International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:8, 2009

Joint Adaptive Block Matching Search (JABMS) Algorithm

V.K.Ananthashayana and Pushpa.M.K

Abstract**—** In this paper a new Joint Adaptive Block Matching Search (JABMS) algorithm is proposed to generate motion vector and search a best match macro block by classifying the motion vector movement based on prediction error. Diamond Search (DS) algorithm generates high estimation accuracy when motion vector is small and Adaptive Rood Pattern Search (ARPS) algorithm can handle large motion vector but is not very accurate. The proposed JABMS algorithm which is capable of considering both small and large motions gives improved estimation accuracy and the computational cost is reduced by 15.2 times compared with Exhaustive Search (ES) algorithm and is 1.3 times less compared with Diamond search algorithm.

Keywords**—** Adaptive rood pattern search, Block matching, Diamond search, Joint Adaptive search, Motion estimation.

I. Introduction

THE limited channel bandwidth and stringent requirements of real-time video playback, video coding is an indispensable process for many visual communication applications and always requires very high compression ratio. Motion estimation is a key problem in video processing, video compression and computer vision. An effective and popular method to reduce the temporal redundancy, called block matching motion estimation (BMME), has been widely adopted in various video coding standards[1] like H.261, H.263, MPEG1, MPEG2, MPEG4 and H.264. Due to the effectiveness and simplicity for implementation in software/hardware, BMME is used in many motion compensated video Codec. Therefore, fast and accurate block based search technique is highly desirable to assure much reduced processing delay, while maintaining good reconstructed image quality.

In video compression, successive video frames are similar except for changes induced by objects moving within the frames. In the trivial case of zero motion between frames, it is easy for the encoder to efficiently predict the current frame as a duplicate of the prediction frame. When this is done, the information is transmitted to the decoder to reconstruct the picture from the original reference frame. When there is a motion in the images, the displacement of moving objects between successive frames will be estimated (motion estimation) first. The resulting motion information is then exploited in efficient inter-frame predictive coding (motion compensation). Consequently the prediction error is transmitted. The motion information also has to be transmitted to the decoder, which is able to estimate the motion field. An efficient representation of the motion is thus critical in order to reach high performance in video compression.

In the field of motion estimation (ME), many techniques have been proposed [1]-[6]. Basically ME techniques can be broadly classified as: gradient techniques, pixel-recursive techniques, block matching techniques and frequency-domain techniques. Among these four groups, block matching is particularly suitable in video compression schemes based on discrete cosine transform (DCT) such as those adopted by the recent standards H.261, H.263 and MPEG family[7]. The most commonly used block matching algorithm (BMA) is the Full search (FS)/Exhaustive search (ES), which exhaustively searches for the best matching block within the search window. However, FS yields very high computational complexity and makes ME the main bottleneck in real-time video coding applications. In fast BMA using a fixed set of search patterns, the assumption is that, the matching error decreases monotonically as the search moves towards the position of the global minimum error [2] and the error surface is uni-modal as shown in Fig(1). The well known algorithms for this assumption are 2-D logarithmic search (2DLOG)[2], Three step search (TSS)[8], four-step search (4SS)[9], Block

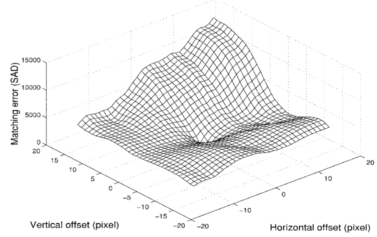


Fig.1 Uni-modal error surface with a global minimum error point.

based gradient descent search (BBGDS)[10] and the Diamond search (DS)[11]. Due to simplicity and regularity, DS algorithm implementation is very simple. However they have less adaptability and search efficiency in tracking large motions. So, Adaptive rood pattern search (ARPS) is proposed in [12] to track large motions, with less number of computations by using zero motion prejudgment (ZMP) for the

V. K. Ananthashayana, Professor and Head, Department of Computer Science and Engineering, Image processing and computer vision (IPCV) lab, M.S.Ramaiah Institute of Technology, Bangalore, India.

