ADAPTIVE MOTION SEARCH WITH ELASTIC DIAMOND FOR MPEG-4 VIDEO CODING[1]

Weiguo Zheng[2], Ishfaq Ahmad and Ming Lei Liou

Multimedia Technology Research Center
Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

ABSTRACT

Video coding is a complex process, comprising a combination of spatial, temporal and statistical data reduction techniques. Of these techniques, motion estimation taking advantage of inter-frame information redundancy plays the most vital role. The overwhelming complexity of motion estimation using a brute-force search has prompted researchers to propose myriad of algorithms, yet finding the "most efficient" algorithm remains an open research problem. Recently, the MPEG-4 committee, after a rigorous evaluation, have recommended a motion estimation algorithm that performs very well in terms of speed and picture quality. In this paper, we propose a motion estimation algorithm that is a combination of a number of novel ideas for finding more accurate motion vectors and with a faster speed. The proposed algorithm, named as adaptive motion search with elastic diamond (AMSED) algorithm, takes advantages of the correlation between motion vectors in both spatial and temporal domains, and uses special diamond shape search patterns to accelerate motion search. The new algorithm achieves very close quality compared to full search but with several hundred times speedup. In term of speed, compared with the motion estimation technique recommended by the MPEG-4 committee, AMSED can achieve more than 200% speedup.

1. INTRODUCTION

In video compression standards, such as H.26x and MPEG, the basic idea is to estimate the frame-to-frame or field-to-field displacement of moving pixels, transform the prediction error between reference and predicted picture (DCT), and quantize and encode these DCT coefficients. Accurate motion vector identification with reasonable computational complexity is important for designing an efficient and cost-effective video encoder.

The most common motion estimation method is the block matching technique. In this framework, a video frame is divided into blocks (usually 16×16 pixels or 8×8 pixels) and a search window is defined. Each block of the current frame is compared with all the blocks of the reference frame within the search window. Displacement with the maximum correlation or the minimum distortion between the current block and the reference blocks within the search window is selected as the motion vector. Usually, the sum of absolute differences (SAD) is used to measure the distortion of matching.

$$SAD\ (u,v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left| p_{t-1}(i+u,\ j+v) - p_t(i,\ j) \right|, \qquad -R \le u,v \le R \tag{1}$$

where $p_{t-1}(i,j)$ and $p_t(i,j)$ refer to the gray levels of pixels in the reference and the current pictures, respectively, and N is the dimension of the block, and R refers to the size of the search window.

The full search algorithm checks every point within the search window. While the performance of the full search is considered to be "optimal", its complexity is extremely high. It consumes up to 85% of the computational resource used for encoding in our experiments. But since the full search aims to find the minimum SAD, presence of noise in a video can cause the search to find motion vectors that do not lead to best encoding. Presence of noise can also cause full search to produce chaotic motion field for a smooth motion video, costing more bits to encode motion vectors with fewer bits left for encoding DCT coefficients within a given bit budget.

For MPEG-4 video encoding, motion estimation (ME) is especially important, since MPEG-4 video encoder is usually used for low bit rate encoding, accurate motion prediction and the more bits used for encoding prediction error would result in the higher picture quality.

To overcome the high computational complexity of motion estimation, a large number of fast algorithms are reported. Fast motion estimation algorithms fall into following classes: 1) fast search by a reduction of motion vector candidates; 2) fast matching distortion computation; and 3) block pixel decimation [1], [2]. Some examples are three-step search, new three-step search, 2-D logarithmic search, the conjugate directional search [3],

[1] This work was supported by Research Grants Council of Hong Kong under contract # CRC98/01.EG05 and HKUST6228/99E.

[2] Weiguo Zheng is currently with Department of Electronics and Communication Engineering, Zhongshan University, Guangzhou, China

0-7803-6725-1/01/$10.00 ©2001 IEEE

[4], [5], [6], and hierarchical search [7], [8], [9]. Almost all of the fast search algorithms make explicit and implicit assumptions: the matching distortion increase monotonically as the checking point moves away from the global minimum or the error surface is unimodal over the global window. Indeed, this assumption is not always true. Consequently, the resultant motion vectors may be trapped in a local minimum.

Some recent predictive algorithms use motion vectors of neighbour Macroblocks (MBs) as predictors of the motion vector for the current MB. The candidate vector is tested and refined in subsequent processing. The MPEG-4 part 7 has adopted MVFAST (motion vector field adaptive fast search technique) [10] as the core technology for fast motion estimation. A derivative of MVFAST, called Predictive PMVFAST (PMVFAST), is considered as an optional approach for MPEG-4 fast motion estimation [10]. Both MVFAST and PMVFAST are based on diamond search patterns [11]. PMVFAST is reported to be about 50% faster than MVFAST with the similar picture quality. But the implementation complexity of PMVFAST is also higher than that of MVFAST.

