{ E\left[\sum_{n=0}^{N-1} \left[(I_t(n) - R(n))^2 - (I_t(n) - m)^2 \right]^2 \right] \right\}$$

In solving this equation, we have

$$\frac{d}{dm} E\left[\sum_{n=0}^{N-1} \left[\left(I_t(n) - R(n)\right)^2 - \left(I_t(n) - m\right)^2 \right]^2 \right] = 0 \quad (3)$$

By substituting $R(n) = I_t(n) - e(n)$ into eqn. (3) and assuming that natural images are dominated by low frequency components, we finally can obtain three approximate real roots m of a cubic equation,

$$m \approx \bar{I}_t \quad \text{or} \quad m \approx \bar{I}_t \pm \sqrt{\overline{e^2}} \quad (4)$$

It must be note that these solutions are valid only if an image consists mainly low frequencies and the standard deviation of $I_t(n)$ is small enough. However, a shifting of m would not affect our purpose dramatically in general.

The first approximated root is the mean of pixel values in the target MB. The meaning of the other roots can be interpreted as the following. Intuitively, m is a function of pixel values in a candidate MB, i.e. $m = m(R(n))$. The other roots, $\overline{I_t} \pm \sqrt{\overline{e^2}}$ can also be obtained by the following consideration. We observed that $e(n)$ often consists of two components, $e_m(n)$ and $e_w(n)$. $e_w(n)$ denotes zero-mean white noise with negligible magnitudes. $e_m(n)$ represents errors due to irregular motions, light variation, etc. They tend to have a same sign in a block. Hence, we assume that $\bar{R} \approx \bar{I}_t \pm |\bar{e}| \approx \bar{I}_t \pm \sqrt{\overline{e^2}}$. It indicates an approximation of the mean of pixel values in a candidate MB.

3.2 Clustered Pixel Matching Errors for Adaptive Partial Distortion Search (CPME-PDS)

The earlier that the global minimum is met in a search can improve better the computational efficiency of a PDS. To achieve this purpose, we use two strategies as listed below.

1. The outward spiral scanning can be used to exploit the center-biased motion vector distribution characteristics of the real world video sequence [6].
2. The correlation in the motion field is exploited by using a median predictor described in [7].

According to the above considerations and our analytical results, we suggest using the mean of pixel values in the candidate MB of the initial searching point to compute the reference value, m, because we can assume that $\overline{I_{t-1}(i+u_{med}, j+v_{med})} \approx \overline{I_t(i,j)}$, where (u_{med}, v_{med}) = the median predictor. The CPME-PDS approach can be summarized as follows:

Note that all division operations in the following description are integer division with truncation toward zero for the sake of lower complexity.

Step 1) Determine the median predictor, (u_{med}, v_{med}), according to the description in [7]

Step 2) Calculate the reference value, m, with the median predictor, (u_{med}, v_{med}).

$$m = \frac{1}{256} \sum_{j=0}^{15} \sum_{i=0}^{15} I_{t-1}(x + u_{med} + i, y + v_{med} + j)$$

Step 3) Initialize an index set, $S' = \{(k'_n, l'_n) \mid n = 0,..., N-1\}$.

Step 4) Calculate the expected absolute pixel matching error, $|p_{exp}(n)|=|I_t(k'_n,l'_n) - m|$, of each pixel in the target MB.

Step 5) Rearrange the order of set S' to obtain an adaptive index set S by sorting the expected absolute pixel matching error, $|p_{exp}(n)|$, in descending order, i.e. $|p_{exp}(0)| \geq ... \geq |p_{exp}(n)| \geq ... \geq |p_{exp}(N-1)|$ with $S = \{(k_n, l_n) | n = 0,..., N-1\}$

Step 6) Apply the adaptive index set, S, to calculate the partial SAD in eqn. (1) during the searching in an outward spiral scanning.

It is straightforward to modify the above procedure for boundary MBs of an arbitrary shaped video object in MPEG-4 [7]. First, the reference value, m, is calculated after that the repetitive padding is applied to a reference video object plane (VOP). It is only necessary to compute the expected pixel matching error, $p_{exp}(n)$, for opaque pixels in the case of a boundary MB. Note that, for the index set, N is equal to number of opaque pixels in the boundary MB. Second, the partial SAD in eqn. (1) is rewritten as,

$$SAD_p(x,y;u,v) = \sum_{j=0}^{p}\sum_{i=0}^{q}\left|I_t(x+k_n, y+l_n) - I_{t-1}(x+k_n+u, y+l_n+v)\right|$$

where $p \in \{\aleph \mid 0,...,\alpha; \alpha = \text{integer division of } N/16\}$

$$q = \begin{cases} 15 & , p \neq \alpha \\ N - 16\times\alpha & , p = \alpha \end{cases}$$