Pushpa.M.K, Assistant Professor, Department of Instrumentation Technology, M.S.Ramaiah Institute of Technology, Bangalore, India (e-mail: pushpachandan@rediffmail.com)

reduction in the computation complexity.

In order to achieve accurate motion estimation for large and small displacements, this paper proposes a new JABMS algorithm, to generate both fast and slow motion vectors based on the prediction error. Organization of this paper is as follows: Section 2 discusses Methodology. Section 3 discusses Simulation results and Conclusions are reflected in Section 4.

II. Methodology

In HDTV applications, the movement of camera and images is rather large and it is necessary to use large scale search method to deal with such movements. But as for the field such as video telephone and videoconference, the movement of images is very small. So we need a method which can adapt by itself to different movements for proper compensation with less error and fewer computations.

Hence in this paper we tried to generate a new block matching algorithm, which gives high estimation accuracy disregarding whether the motion vector (MV) is small or large, and the computation cost is not too much by adaptively choosing minimum sum of absolute difference (SAD) from Diamond search and ARPS algorithms.

A. *Diamond Search Algorithm.*

The DS algorithm is summarized as below.

Step 1) The initial large diamond search pattern(LDSP) is centered at the origin of the search window, and the 9 checking points of LDSP are tested. If the calculated minimum block difference (MBD) point is located at the center position, go to Step 3; otherwise, go to Step 2.

Step 2) The previous MBD point is re-positioned as the center point to form a new LDSP. If the new MBD point obtained is located at the center position, go to Step 3; otherwise, recursively repeat this step.

Step 3) Switch the search pattern from LDSP to small diamond search pattern (SDSP). The MBD point found in this step is the final solution of the motion vector which points to the best matching block.

The example for DS algorithm search pattern is as shown in Fig(2). The MBD point in each stage is shown with different color pixel. With DS search algorithm it is possible to estimate the motion vectors which are small. However it fails for large motions.

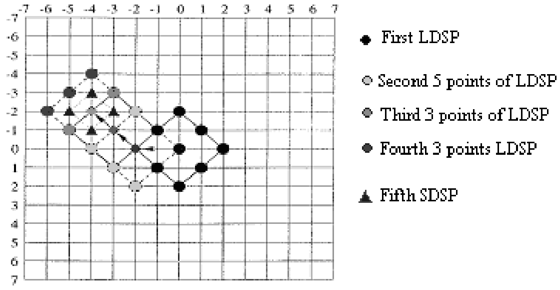


Fig. 2. Search path example which leads to the motion vector(-4,-2) in five search steps-four times of LDSP and one time SDSP at the final step. There are 24 search points in total-taking nine, five, three, three, and four search points at each step, sequentially.

B. *Adaptive Rood Pattern Search Algorithm.*

ARPS algorithm make use of the fact that the general motion in a frame is usually coherent, i.e. if the macroblocks around the current macroblock moved in a particular direction then there is a high probability that the current macroblock will also have a similar motion vector. In order to obtain an accurate motion vector prediction of the current block, two factors need to be considered: 1)choice of the *region of support* (ROS) that consists of the neighboring blocks whose motion vectors will be used to calculate the predicted MV, and 2) Algorithm used for computing the predicted MV.

The ARPS algorithm is summarized as below.

Step 1: Compute the matching error (SAD_{centre}) between the current block and the block at the same location in the reference frame (i.e., the center of the current search window).

If
SAD_{centre} < Threshold(T)
$\overrightarrow{MV}_{target}$= [0 0],
Stop;
else
if
the current block is a leftmost boundary block,
Pattern Size(Γ) =2;
else
Γ=Max {|$MV_{predicted}(x)$, $MV_{predicted}(y)$|}
Go to Step 2**.**

Step 2: Align the center of ARP with the center point of the search window and check its four search points plus the position of the predicted MV to find out the current minimum matching error (MME) point.

Step 3: Set the center point of the unit-size rood pattern (URP) at the MME point found in the previous step and check its points. If the new MME point is not incurred at the center of the current URP, repeat this step; otherwise, the MV is found, corresponding to the MME point identified in this step.

An example for ARPS algorithm search pattern is as shown in Fig(3). Calculating the statistical average of MVs in the region of support (ROS) is a common practice to obtain the predicted MV. We used TypeD ROS in this algorithm. The Mean and median prediction have been tested in these experiments.

