IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 2, Issue 5 (May. – Jun. 2013), PP 44-50
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Motion estimation Using Color Based Segmentation

Ravindra Kumar Purwar[1], Ritika Sethi[2]

[1]*(University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University, India)*

[2]*(University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University, India)*

Abstract: *In block based motion estimation, each frame is divided into macroblocks and motion estimation is performed macroblock wise. It has high computational cost. In this paper, a block based motion estimation technique using color segmentation has been proposed which has single block for each object and motion estimation based on diamond search technique has been used to compute motion vectors. Experimental results show that the performance of Diamond search technique using color segmentation is better than conventional Diamond Search Technique in terms of PSNR, average number of search points and compression ratio.*

Keywords: *colour based segmentation, diamond search, macro block, motion vector, video compression.*

I. INTRODUCTION

Video compression has nowadays become an essential element in multimedia data storage and transmission. Video compression techniques commonly available remove spatial and temporal redundancy, through intraframe and interframe codings, respectively, where intraframe coding finds redundancy within a frame, while interframe coding finds redundancy between consecutive frames. Motion estimation is used to remove temporal redundancy. Most commonly used motion estimation technique is Block matching interframe motion estimation (BMIME), which has been used widely in video compression standards [1] and now also in motion compensated video compression. This is because it is effective in predicting the motion of an object prudently and is simple to implement.

There may be many objects in a video all in independent motion. In BMIME, every frame is segregated into many blocks, each of same size and their motion is observed. A search window is predefined for each block and a best match for the current block in the frame is searched within the window. The vector of the best matched block w.r.t current block is the motion vector (as shown in Fig. 1).

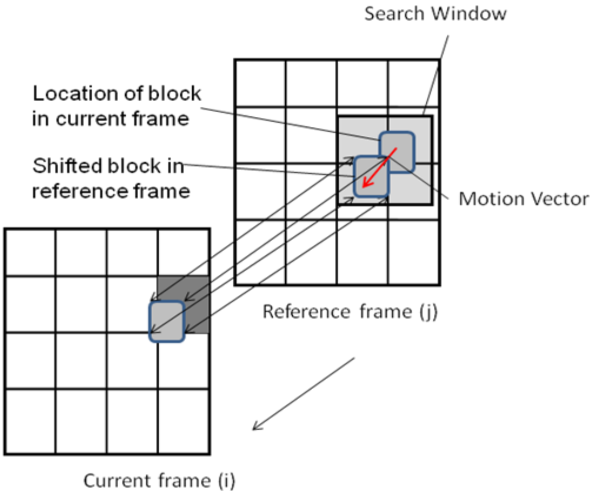

Search Window
Location of block in current frame
Shifted block in reference frame
Motion Vector
Reference frame (j)
Current frame (i)

Fig. 1 The best matched block in reference frame gives the motion vector

There are a number of block based motion estimation algorithms and full search is one which evaluates all points within the search window. The location of the point with minimum error is the motion vector. But, this algorithm searches all possible points and therefore has a high computational cost, and thus is not used in real time applications.

Other BMIME algorithms that reduce this computation cost have been developed, which do not use exhaustive testing of all candidate blocks in a frame. These are 2-D logarithmic search (LOGS) [2], three-step search (TSS) [3], Cross search [4], new three-step search (NTSS) [5], four-step search (4SS) [6], block gradient descent search (BGDS) [7], Diamond Search (DS) [1, 8], Hexagonal Search [9, 10], dynamic pattern search [12], etc.

In all these algorithms, each frame is divided into equal size blocks and motion estimation is performed block by block. If an object expands over multiple blocks, its motion information is described by multiple motion vectors, one for each block. It increases computation cost. In this paper, a color based segmentation of each object is first performed. Using this segmentation, a block can be defined by a single block whose motion estimation is done only once, reducing cost computation. Most importantly this algorithm gives highly reduced computational cost, the reduction of average number of search points by 99% while an increase in the PSNR value by 4.5% in comparison with the conventional diamond search algorithm. The quality/compression ratio has also increased by approximately 4% than the conventional DS algorithm.

The proposed algorithm is given in Section II. Section III presents the analysis of the new algorithm, experimental results of the comparison of this algorithm with conventional DS algorithm is given in section IV. Conclusion is materialized in section V.

II. PROPOSED COLOR SEGMENTATION BASED MOTION ESTIMATION ALGORITHM

The proposed algorithm consists of two basic steps:

(i) Identify Object in video using color based segmentation, and determine the minimum size block for each object which contains it.

(ii) Apply centre biased Diamond search algorithm on the block containing the object only and determine the motion vector of the object block for motion estimation.

