

MOTION VECTOR FIELD ADAPTIVE FAST MOTION ESTIMATION

*Prabhudev I. Hosur and Kai-Kuang Ma **

School of Electrical and Electronic Engineering
Nanyang Technological University
Nanyang Avenue
Singapore - 639798
*Email: ekkma@ntu.edu.sg

Abstract

This paper presents a fast block matching motion estimation algorithm utilizing the information on spatial distribution of local motion vectors. Both the search center and the search strategy for each macroblock are determined based on the cityblock lengths of the neighborhood motion vectors so as to increase the accuracy of motion estimation and achieve high computational gain. The simulations are carried out using MPEG-4 Verification Model software. Experimental results demonstrate that the proposed algorithm achieves the objective and subjective quality close to that provided by full search and at the same time offers a substantially higher search speed.

1. INTRODUCTION

Motion estimation and compensation are employed in video compression to reduce temporal redundancy between successive frames in a video sequence and thereby achieve high compression. Among all motion estimation techniques, block matching algorithm (BMA) has been adopted by most of the video coding standards including recently established MPEG-4 standard [1]. In the conventional BMA called full search, a block of pixels in the current frame is compared with the reference blocks within a search window of the previous frame for every possible displacement. Consequently full search requires a large number of computations. To alleviate the computational burden, a number of fast search algorithms have been proposed in the past [2].

A higher search speed is obtained by making use of the spatial correlation property of motion vectors. The main aim is to select a point called *search center* close to the global minimum on the error surface and then obtain the motion vector by performing local search around the search center according to a search strategy. The search center may either be predicted from

the neighborhood motion vectors [3,4] or obtained by choosing the best vector from the set containing the motion vectors of the adjacent blocks [5,6]. The best vector is the one which gives minimum matching error. Thus the search center is always a function of neighborhood motion vectors. In other words, the algorithms belonging to this family of fast motion estimation schemes assume that these motion vectors are always spatially correlated. However, this is not always true because in general both high and low motion activity regions are often present in the same frame of a typical video sequence.

We propose *motion vector field adaptive search technique* (MVFAST) which effectively utilizes the information obtained from motion vectors of adjacent blocks for efficient motion estimation. In MVFAST, the motion activity at the current block position is first determined using the distribution of motion vectors of adjacent blocks [4]. The motion activity is categorized as *low*, *medium* or *high* motion. The selection of search center and search strategy is adaptive to the motion activity.

2. MOTION VECTOR FIELD ADAPTIVE SEARCH TECHNIQUE (MVFAST)

A luminance frame is subdivided as a raster of nonoverlapping blocks with size of 16×16 pixels each, called *macroblocks* (MBs). The MBs are processed in the raster-scan order for motion estimation. The two components (i.e., horizontal and vertical coordinates) of the motion vector of each MB are coded differentially with respect to the corresponding components of the predicted vector. In MPEG-4, the predicted vector is the *median* of the three spatial neighborhood motion vectors, that are already determined. In the proposed algorithm, we make use of the distribution of the same three neighborhood motion vectors for the dual purposes—increasing the search speed and reducing the

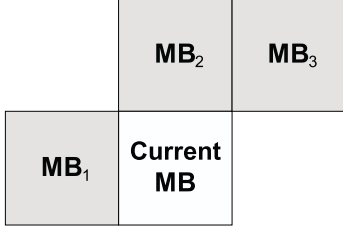


Figure 1: Region of support for the current MB consists of MB₁, MB₂ and MB₃.

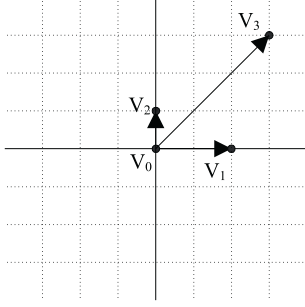


Figure 2: Example of distribution of vectors belonging to set V . In this case, $l_{v_1} = 2$, $l_{v_2} = 1$, $l_{v_3} = 6$; thus $L = 6$.

bits needed to code the motion vector.

(a) Determination of local motion activity

A motion vector field is a two-dimensional array of motion vectors of all the MBs in a frame. The *local motion vector field* at a MB position is defined as the set of motion vectors in the region of support (ROS) of that MB. The ROS of a MB includes the spatially adjacent MBs 1-3, as shown in Fig. 1. The local motion vector field at a MB position influences the estimation and coding of motion vector of that MB.