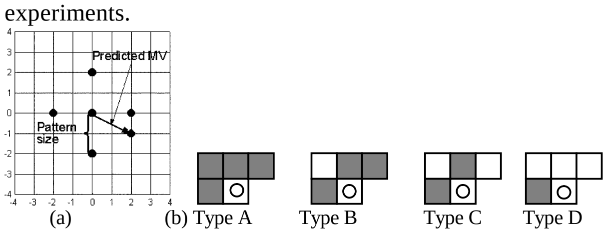


Fig.3. (a) Adaptive Rood Pattern (b) Four types of ROS, depicted by the shaded blocks. The block marked by "o" is the current block.

The main advantage of ARPS algorithm over DS is if the predicted motion vector is (0, 0), Then LDSP is skipped and search is started directly using SDSP. If the predicted motion vector is far away from the centre, then computations are minimized by jumping directly to predicted motion vector vicinity and then perform SDSP. Whereas time consumed by DS is relatively more by doing LDSP.

C. Joint Adaptive block matching search algorithm.

In our proposed algorithm we adaptively choose both DS and ARPS techniques, which are well suited for small and large movements of motion pictures. In each macroblock of size $N \times N$ (i.e. 16×16) pixels in the current frame is compared with shifted regions of the same macroblock from the previous frame(reference) using Diamond search and Adaptive rood pattern search with Zero motion prejudgment (ZMP). To distinguish the static and motion blocks, a technique called Zero motion prejudgment[12] is implemented. The prejudgment is made by first computing the matching errors between the current block and the block at the same location in the reference frame (i.e., the candidate block corresponds to zero-MV) and comparing it with a predetermined threshold, T=512 for large movements. If the matching error is smaller then, the current block will be decided to be a static block without performing the remaining search.

The shift which results in a minimum error (minimum SAD) is selected as the best match for that macroblock. After computing the prediction error, the motion vector which produces smaller error is used as the final motion estimation result. The motion compensated prediction frame is then formed from all the shifted regions from previous decoded frame. The flow diagram of our proposed algorithm is as in Fig (4). In this implementation, a checking bit-map (one bit for each macroblock) has been employed to record whether a search point under checking has already been examined before, so that duplicated checking computation can be avoided.

In most of the algorithms, the "best match" is found using various cost functions like Sum of Absolute Difference(SAD), Mean Absolute Difference(MAD) and Mean Square Error(MSE) as given in the equations (1), (2) and (3) respectively.

$$SAD = \sum_{i=1}^{N}\sum_{j=1}^{N} |C[i,j] - R[i,j]| \quad (1)$$

$$MAD = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} |C[i,j] - R[i,j]| \quad (2)$$

$$MSE = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} (C[i,j] - R[i,j])^2 \quad (3)$$

$$PSNR = 10\log_{10}\frac{(Peak\ to\ peak\ value\ of\ original\ data)^2}{MSE} \quad (4)$$

Where $N \times N$ is the size of the macroblock, $C[i,j]$ and $R[i,j]$ are the pixels being compared in current macroblock and reference macroblock respectively.

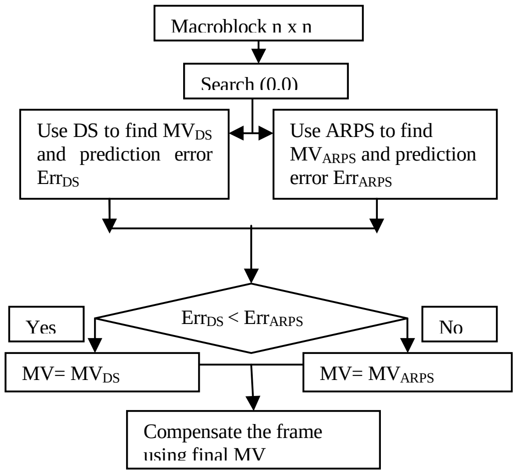


Fig.4 Flow diagram for proposed JABMS algorithm.

In this paper we consider SAD as cost function, and also calculate Peak-Signal to Noise ratio (PSNR) from Eq.4 as quality assessment for compensated frame.

