



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

REVIEW AND COMPARATIVE STUDY OF MOTION ESTIMATION TECHNIQUES TO REDUCE COMPLEXITY IN VIDEO COMPRESSION

Ms. Bhavina Patel¹, Dr.R.V.Kshirsagar², Dr.Vilas Nitnaware³

Assistant professor, Dept. of I&C., Sarvajanic College of Engg. & Tech., Surat, India 1

Vice Principal, Dept. of ECE, Priyadarshini College of Engineering College, Nagpur, India 2

Technical Director of Project Team, Liny InfoTop Network, Shandong Province, China, India 3

ABSTRACT: Block matching motion estimation is a key Component in video compression because of its high computational complexity. The process of motion estimation has become a bottleneck problem in many video applications. Typical applications include HDTV, multimedia communications, video conferencing, etc. Motion estimation is a useful in estimating the motion of any object. Motion estimation has been conventionally used in the application of video encoding but nowadays researchers from various fields other than video encoding are turning towards motion estimation to solve various real life problems in their respective fields. In this paper, we present a review of block matching based motion estimation algorithms, reduced complexity of motion estimation techniques and a comparative study across all different algorithms. Also the aim of this study is to provide the reader with a feel of the relative performance of the algorithms, with particular attention to the important trade-off between computational complexity, prediction quality, result quality and other various applications.

Keywords: Fixed size block motion estimation (FSBME), Block-based motion estimation (BMME), Peak-Signal-to-Noise-Ratio (PSNR), Hybrid block matching algorithm (HBMA).

I.INTRODUCTION

Motion compensated transform coding forms the basis of the existing video compression Standards H.26 1/H.262 and MPEG- 1 /MPEG-2, where the compression algorithm tries to exploit the temporal and spatial redundancies by using some form of motion compensation followed by a transform coding, respectively. The key step in removing temporal redundancy is the motion estimation where a motion Vector is predicted between the current frame and a reference frame. Following the motion estimation, a Motion compensation stage is applied to obtain the residual image, i.e. the pixel differences between the current frame and a reference frame. Later this residual is compressed using transform coding or a combination of transform and entropy coding. The above Video compression standards employs block motion estimation techniques. The main advantages of FSBME (fixed size block motion estimation) are simplicity of the algorithm and the fact that no segmentation information needs to be transmitted [1]-[2]-[3].

In block motion compensated video coding; first image frames are divided into square blocks (FIXED SIZE). The next step is to apply a three-step procedure, consisting of Motion Detection, Motion Estimation and Motion Compensation. Motion detection is used for classifying blocks as moving or non-moving based on a predefined distance or similarity measure. This similarity measure is usually done by MSE (minimum mean square error) criteria or minimum SAD (sum of absolute different) criteria. The output of the motion-estimation algorithm comprises the motion vector for each block, and the pixel value differences between the blocks in the current frame and the “matched” blocks in the reference frame. We call this difference signal the motion compensation error, or simply block error. Many techniques have been proposed for motion estimation for video compression so far. All the methods are proposed keeping any one or more of the three directions aimed that 1.reducing computational complexity 2.representing true motion (proving good quality) 3.reducing bit rate(high compression ratio).

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

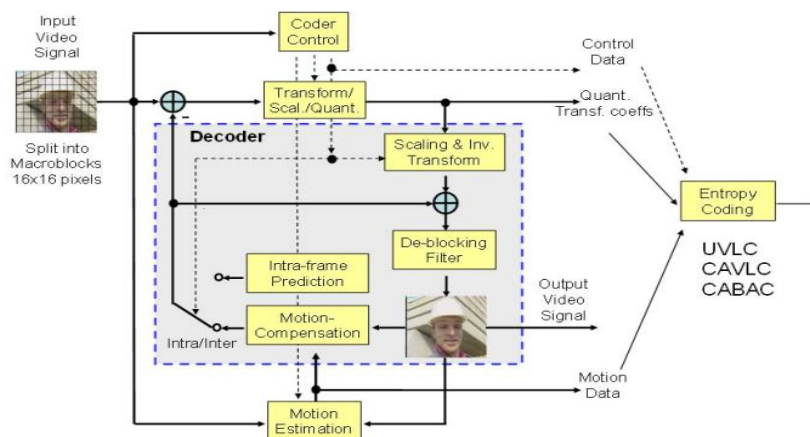


Fig. 1 H.264 video compression process

The key to high performance of video compression lies in an efficient reduction of the temporal redundancy. For this purpose, the block-based motion estimation (BBME) technique has been successfully applied in the video compression standards from H.261 to H.264. The most straightforward BBME method must be full search algorithm (FSA) that searches every candidate position within the search range. Since FS consumes extremely high computational cost, development and refinement on ME algorithms have been feled to archive better trade-off between the computational cost and the ME speed.