1. Color Based Segmentation

In the beginning of this motion estimation algorithm, in each frame, thus the desired object block is identified. This is done by color based segmentation. Segmentation is subdividing an image into its constituent regions or object. Different objects in videos may contain different colors and the aim is to identify the color range of the color contained in object concerned, which is clearly a subspace of the total color space. The algorithm takes advantage of RGB color space to identify objects in the video frames. Each color has its own hexadecimal value for Red, Green and Blue True colors in the RGB color space. Keeping this as a constraint the object was identified using connected component theory for multicolored object. The range for each color in the object is specified in the color space, and a separate binary image is obtained for each color. After addition of all these binary images, the object is segregated by labeled components.

The binary image then obtained contains only the object concerned. The dimensions of the object are derived by scanning the frame and identifying pixel values equivalent to zero as a part of object as shown in Fig. 2. A left to right scan of the binary image gives the first pixel of object on top left corner; a right to left scan gives first pixel of object on top right corner. Similarly top to bottom scan gives first pixel of object on top, and a bottom to up scan gives first pixel of object at bottom. The difference of the coordinates gives the width and height of object and hence the sizes of the block containing object.

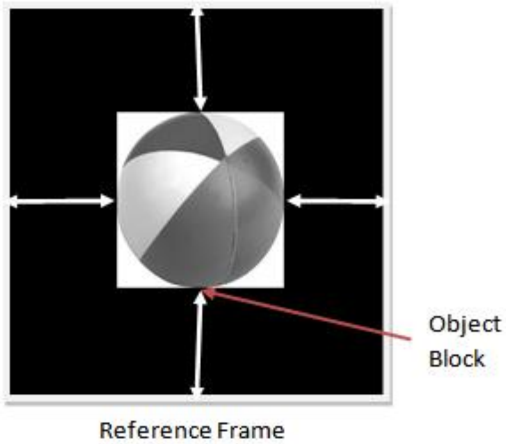


Fig.2 scanning the frame from all sides to obtain the size of block containing object

2. Motion Estimation Using Diamond Search

Motion estimation must be done after object identification. Motion vector is computed using diamond search technique.

The DS algorithm employs two search patterns. The first pattern, called large diamond search pattern

(LDSP), comprises nine checking points from which eight points surround the center one and form diamond shape around it. The second pattern consisting of five checking points forms a smaller diamond shape, called small diamond search pattern (SDSP) [1]. In the searching procedure, LDSP is repeatedly used until the step where minimum block distortion (MBD) occurs at the center point. The search pattern is finally switched from LDSP to SDSP to reaching the final stage. Among the five checking points in SDSP, the position giving the MBD provides the motion vector of the best matching block. This step is repeated for all blocks in the frame, to obtain different motion vectors for all blocks in it.

The block dimensions are obtained by color based segmentation and determination of size of the object block. As already mentioned The FFS algorithm searches all the points in the search area, which has high cost. Like any other low bit-rate videos, this algorithm also uses a search area of ±7 (or 15×15) search window [11]. The search window is formed around the centroid of the object block as shown in Fig. 3

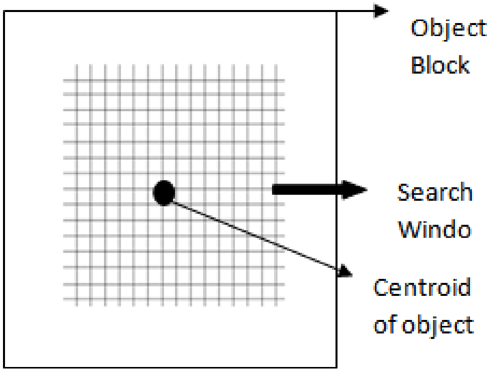


Fig. 3 block containing object showing search window around centroid of object block

Diamond search algorithm follows two designs; the first design, called large diamond search design (LDSD), consists nine candidate points of which eight surround the center to form a diamond shape (◊), while second design consists of five candidate points forming a smaller diamond shape, this is small diamond search design (SDSD). Diamond search in this algorithm begins from the centre of search window, which also forms the centroid of object block, first following the LDSD around centroid. The next step depends on whether the minimum is obtained at the end points or the line points.

(a) If the minima is at the diagonal point, only three new candidate points need to be calculated for minima as in Fig4(a), else,

(b) If minima is at the corner point five new candidate points need to be calculated for minima as shown in fig 4(b).