III. Simulation Results

In many visual communication applications such as video telephony, there is little motion between the adjacent frames. Hence, a large percentage of zero-motion blocks are encountered in such type of video sequences. Results of some typical test video sequences are documented in Table 1.

Note that the total number of static blocks per frame could be easily as high as more than 70% except in Foreman and Coastguard. Thus, significant additional reduction in computational cost is possible if we perform a zero-motion prejudgment(ZMP) at the beginning of ME. Further investigations indicate that the average matching errors of these static blocks are much smaller than that of moving blocks.

TABLE I
Average Static Blocks per Frame (as per FS method)

	Video and coding bitrate (Kbps)	Average Static Blocks Percentage per Frame
1	Akiyo (10)	93.29%
2	Container (10)	90.65%
3	Hallmonitor (10)	95.1%
4	Mom & Daughter (24)	80.3%
5	Silence (24)	78.69%
6	News (112)	84.6%
7	Tennis (1024)	70.4%
8	Coastguard (112)	17.2%
9	Foreman (1024)	37.4%

In order to evaluate the performance of Diamond Search, Full search and ARPS algorithm with the proposed JABMS algorithm, this paper consider two test sequences of size 288×352 pixels. One is the "Claire" video sequence, which is composed of small motions before a still background. Another is the "Tennis" video sequence, which is composed of a high speed moving objects. Fig. 5 and 6 shows the frame6 from the sequences. We consider 30 frames from each sequence to calculate Average Prediction error, Average PSNR and

Average computations (search points).

Fig.5 Claire

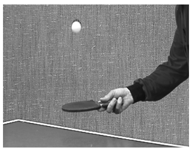
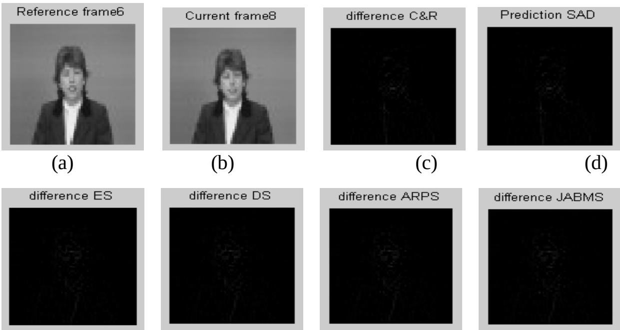


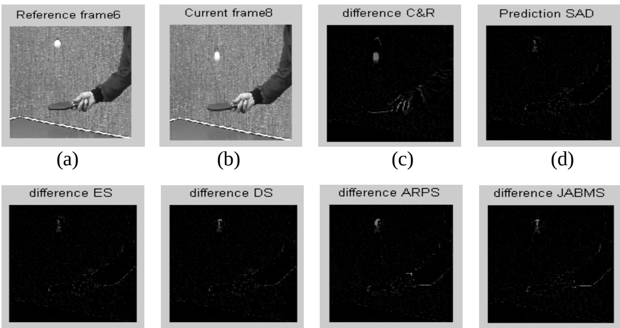
Fig.6 Tennis sequence

From Table.2, The Average SAD of DS, ARPS, ES and the proposed JABMS algorithms for “Claire” and “Tennis” sequences shows clearly that the Average SAD of the proposed JABMS algorithm is nearly as small as the Full search algorithm and the prediction error & differences are shown in Fig.7 and Fig.8 respectively.



(a) (b) (c) (d)
(e) (f) (g) (h)

Fig.7 (a) Reference frame (b) Current frame (c) Difference between Current and Reference frames (d) Prediction error (e)-(g) Difference for existing algorithms (h) Difference for our method.



(a) (b) (c) (d)
(e) (f) (g) (h)

Fig.8 (a) Reference frame (b) Current frame (c) Difference between Current and Reference frames (d) Prediction error (e)-(g) Difference for existing algorithms (h) Difference for our method.

From Table.3, The sequence shows clearly that the prediction error (MSE) of the proposed JABMS algorithm is less than the DS and ARPS search algorithms.

From Table.4, The sequence shows clearly that the Average PSNR of the proposed JABMS algorithm is nearly close to the Full search algorithm and better than DS and ARPS algorithm.