The various block matching algorithms are Exhaustive Search (ES), Three Step Search (TSS), New Three Step Search (NTSS), Four Step Search (4SS), Diamond Search (DS), and Adaptive Rood Pattern Search (ARPS), umhexagons, Optical flow, Variable block size, Hierarchical block matching.

In this paper, we present a review of the various block matching based motion estimation algorithms. In Section II explains block matching in general and then the above algorithms in detail. Section III True motion or good quality oriented methods. Section IV compares them. Section V is a conclusion. Section VI is a literature survey of the more recent algorithms, followed by summary and references.

SECTION-I

A. Block Matching Algorithms:

Block matching motion estimation (BMME) is the most widely used motion estimation method for video coding. The underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to frame. The idea behind block matching is to divide the current frame into a matrix of 'macro blocks' that are then compared with corresponding block and its adjacent neighbours in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame. The search area for a good macro block match is constrained up to p pixels on all fours sides of the corresponding macro block in previous frame. This ' p ' is called as the search parameter. Larger motions require a larger p and the larger the search parameter the more. Computationally expensive the process of motion estimation process. Usually the macro block is taken as a square of side 16 pixels, and the search parameter p is 7 pixels.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

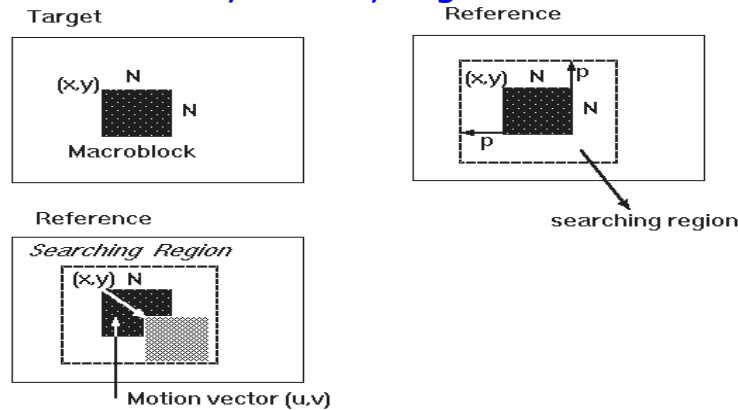


Fig. 2: Block based Motion Estimation process

The idea is represented in Fig 2. The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block. The most intuitive approach for block matching is to use the full search algorithm (FSA). For search reference block, all possible $(2p+1)^2$ blocks are searched to obtain the optimum best match block using a criterion such as MSE, MAE, MAD or SAD which are given as follows:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I(i, j) - K(i, j)|^2 \quad (1)$$

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I(i, j) - K(i, j)| \quad (2)$$

$$SAD(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I(i + u, j + v) - K(i, j)| \quad (3)$$

$$PSNR = 20 \cdot \log_{10}(MAX) - 10 \cdot \log_{10}(MSE) \quad (4)$$

Where N is the side of the macro block, in equation (3) $I(i+u, j+v)$ and $R(i, j)$ are the pixels being compared in current macro block and reference macro block, respectively. Peak-Signal-to-Noise-Ratio (PSNR) given by equation (4) characterizes the motion compensated image that is created by using motion vectors and macro blocks from the reference frame.

SECTION-II

A. LOW COMPUTATION COMPLEXITY OR FAST METHODS:

1. Full Search Method:

This algorithm, also known as Full Search, is the most computationally expensive block matching algorithm of all. This algorithm calculates the cost function at each possible location in the search window. As a result of which it finds the best possible match and gives the highest PSNR amongst any block matching algorithm. Fast block matching algorithms try to achieve the same PSNR doing as little computation as possible. The obvious disadvantage to ES is that the larger the search window gets the more computations it requires 169 iteration if $p=6$ and it requires 289 iteration if $p=8$.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

B. REGTANGULAR PATTERN BLOCK BASED FAST SEARCH ALGORITHM

1. Three Step Search Method:

[4] This algorithm is based on a coarse-to-fine approach with logarithmic decreasing in step size as shown. The initial step size is half of the maximum motion displacement d . For each step, nine checking points are matched and the minimum BDM point of that step is chosen as the starting center of the next step. For $d = 7$, the number of checking points required is $(9 + 8 + 8) = 25$. For larger search window (i.e. larger d), 3SS can be easily extended to n -steps using the same searching strategy with the number of checking points required equals to $[1 + 8 \log_2(d + 1)]$.