The candidate points for the second step are chosen keeping in mind maximum possible overlapping, which in turn reduces the average number of search points to be calculated. The main aim of keeping the starting search point at the centre of the block is to reach out as far as possible for covering maximum possibilities of the optimum point, showing the movement. This reduces the chances of being restricted to local minima. The final search step is when either the point reaches boundary of the search window or when minimum of first and second step coincide with each other, this is when SDSD is applied, and only four candidates (shown by ● in Fig. 5)need to be calculated.

The minima among all candidate points in any step are calculated using MAE (Mean Absolute Error), this is found out for the object block of size M×N, which was determined in Section III(A). The upper left corner of block is given by(x, y) in the Current frame (cur), which is estimated using motion vector (vec(x,y)) with the reference frame(ref) :

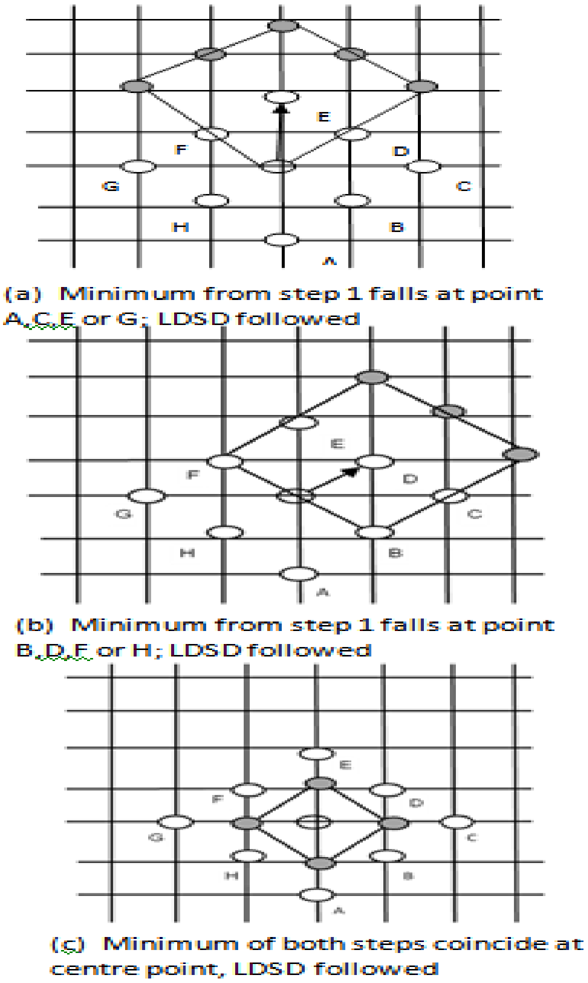
$$MAE_{(xy)}(vec) = \sum^{M} \sum^{N} (cur(x,y) - ref(x + vec(x), \quad y + vec(y)) \tag{1}$$

The search for minima in the search window for the object block can be completed by recursive application of LDSD algorithm.

The motion estimation algorithm using color based segmentation is given as:

1. Identify the object concerned using RGB color space and application of color based segmentation.
2. Determine the dimensions of object and its centroid.
3. Calculate motion vector by diamond search algorithm on the object block

a) The algorithm starts by placement of LDSD at (0, 0), centre of search window, centroid of object block. Calculate the MAE for all nine candidate points specified as ‘1’ in Fig. 5. If MAE is minimum at centre of diamond, follow (c), else follow (b).
b) If MAE is minimum at any of the candidates other than centre, follow LDSD (as in Fig. 4(a), (b)), else MAE is at centre, so, follow (c). Repeat this step.
c) Switch from LDSD to SDSD as in Fig. 4(c). The four internal candidates around the minima obtained in previous step are evaluated.



(a) Minimum from step 1 falls at point A,C,E or G; LDSD followed

(b) Minimum from step 1 falls at point B,D,F or H; LDSD followed

(c) Minimum of both steps coincide at centre point, LDSD followed

Fig. 4 Diamond search design (a) when minimum at end point (b) when minimum at line point (c) Final step

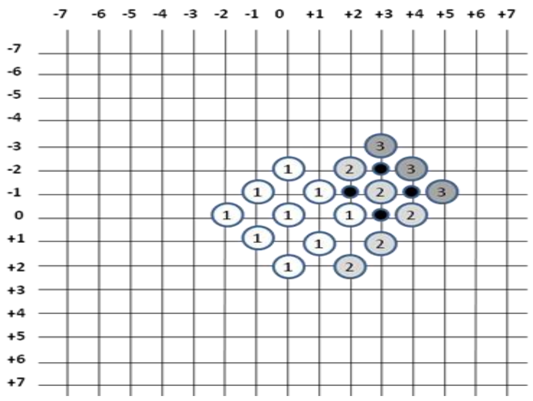


Fig. 5 A search path at centre of the object block. Minimum error at step 3 coincides with step 2; a small diamond search gives the final minimum error at centre of step 2 large diamond.