In this paper, we propose a motion estimation algorithm that is a combination of a number of novel ideas for finding more accurate motion vectors and with a faster speed. The proposed algorithm, named as adaptive motion search with elastic diamond (AMSED) algorithm, takes advantages of the correlation between motion vectors in both spatial and temporal domains, and uses special diamond shape search patterns to accelerate motion search.

2. ADAPTIVE MOTION SEARCH WITH ELASTIC DIAMOND (AMSED)

AMSED uses three search patterns: Small diamond pattern (SDP) and large diamond pattern (LDP), and elastic diamond pattern (EDP) (see Fig.1). The algorithm selects these patterns adaptively depending upon a notion called *local motion activity* (LMA, defined below). The pattern can switch to each other if needed; the switching pattern is the EDP pattern

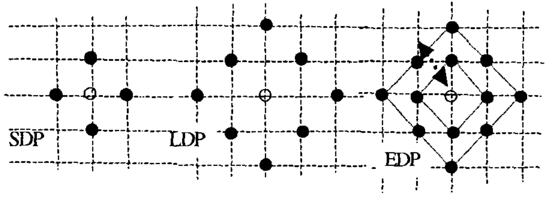

Fig.1: Search Patterns

A motion vector candidate list (MVCL) includes motion vectors from its neighbour MBs and the reference picture. The basic MVCL includes motion vectors from neighbour MBs as shown in Fig.2 where MB(i, j) is the current MB.

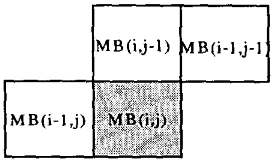

Fig.2: Basic MVCL

The major features of AMSED are:

- Local motion activity (LMA) measurement.
- Using LMA to detect MBs belonging to the same moving object.
- Adaptive threshold for identifying stationary blocks.
- Adaptive threshold to enable half-stop.
- Keeping track of accessed checking points.
- Extended motion vector candidate list.
- Motion inertia from temporal domain.
- Interpolated motion vector for BVOP.
- Motion search with elastic diamond pattern.

The smoothness of motion field, especially, within the same moving object, means high correlation of motion vectors. This property can be used to accelerate motion search if we can identify the current MB is within the same moving object of its adjacent MBs.

$$l_i = \left|x_i - \bar{x}\right| + \left|y_i - \bar{y}\right| \qquad (2).$$

$$L = Max\ \{l_i\} \qquad (3).$$

Where L is LMA measurement factor. $(\bar{x}, \bar{y})$ is the average of motion vectors in basic MVCL, and (x_i, y_i) is motion vector in MVCL.

The notion of LMA makes the measurement more localized and more detail. It also reflects the consistency of motion vectors within MVCL. It can be observed that if the current MB and MBs in MVCL belong to the same moving object, L should have a small value. The LMA is defined as:

$$LMA = \begin{cases} Low\ ,if & L \le L1 \\ Medium\ ,if & L1 < L \le L2 \\ High\ ,if & L > L2 \end{cases} \qquad (4)$$

In our experiments, $L1 = 1$, $L2 = 4$. When LMA is low, it means its adjacent MBs have similar motion property. The current MB may possibly be inside the same object with its adjacent MBs, and may have the same motion vector.

Based on the research results and observations, most stationary MBs have small SADs at (0, 0). If we can detect a stationary macroblock (MB), we can just set its motion vector as (0, 0) and skip the motion search.

Usually, the stationary blocks are detected by static threshold, like 512 for MB size of 16×16 [10]. We propose an adaptive threshold TH0 to make this detection faster and robust because it can enhance the resistance of noise.

- If all adjacent MBs in MVCL have MVs at (0, 0), we use the maximum of their SAD as threshold TH0.
- If one of adjacent MBs in MVCL has MV unequal to (0, 0), we use the minimum of SAD of adjacent MBs as TH0 since possibility of current block inside stationary area become lower.
- To make the detector more robust, we set the lower bound of TH0 as MB_SIZE×2; the upper bound of TH0 is set MB_SIZE×4.
- If the SAD at (0, 0) is less than Th0, we skip the rest of the search and use (0, 0) as the motion vector of the current block.