From Table.5, The sequence shows clearly that the Average Computations (Search points) of the proposed JABMS algorithm is less than the Diamond search and Full search algorithm.

TABLE II
COMPARISON OF AVERAGE PREDICTION ERROR (SAD).

Sequences	DS (Err_{DS})	ARPS (Err_{ARPS})	ES (Err_{ES})	JABMS (Err_{JABMS})
Claire	398.63	398.22	392.79	397.8
Tennis	2011.1	2021.2	1933.6	2008.7

TABLE III
COMPARISON OF AVERAGE PREDICTION ERROR (MSE).

Sequences	DS (Err_{DS})	ARPS (Err_{ARPS})	ES (Err_{ES})	JABMS (Err_{JABMS})
Claire	13.93	13.8004	13.381	13.8
Tennis	212.36	218.53	186.64	211.25

TABLE IV
COMPARISON OF AVERAGE PSNR

Sequences	DS	ARPS	ES	JABMS
Claire	34.4617	34.4883	34.5849	34.5025
Tennis	23.2214	23.0862	23.7473	23.2454

TABLE V
COMPARISON OF AVERAGE COMPUTATIONS(SEARCH POINTS).

Sequences	DS	ARPS	ES	JABMS
Claire	16.6303	10.18	191.1033	12.6051
Tennis	14.3077	7.0991	191.1033	12.8302

IV. CONCLUSIONS

The paper shows that an efficient Joint adaptive block matching search algorithm can achieve improved accuracy, the computation cost is reduced by 15.2 times compared to Exhaustive Search, 1.3 times compared to Diamond Search and increased by 0.6 times compared to Adaptive Rood Pattern Search method. Our Average PSNR is very close to Exhaustive Search whereas the Average prediction error is less than Diamond Search and Adaptive Rood Pattern Search algorithms. Our method can be useful regardless of the motion vector being small or large.

REFERENCES

[1] K.R.Rao and J.J Hwang, *Techniques and Standards for Image, Video and Audio Coding.* Englewood Cliffs, NJ: Prentice Hall, (1996)
[2] J.R.Jain and A.K.Jain, “Displacement measurement and its application in interframe image coding”, in IEEE Trans. Commn., Vol. COM-29, pp.1799-1808, Dec 1981.
[3] H.G.Musmann, P.Pirsch and H-J.Grallert, “Advances in picture coding” in Proc. IEEE, Vol.73, No.4, pp.523-548, 1985
[4] V.Bhaskaran and K.Konstantinides, *Image and video compression standards: Algorithms and Architecture*, Kluwer Academic Publishers, 1995
[5] A.Murat Tekalp, *Digital video processing*, Prentice Hall, 1995
[6] Aroh Barjatya, “Block Matching Algorithms for Motion Estimation” DIP 6620 Spring Final project paper, 2004
[7] F.Dufaux and F.Moscheni, “Motion estimation techniques for digital TV: A review and a new contribution”, in Proc. IEEE, Vol.83, No.6, June 1995

[8] T.Koga, K.Linuma, A.Hirano, Y.Lijima and T.Ishiguro, "Motion compensated interframe coding for video conferencing". in Proc. Nat. Telecomn. Conf. pp.G5.3.1-G5.3.5, Nov.29-Dec.3 1981.
[9] L.M.Po and W.C. Ma, "A novel four-step search algorithm for fast block motion estimation". in IEEE Trans. Circuits Syst. Video tech., Vol.6, pp.313-317, June 1996.
[10] L.K.Liu and E.Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding". in IEEE Trans. Circuits Syst. Video tech., Vol.6, pp.419-423, August 1996.
[11] S.Zhu and K.K.Ma, "A new diamond search algorithm for fast block-matching motion estimation". in Proc. 1997 Intl. Conf. Information, Commn. And Signal Processing(ICICS). Vol.1, pp.292-296, Sept9-12 1997
[12] Yao Nie and Kai-Kuang Ma "Adaptive rood pattern search for fast block-matching motion estimation". in IEEE Trans. Image Processing, Vol.11, No.12, pp 1442-1448, Dec 2002.