III. ANALYSIS OF PROPOSED ALGORITHM

We particularly compare this algorithm with DS algorithm. This comparison is done using MATLAB Version 7.4.0.287 (R2007a) using Windows XP Platform, Intel dual Core processor with clock speed 2 GHz and 1024 MB RAM.

First, color based segmentation provides easy, simple and effective segmentation of object. Compared to other ways of identification of objects, this method particularly gives optimal boundary of object, combined with connected component. The color of object can distinguish it from the background as compared to other criterion.

Secondly, the algorithm applies DS on only one block for an object rather than multiple blocks as in conventional DS motion estimation technique. This block is special as it contains the object, whose motion is to be monitored for motion estimation. In comparison to conventional DS the proposed algorithm greatly reduces computation cost.

Third, it is advantageous over other motion estimation algorithms as it includes overlapping of candidate points, reducing the computation even more. Also the search begins from centre of the search window and not from corner of the block as in other BMIME algorithms and DS algorithm. This gives maximum coverage of finding optimum minimum search point as search is centre biased and follows diamond pattern.

PSNR value, Number of search points and compression ratio are taken as the performance parameters for proposed algorithm. To compute the PSNR, the block first calculates the mean-squared error using the following equation:

$$MSE = \frac{\sum_{M,N} \quad [Cur(m,n) - est(m,n)]^2}{M \times N} \qquad (2)$$

In (2), *M* and *N* are the number of rows and columns in Current frame *(cur (m,n)* and estimated frame*(est (m,n))*. Then the block computes the PSNR using the following equation:

$$PSNR = 10 \log_{10}(\frac{R^2}{MSE}) \qquad (3)$$

In (3), *R* is the maximum fluctuation in the input frame data type. The images used for PSNR calculation are grayscale, thus it has an 8-bit unsigned integer data type, and R is 8^3=255 here. The unit of PSNR in this case is decibels (dB).

IV. EXPERIMENTAL RESULTS

All the theoretical assumptions cannot be followed in real time, and do not give the expected results. Thus, it is necessary to actually perform the algorithms on a tool for actual observations and results. In all experiments, motion vector, average search points, all the blocks contained in frame are considered for DS algorithm. The block size is taken as 16×16 for all blocks in DS, thus motion vectors are also many according to the number of blocks. However in new algorithm A single motion vector is generated for a single current frame, average search points exist only for the block containing object, and window size in both the algorithms is taken ±7.

For complete tests, four video sequences were considered, all with different motion content. We considered 100 frames for each video. First, considering the “box” sequence, this is a body in motion sequence, with object moving and a still background. The second sequence, “ball” is that of an object in motion with zooming effects. The video sequence, “purse” is a video with still object and moving background.

Fig. 6 summarizes the frame wise comparison of PSNR value, Average Search point numbers and compression ratio after applying DS and New Motion estimation algorithm using Color Based Segmentation on the video sequence “person”. Since the number of search points in conventional DS comes out to be much more than proposed algorithm, i.e., in range of 500-5000 search points as compared to only 10-40 in case of proposed algorithm, the graph is plotted as log (search points) + c) on y axis, and number of frames on x axis, where c is a constant (here c=2).

Table I gives the average PNSR value for different videos observed over the two motion estimation algorithms. The first column enlists all the videos that have been experimented upon. Second and third column present values for respective algorithm for all video sequences.

Table II summarizes average number of search points in each video sequence. Clearly, new motion estimation algorithm outperforms the conventional DS algorithm. This is obvious as DS finds average search points in a frame by dividing the object block into further smaller blocks of size 16×16, each block having its own average number, which is not the case in the new motion estimation algorithm. Thus, new algorithm has a faster search speed, and since it has to search very less number of candidate points, it also incurs low computational cost.

The question that arises is - does proposed color segmentation based motion estimation algorithm compromise with the quality of compression and error detection in lieu of faster search and low computational cost? Column 2 and 3 of Table III already gives the answer; the quality in terms of compression ratio is not compromised, but has improved by 4% as stated earlier. Also error detection and distortion has improved by 4.5%, as seen in Table I, where all video sequences show higher PSNR value. Also the estimated frames obtained are much better than that obtained in conventional DS algorithm.