2. New Three Step Search Method:

New Three Step Search Method NTSS [5] improves on TSS results by providing a center biased searching scheme and having provisions for half way stop to reduce computational cost. It was one of the first widely accepted fast algorithms and frequently used for implementing earlier standards like MPEG 1 and H.261. The TSS uses a uniformly allocated checking pattern for motion detection and is prone to missing small motions. Figure shows two search paths with $d = 7$. The centre path shows the case of searching small motion. In this case, the minimum BDM point of the first step is one of the 8 neighbour checking points. The search is halfway-stopped with matching three more neighbour checking points of the first step's minimum BDM point. The number of checking points required is $(17 + 3) = 20$. The motion vector is $(0, 2)$.

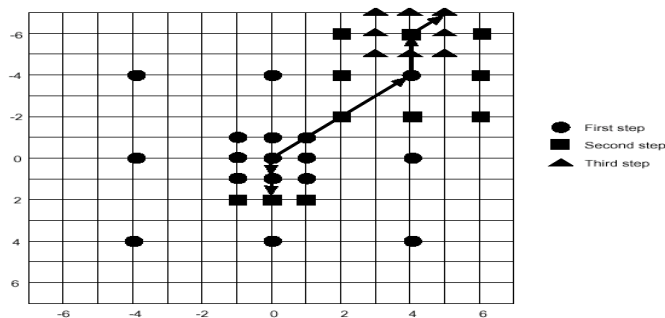


Fig. 3: New 3 stapes search method. The motion vector is $(4,-6)$

The upper-right path shows the case of searching large motion. In this case, the minimum BDM point of the first step is one of the outer eight checking points. Then the searching procedures precede the same as the 3SS algorithm. The number of checking points required is $(17 + 8 + 8) = 33$. The motion vector is $(4,-6)$

3. 4step Search Algorithm:

[6] This algorithm also exploits the centre-biased characteristics of the real world video sequences by using a smaller initial step size compared with 3SS. The initial step size is fourth of the maximum motion displacement d (i.e. $d/4$). Due to the smaller initial step size, the 4SS algorithm needs four searching steps to reach the boundary of a search window with $d = 7$. Same as the small motion case in the N3SS algorithm, the 4SS algorithm also uses a half-way-stop technique in its second and third step's search. Figure shows two search paths of 4SS for searching large motion. For the lower-left path, it requires $(9+5+3+8) = 25$ checking points. For the upper-right path, it requires $(9+5+5+8) = 27$ checking points that is the worst case of the algorithm for $d = 7$.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

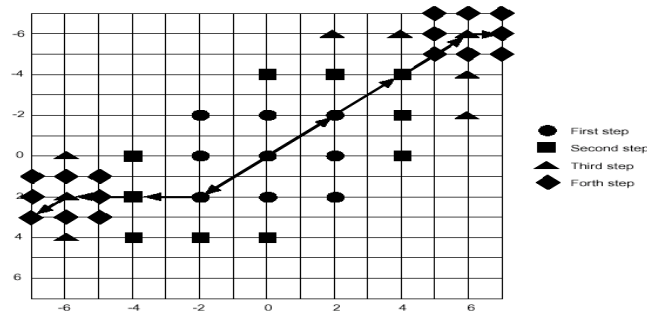


Fig. 4:4 steps search method. The motion vector (6,-6) for upper –right path and (-6, 2) for the lower- path

C. NON RECTANGULAR PATTERN BLOCK BASED MOTION SEARCH ALGORITHMS

1. Diamond& hexagonal Search method:

Many Non rectangular block based motion search algorithms are proposed such as the diamond search (DS) [7]–[8] and hexagon-based search (HEXBS) [9] algorithm's [7] algorithm is exactly the same as 4SS, but the search point pattern is changed from a square to a diamond, and there is no limit on the number of steps that the algorithm can take. DS uses two different types of fixed patterns, one is Large Diamond Search Pattern (LDSP) and the other is Small Diamond Search Pattern (SDSP). These two patterns and the DS procedure are illustrated in Fig.5.

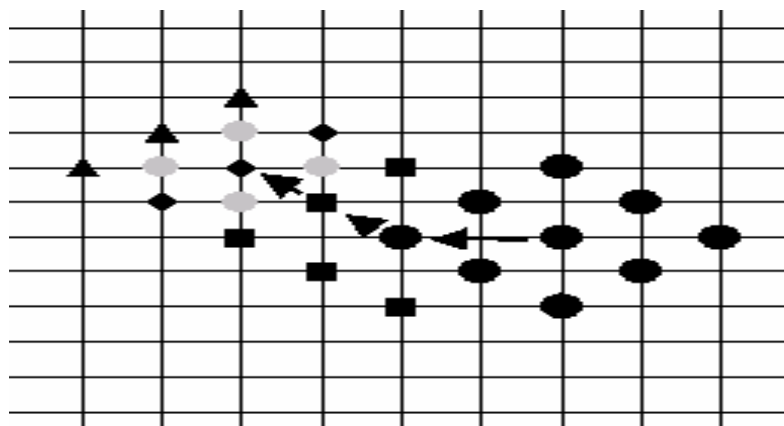


Fig. 5: LDSP & SDSP
 ●LDSP1 ■LDSP2 ◆ LDSP3
 ▲ LDSP4 ○ SPSD1.The MV is (-4, 2).