An adaptive half-stop threshold (TH_HS) is added to make search stop more quickly. Threshold is selected according to LMA of its MVCL.

- TH_HS is equal to the mean value of SAD of adjacent MBs if the local motion activity is less than 4.
- TH_HS is equal to the minimum value of SAD of adjacent MB.

In diamond search, some search points are visited more than once. Duplicate search can be avoided by keeping the record of all the search points visited and testing if the current search point has been accessed before. Thus further improvement of speedup can be achieved by removing unnecessary actions.

Since AMSED is also based on motion prediction technique, the candidates in MVCL are crucial for speed and quality of AMSED. If there are more motion vector candidates, the chance to find true vector faster and more precise is higher. Fig.1 shows ordinary members in MVCL, presenting the prediction from the spatial domain. However, the correlation of motion vectors does not exist only in the spatial domain, but exists in the temporal domain as well. AMSED uses one motion vector prediction from the temporal domain.

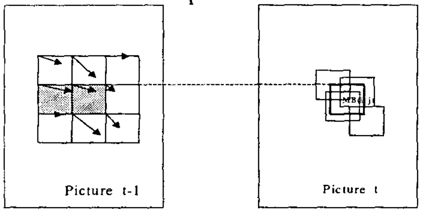

Fig.3: Temporal motion vector prediction.

Some temporal motion vector prediction algorithms are reported [10]. But they usually use only the average of motion vectors of adjacent MBs in the reference picture, or the motion vector of corresponding MB in the reference picture. As shown in Fig.3, for $MB(i, j)$, the prediction of motion vector from temporal domain would be $V_{t-1}(i, j)$, or the average of motion vectors of adjacent MBs in picture t-1. Based on our experiments and observations, the motion track of a moving object in a video sequence is continuous except when scene change occurs. That means there is a so-called *motion inertia* property in the temporal domain. The property can be used for motion vector prediction and interpolation.

Let (x_0, y_0) presents the coordinates of left-top corner of current $MB(i, j)$. (V_x, V_y) is the motion vector of surrounding MB in reference VOP, and (x_t, y_t) is coordinates of the left-top corner of the MB. Based on the inertia property (the MB moves with the same motion vector), the position of MB in next picture will be:

$$x_t = x_{t-1} + V_x$$

$$y_t = y_{t-1} + V_y \qquad (5).$$

$$d = |x_0 - x_t| + |y_0 - y_t| \qquad (6)$$

And the goal of motion vector prediction is to minimize *d*.

$$V_t(x, y) = \arg\min \sum_{V_{t-1}} d \qquad (7)$$

For example, in Fig. 3, motion vector of $MB(i-1, j)$ in picture (t-1) is selected as V_t. V_t is added into MVCL as motion vector candidate.

When LMA is high, an additional motion vector resulting from median filter is added to MVCL.

$$V_{Mx} = Median(V_{1x}, V_{2x}, V_{3x})$$

$$V_{My} = Median(V_{1y}, V_{2y}, V_{3y})$$

For instance, if V1=(-1,4), V2=(1,3) and V3=(4,5), then V_{Mx} = 1 and V_{My} = 4.

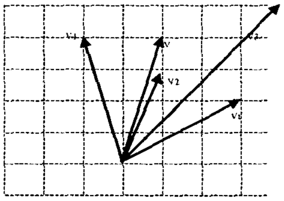

Fig.4: An example of motion vector candidate list.

For BVOP, the motion vectors can be interpolated from its reference VOPs. The interpolated motion vectors are computed by inertia criteria and scaling. The motion vectors are used as V_t for BVOP. From observation, we found that the interpolated motion vector can provide good prediction of motion vector for BVOP, and this the motion search can be terminated quickly.

The selection of the search patterns is according to the following principles:

- SDP is used to refine predicted motion vector. Once the minimum of SAD is located at the center of diamond, the center represents the motion vector, and the search can be terminated.
- LDP is used for fast large-range search and to avoid the search from being trapped at local minima. Once

the SAD of search center is found as the minimum, the search pattern changes to SDP in order to refine this estimation. SDP is used once in this scenario.

- If the number of successively executed SDP exceeds a certain predefined constant, *called elastic factor K*, the search pattern switches to LDP.

According to LMA classification, the search strategies are given in the following:

1) When LMA = Low, the search starts from SDP, and elastic factor $K = 2$, and initial search center is $\bar{V}$ of basic MVCL.
2) When LMA = Median, the basic MVCL is extended by adding V_t, and all motion vectors in MVCL are checked respectively. The candidate with the minimum SAD is selected as search center for consequent search. Then the search pattern is SDP, and elastic factor $K = 4$.
3) When LMA = High, the basic MVCL is extended by adding V_M and V_t. The candidates in MVCL are checked, and the one with the minimum SAD is selected as new search center. Consequent search is starts from SDP, and elastic factor $K = 1$.