3.3 Analysis of the Computational Overhead

From the above description, it is shown that the additional computation introduced by the CPME-PDS is the process to construct the adaptive index set for each target MB in the current frame or VOP. These overheads include the calculations of m, $|p_{exp}(n)|$, and the check to ensure that only opaque pixels are involved in a boundary MB. To obtain the final adaptive index set, S, a sorting process is required in step 5 of Section 3.2. The counting sort [8] is used in our implementation. Note that all multiplications and divisions required can be implemented with simple bitwise shift operations. In our analysis, however, each multiplication or division is counted and assumed to be equivalent to 8 additions for the sake of simplicity.

4. EXPERIMENTS

In order to compare the efficiency between the CPME-PDS and pixel gradient based PDS, we have to modify the adaptive PDS to become a pixel gradient based algorithm (PG-PDS).

4.1 Pixel Gradients based Adaptive PDS (PG-PDS)

In the PG-PDS, an adaptive index set, S_{pg}, is obtained based on the magnitude of individual pixel gradient. For each pixel in an opaque MB, let us express the magnitudes of x-directional gradients, G_x, and y-directional gradients, G_y, as,

$$\begin{aligned} |G_x(x,y)| &= |I_t(x,y) - I_t(x+1,y)| \quad \text{where } x = 0,\ldots,14 \text{ ; } y = 0,\ldots,15 \\ |G_y(x,y)| &= |I_t(x,y) - I_t(x,y+1)| \quad \text{where } y = 0,\ldots,14 \text{ ; } x = 0,\ldots,15 \end{aligned} \tag{8}$$

These magnitudes are sorted in descending order with a counting sort. The S_{pg} is then established by extracting the pixel's position according to the order of the sorted gradient magnitudes. Obviously, each pixel must appear only once in S_{pg}. It is necessary to check this to prevent double extraction of a pixel, because each pixel involves two directional gradient magnitudes. The adaptive index set, S_{pg}, is applied for the calculation of the SAD_p in eqn. (1).

There are some differences between the implementation of a boundary MB and an opaque MB. In calculating eqn. (8), if one of the involved pixels is a transparent pixel, the magnitude of the corresponding gradient is regarded as zero.

4.2 Experimental Results

To evaluate the performance of the CPME-PDS, we have implemented five algorithms: the FSA, the conventional PDS, the AMS-PDS [5], the PG-PDS and the CPME-PDS. The outward spiral scanning was applied to all five algorithms. In addition, a median predictor was used as an initial searching centre to exploit the correlation in the motion field for all tested algorithms. Because AMS-PDS is an algorithm developed only suitable for block based motion estimation, experiments which involved arbitrary shaped video objects did not include AMS-PDS. The computational efficiencies of the algorithms were assessed in terms of the number of operations required for the searching.

In our implementation, quick sort was used as the sorting approach for AMS-PDS. Note that the selection of a sorting algorithm seriously affects the performance of an adaptive PDS. In order to prevent under-evaluation of AMS-PDS, the numbers of operations in its sorting process were excluded in all of our experiments.

Table 1 demonstrates a comparison between the computational efficiencies of the tested algorithms. The computational efficiencies were compared in terms of the average number of operations per MB including their overheads (except AMS-PDS) and the speed-up ratios. The results show that our proposed algorithm, CPME-PDS, can successfully improve the computational efficiency of the conventional PDS and is better than other adaptive PDSs. In terms of speed-up ratios, it can achieve a speed-up ranging between 3 and 9 times of the FSA.

5. CONCLUSIONS

We have proposed an adaptive partial distortion search algorithm, the clustered pixel matching error for adaptive partial distortion search (CPME-PDS), for motion vector estimation. The algorithm makes use of a phenomenon that pixel matching errors in a MB with similar magnitudes tend to appear together and in cluster form, in natural video sequences. According to this phenomenon, we have found that means of pixel values in a candidate MB be a good reference value to predict the magnitude of each pixel matching error in a target MB. Hence, the mean of pixel values in the initial candidate MB at the centre of a search window have been used to calculate a reference value and to construct an adaptive index set. As a result, the pixel matching error with larger magnitude can be accumulated to the SAD_p sooner than others and the SAD calculation can be terminated at an early stage. Our experimental results show that the computational efficiency of CPME-PDS outperforms other tested algorithms, including PG-PDS which were introduced for comparison in this paper. Our experimental results show that CPME-PDS can have a speed-up of 3 to 9 as compared with FSA, depending upon the contents of the coded video sequences. The major advantages of CPME-PDS are its high efficiency and conceptual simplicity.