Let $V = \{V_0, V_1, V_2, V_3\}$, where $V_0 = (0, 0)$ and V_i is the motion vector of MB _{i} in the ROS (see Fig. 2). The cityblock length $V_i = (x_i, y_i)$ is defined as

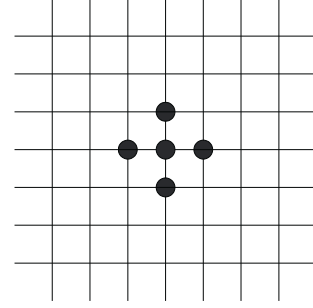
$$l_{v_i} = |x_i| + |y_i|. \quad (1)$$

Let

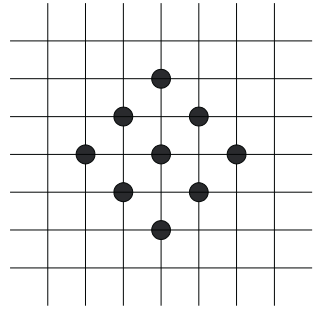
$$L = \text{MAX}\{l_{v_1}, l_{v_2}, l_{v_3}\}. \quad (2)$$

Thus the motion activity at the current MB position is defined as follows.

$$\begin{aligned} \text{Motion Activity} &= \text{Low}; \text{ if } L \leq L_1 \\ &= \text{Medium}; \text{ if } L_1 < L \leq L_2 \\ &= \text{High}; \text{ if } L > L_2; \end{aligned} \quad (3)$$



(a) Small Diamond Search Pattern (SDSP)



(b) Large Diamond Search Pattern (LDSP)

Figure 3: Search patterns

where $L_1 (= 1)$ and $L_2 (= 2)$ are the cityblock distance from the center point of the pattern to any other point on the small and large search patterns (see Fig. 3), respectively.

(b) Selection of search center

The choice of the search center depends on the local motion activity at the current MB position. If the motion activity is low or medium, the search center is the origin. Otherwise, the vector belonging to set V that yields the minimum sum of absolute difference (SAD) is chosen as the search center.

(c) Local search

A local search is performed around the search center to obtain the motion vector for the current MB. The search patterns employed for the local search are shown in Fig. 3. Two strategies are proposed for the local search and their choice depends on the motion activity.

ity identified. If the motion activity is low or high, we employ small diamond search (SDS). Otherwise, we choose large diamond search (LDS) [2].

i) *Small diamond search* (SDS):

Step 1: Small diamond search pattern (SDSP) is centered at the search center, and all the checking points of SDSP are tested. If the center position yields the minimum SAD, then the center represents the motion vector (i.e., no motion); otherwise, go to Step 2.

Step 2: The center of SDSP moves to the point where the minimum SAD was obtained in the previous step, and all the points on SDSP are tested. If the center position yields the minimum SAD, then the center represents the motion vector; otherwise, recursively repeat this step.

ii) *Large diamond search* (LDS):

Step 1: Large diamond search pattern (LDSP) is centered at the search center, and all the checking points of LDSP are tested. If the center position gives the minimum SAD, go to Step 3; otherwise, go to Step 2.

Step 2: The center of LDSP moves to the point where the minimum SAD was obtained in the previous step, and all the points on LDSP are tested. If the center position gives the minimum SAD, go to Step 3; otherwise, recursively repeat this step.

Step 3: Switch the search pattern from LDSP to SDSP. The point that yields the minimum SAD, is the final solution of the motion vector.

Table 1 summarizes the methodology for selection of search center and search strategy depending on the motion activity at the current MB position.

Note that the origin is always included as one of the search points in all the cases. Furthermore, a large number of motion vectors in the video sequences (e.g., *talking head* sequences) are equal to (0,0) due to their low-motion content. For such video sequences, the speed of motion estimation can be further increased by detecting *no-motion* activity simply based on the SAD at the origin. Therefore, we suggest an optional phase, called *early elimination of search*, as the first step in motion estimation as follows.

(d) Optional Phase : Early elimination of search

The search is terminated immediately when the SAD obtained at (0,0) is less than a threshold T , and the motion vector is assigned as (0,0). Due to this, a much higher computational gain is achieved at a graceful or unnoticeable degradation of quality, particularly for the video sequences containing low-motion content. Note that this option of early elimination of search can be disabled by imposing $T = 0$.