3. PERFORMANCE COMPARISON

The experiments were done with Microsoft VM software at various bit rates with TM5 rate control. All testing sequences are rectangular videos with the CIF format, with two BVOPs between reference VOPs. The GOV size is 15. We use different types of motion sequences in the experiments. The sequences include the simple head & shoulder conference sequences and the sport video with fast and large dynamic motion. The search range and bit rate are set according to the characteristics of video sequences. The experiments at other bit rates also resulted with the similar behaviours. Full search and MVFAST are used for comparison. The results are listed in Table 1.

4. CONCLUSIONS

As we have shown in Table 1, AMSED outperforms MVFAST with much faster speed while yielding the similar picture quality with full search. MVFAST causes steep degradation when coding fast motion sequences, like Stefan, in our experiments. However, AMSED not only achieves the similar quality with full search, but also uses less checking points compared to MVFAST. The experiments show that AMSED can cope well with both large dynamic motion variation sequence and simple uniform motion video. AMSED is very suitable for real-time high quality MPEG-4 video encoding.

5. REFERENCES

[1] F.Dufaux and F.Moscheni, "Motion Estimation Techniques for Digital TV: A Review and a New Contribution," *Proceeding of IEEE*, vol.83, pp. 858-876, June 1995.

[2] J.R. Jain and A.K. Jain, "Displacement Measurement and Its Application in Interframe Image Coding," *IEEE Transactions on Communication*, Vol. COM-29, pp. 1799-1808, Dec. 1981.

[3] B. Liu and A. Zaccartin, "New Fast Algorithms for Estimation of Block Motion Vectors," *IEEE Transactions on Circuits & Systems for Video Technology*, Vol.3, No.2, pp. 148-157, Apr.1993.

[4] R. Li, B. Zeng, and M. L. Liou, "A New Three-Step Search Algorithm for Fast Motion Estimation," *IEEE Transactions of Circuits & Systems on Video Technology*, Vol.4, pp. 438-442, Aug.1994.

[5] B. Zeng, R. Li, and M. L. Liou, "Optimi-zation of Fast Block Motion Estimation Algorithms," *IEEE Transactions on Circuits & Systems for Video Technology*, Vol.7, pp. 833-844, Dec.1997.

[6] Z. He, and M. L. Liou, "A High Performance Fast Search Algorithm for Block Matching Motion Estimation," *IEEE Transactions on Circuits & Systems for Video Technology*, Vol.7, pp. 826-828, Oct.1997.

[7] C.K. Cheung and L.M.Po, "A Hierarchical Block Motion Estimation Algorithm Using Partial Distortion Measure," *Proceeding of ICIP'97*, Vol3, pp. 606-609, 1997.

[8] X. Song, T. Chiang, and Y.-Q. Zhang, "A Scalable Hierarchical Motion Estimation Algorithm for MPEG-2," *Proceeding of ICIP'98, pp.* IV126-IV129. 1998.

[9] X. Lee, Y.-Q. Zhang, "A fast hierarchical motion-compensation scheme for video coding using block feature matching," *IEEE Transaction of Circuits & Systems on Video Technology*, Vol.6, No.6, Dec.1996.

[10] ISO/IEC JTC1/SC29/WG11 N3324, "Optimization Model Version 1.0," Noordwijkerhout, NL, March'2000.

[11] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits & Systems for Video Technology*, Vol.8, No.4, pp. 369-377, Aug. 1998.

Table 1: Performance comparison of MVFAST and AMSED

Sequence Name	Search Range	Encoding bit rate (bps)	Full Search		MVFAST		AMSED	
			Checking points	PSNR (dB)	Checking points	PSNR (dB)	Checking points	PSNR (dB)
Akiyo	±16	384k	676640320	41.45	475595	41.50	233564	41.46
Foreman	±16	384k	676514032	31.74	2403299	32.08	1242178	32.10
Silent	±16	384k	676578496	33.63	1470387	33.64	746245	33.58
Table Tennis	±16	500k	676588912	32.89	1670492	32.66	1083018	32.72
Mobile	±16	1M	676551328	28.13	2320572	28.08	1581045	28.11
Stefan	±32	1M	2477291440	30.78	2207998	29.57	1714428	30.51