6. ACKNOWLEDGMENTS

This research is supported by the Centre for Multimedia Signal Processing, Hong Kong Polytechnic University. K.C. Hui acknowledges the research grant G-V867 provided by the University.

7. REFERENCES

[1] ITU-T H.263, Video coding for low bit rate communication, Mar. 1996.

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

[3] ITU-T recommendation H.263 software implementation, Digital Video Coding Group, Telenor R&D, 1995.

[4] S. Eckart and C. Fogg, ISO/IEC MPEG-2 software video codec, Proc. SPIE 2419, 1995, 100-118.

[5] Jong-Nam Kim and Tae-Sun Choi, "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, No. 7, pp. 1040 - 1048, Oct. 2000.

[6] Reoxiang Li, Bing Zeng and Liou, M.L., "A New Three-Step Search Algorithm for Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 4, No. 4, pp. 438-442, August 1994.

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

[8] Hosam M. Mahmoud, "Sorting: A Distribution Theory", John Wiley & Sons, 2000.

Table 1. Average numbers of operations per MB of the tested algorithms in a search window with a search range equal to 15 (i.e. $-15 \le u, v \le 15$).

	FSA	Speed up ratio	PDS	Speed up ratio	AMS-PDS	Speed up ratio	PG-PDS	Speed up ratio	CPME-PDS	Speed up ratio
Video Sequences										
Football	738048	1.00	299882	2.46	279341	2.64	263057	2.81	232779	3.17
Tabletennis	738048	1.00	219509	3.36	194414	3.80	170269	4.33	149103	4.95
Stefan	738048	1.00	241830	3.05	206729	3.57	191230	3.86	169652	4.35
Salesman	738048	1.00	160614	4.60	140468	5.25	126625	5.83	111693	6.61
Foreman	738048	1.00	153344	4.81	123309	5.99	110916	6.65	106310	6.94
Grand mother	738048	1.00	157032	4.70	129139	5.72	126619	5.83	121658	6.07
Suzie	738048	1.00	170286	4.33	137157	5.38	135321	5.45	121301	6.08
Trevor	738048	1.00	110638	6.67	93004	7.94	89113	8.28	81240	9.08
Arbitrary Shaped Video Objects										
News	689274	1.00	188931	3.65			147191	4.68	138052	4.99
Children	653006	1.00	188931	3.46			116043	5.63	99908	6.54
Bream	698661	1.00	219654	3.18			155870	4.48	139461	5.01
Goldfish	663847	1.00	226663	2.93			151615	4.38	140129	4.74

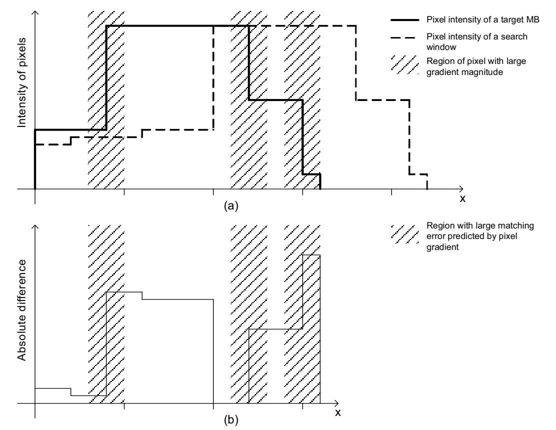

Fig.1. (a) Matching of a 1-D target MB within a 1-D search window. (b) Corresponding pixel matching error of the target MB at the current position.

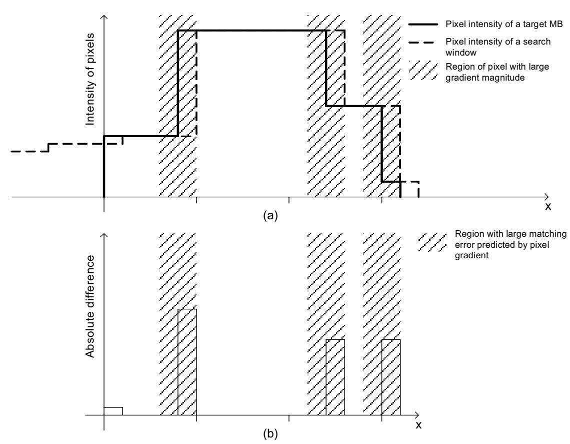

Fig. 2. (a) Matching of a 1-D target MB within a 1-D search window near a minimum distortion location. (b) Corresponding pixel matching error of the target MB at the current position.