TABLE I
AVERAGE PSNR VALUE FOR VIDEOS

Video data	DS	Color based Motion Estimation
Ball	36.2114	37.0697
Box	21.7174	24.2041
Person	26.8747	28.2168
Purse	22.8872	22.9790
Average	26.9227	28.1174

TABLE II
AVERAGE NUMBER OF SEARCH POINTS FOR FRAMES OF VIDEO

Video data	DS	Color based Motion Estimation
Ball	509.56	22.85
Box	5098.46	22.4
Person	4213.98	13.47
Purse	4540.94	10.96
Average	3590.735	17.42

TABLE III
AVERAGE COMPRESSION RATIO FOR VIDEO

Video data	DS	Color based Motion Estimation
Ball	4.7047	4.8718
Box	3.7018	3.8822
Person	3.4149	3.6339
Purse	2.7944	2.8233
Average	3.654	3.803

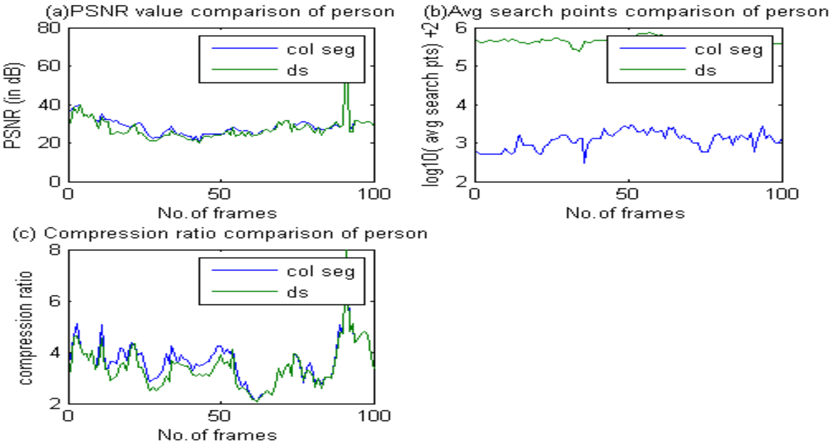


Fig. 6 Comparison of motion estimation algorithm with conventional DS algorithm in terms of (a) PSNR (b) Average Number Of Search Points (c) Compression ratio for video sequence "Person"

V. CONCLUSION

This paper gives the new Motion Estimation Algorithm using Color based Segmentation. Since, it is derived from the conventional DS algorithm, so many search strategies of already existing BMIME algorithms were reviewed along with the DS algorithm. Two criteria were used to compare existing DS algorithm and proposed algorithm along with quality criterion compression ratio. This motion estimation algorithm was designed using color based segmentation and a very realistic motion estimation algorithm, DS, thus it includes the advantages of both.

The new algorithm is efficient and easy in comparison to Conventional DS algorithm. It is efficient as it has reduced the searching time and incurred much lower computational cost, average number of search points which is reduced by nearly 99%. It is easy and simple as only a single block, out of many blocks in the frame needs to be computed for motion vector and search points.

Acknowledgements

The authors would like to thank all anonymous reviewers who provided their valuable reviews and suggestions for improvement of this paper.

REFERENCES

[1]. S. Zhu and K.K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *IEEE Transactions On Image Processing,* Vol. *9,* 287-290, 2000

[2]. J. Jain and A. Jain, "Displacement measurement and its application in Interframe image coding," *IEEE Trans. Commun.*, vol. COMM-29, pp. 1799–1808, Dec. 1981.

[3]. T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov. 29–Dec. 3 1981, pp.G5.3.1–5.3.5.

[4]. M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, pp. 950–953, July 1990.

[5]. R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438–442, Aug. 1994.

[6]. L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 313–317, June 1996.

[7]. L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 419–423, Aug. 1996.

[8]. S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *Proc. Int. Conf. Inform., Commun.,Signal Process.*, Singapore, Sept. 9–12, 1997, pp. 292–296.

[9]. C.Zhu, X. Lin and L. P. Chau, "Hexagon based search pattern for fast block motion estimation," in *IEEE Transactions in circuit systems and video technology,* Vol 12, 349-355, 2002.

[10]. C.Zhu, X. Lin, L. P. Chau,. and L. M. Po, "Enhanced hexagonal search for fast block motion estimation," in *IEEE Transactions in circuit systems and video technology,* Vol 14, 1210-1214, 2004.

[11]. Y.C. Lin and S.C Tai, "Fast Full-Search Block-Matching Algorithm for Motion Compensated Video Compression," in *IEEE Transactions On Communications,* Vol 45, No 5, 1997

[12]. R.K. Purwar and N. Rajpal,"A Fast Block motion estimation algorithm using Dynamic pattern Search" in *Springer-Verlag Journal Signal, Image and Video processing,* Vol 7, Issue 1, pp 151-161, Jan. 2013.