Motion Activity	Search Center	Local Search Strategy
Low	Origin	SDS
Medium	Origin	LDS
High	The position of the vector in the set V that yields minimum SAD	SDS

Table 1: The decision table for the proposed algorithm

3. EXPERIMENTAL RESULTS

Our proposed MVFAST is embedded in the MPEG-4 version-2 Verification Model (VM) software, and its performance is compared with full search (FS) and diamond search (DS) [2] algorithms that are already available in the VM [1]. For the simulations, the performance is measured in terms of the average value of PSNR for luminance component (Y) and the number of search points. The *computational gain* or *speed-up factor* with a fast motion estimation algorithm is defined as the ratio of the total number of search points required for FS to that required for the fast algorithm. The typical MPEG-4 test video sequences, with 300 frames (rectangular VOP) each in CIF format, are used to examine the performance. The search range of (-16, +15) (i.e., with the search window size 32×32) is considered. The video sequences are encoded at the target bit rate of 112 kbits/sec and target frame rate of 10 frames/sec.

Case-1: Early elimination of search is disabled

Table 2 gives the average value of PSNR (Y) for FS, DS and MVFAST with $T = 0$. We observe that the PSNR of MVFAST is close to the PSNR for FS and better than that for DS. For *Coastguard*, the proposed algorithm gives same PSNR as FS. The comparison of computational gain for DS and MVFAST is shown in Table 3. MVFAST is about two times faster than DS on average.

Case-2: Early elimination of search is enabled

Emperically, it is found that about 98% of the MBs whose SAD at (0,0) is less than 512 have zero motion vector. Hence, we choose $T = 512$ in our algorithm MVFAST and enable the early elimination of search. Comparing with the results obtained in Case-1, we observe that there is slight difference in the PSNR values. The computational gain has remained nearly the same for *Coastguard*, *Foreman* and *Table Tennis* whereas it has increased appreciably for *News*.

4. CONCLUSIONS

The *motion vector field adaptive search technique* (MVFAST) presented in this paper makes use of the knowledge on the distribution of motion vectors in the neighborhood of the current macroblock position to decide the search center and search strategy. Experimental results show its high performance in terms of both average PSNR (on Y component) and computational gain for all types of video sequences. This makes it suitable for real-time video encoding applications.

Sequence	FS	DS	MVFAST
News	34.85	34.78	34.76
Foreman	30.04	29.59	29.92
Coastguard	27.05	26.44	27.05
Table Tennis	30.69	29.90	30.50

Table 2: Average PSNR(Y) values in dB for FS, DS and MVFAST (with early elimination of search disabled; i.e., $T=0$)

Sequence	DS	MVFAST
News	70	149
Foreman	43	82
Coastguard	49	97
Table Tennis	58	119

Table 3: Computational gain achieved with DS and MVFAST (with early elimination of search disabled; i.e., $T=0$)

Sequence	PSNR(Y)	Computational Gain
News	34.75	209
Foreman	29.92	84
Coastguard	27.05	97
Table Tennis	30.49	128

Table 4: Average PSNR (Y) and computational gain for MVFAST (with early elimination of search enabled and $T=512$)

References

[1] ISO/IEC JTC1/SC29/WG11 N2932, "MPEG-4 Video Verification Model version 14.0," pp. 301-303, October 1999.

[2] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," *International Conf. On Information, Communications and Signal Processing (ICICS'97)*, pp. 292-296, September 9-12, 1997.

[3] M. Gallant and F. Kossentini, "An Efficient Computation-constrained Block-based Motion Estimation Algorithm for Low Bit Rate video Coding," *The Thirty-Second Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 467-471, November 1-4, 1998.

[4] Prabhudev I. Hosur and Kai-Kuang Ma, "Fast Block Motion Estimation Based on Prediction of Initial Motion Vectors of Interleaved Blocks Using Spatial Correlation," *4th Asia-Pacific Conference on Communications / 6th Singapore International Conference on Communication Systems*, November 23-27, 1998.

[5] J. Chalidabhongse and C. J. Kuo, "Fast Motion Vector Estimation Using Multiresolution-Spatio-Temporal Correlations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 477-488, June 1997.

[6] B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 2, pp. 148-157, April 1993.