Just like in FSS, the first step uses LDSP and if the least weight is at the centre location we jump to fourth step. The consequent steps, except the last step, are also similar and use LDSP, but the number of points where cost function is checked are either 3 or 5 and are illustrated in second and third steps of procedure shown in Fig5. The last step uses SDSP around the new search origin and the location with the least weight is the best match. As the search pattern is neither too small nor too big and the fact that there is no limit to the number of steps, this algorithm can find global minimum very accurately. The end result should see a PSNR close to that of ES while computational expense should be significantly less.

The hexagonal search or hexagon-based search (HEXBS) [10]-[11]-[12] has shown the significant improvement over other fast algorithms such as DS .In contrast with the DS that uses a diamond search pattern, the HEXBS adopts a

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

hexagonal search pattern to achieve faster processing due to fewer search points being evaluated. Two novel cross-diamond-hexagonal search (CDHS) algorithms, which differ from each other by their sizes of hexagonal search patterns, are proposed in [11]. They have claimed that their Experimental results show that the proposed CDHS's perform faster than the diamond search (DS) by about 144% and the cross- diamond search (CDS) by about 73%, whereas similar prediction quality is still maintained.

SECTION III

A. TRUE MOTION OR GOOD QUALITY ORIENTED METHODS

1. Hierarchical Block Matching Method:

The basic idea of hierarchical (multi resolution) block matching is to perform motion estimation at each level successively, starting with the lowest resolution level as shown. The estimate of the motion vector at a lower resolution level is then passed onto the next higher resolution level as an initial estimate. The motion estimation at higher level refine the motion vector of the lower one. At higher levels, relatively smaller search window can be used as it starts with a good initial estimate. For each level, one could use fast BMAs such as 3SS, 4SS and 2DLOG for fast motion estimation. Suppose there is a HBMA with two levels as shown. The lower level is formed by sub-sampling the higher level by a factor of two in both horizontal and vertical directions. One pixel displacement at the lower level corresponds to two pixels displacement at the higher level. That is, the search window size in pixel is fourth of the one at higher level. The HBMA can be applied to video codec with spatial scalability such as MPEG-2 and H.263+ in which the video sequence can be divided into layers of different spatial resolutions.

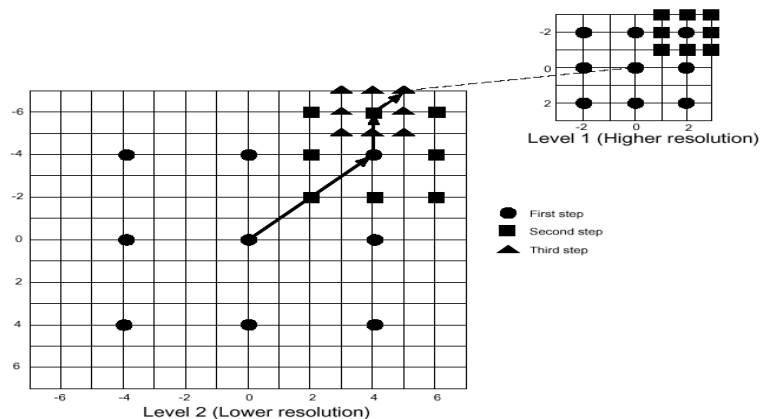


Fig. 6: Hierarchical block matching for multi resolution

This reduces the complexity of calculating BDMs. At higher resolution levels, smaller search ranges are used since motion estimation from a good initial estimate. This reduces the number of locations to be searched. Both factors (i.e. smaller blocks at low resolutions and smaller search ranges at high resolutions) contribute to reducing the overall complexity of the search. Note that when reducing the resolution of the searched frames, the motion speed is also reduced. This makes hierarchical techniques particularly useful for estimating, with reduced complexity, high motion content. Examples of hierarchical motion estimation algorithms are reported in [13].

2. Adaptive Motion Estimation Method:

Adaptive Motion Estimation Method (ARPS) [14] algorithm makes use of the fact that the general motion in a frame is usually coherent, i.e. if the macro blocks around the current macro block moved in a particular direction then there is a high probability that the current macro block will also have a similar motion vector. This algorithm uses the Motion vector of the macro block to its immediate left to predict its own motion vector. An example is shown in Fig. 7. The predicted motion vector points to (3, -2). In addition to checking the location pointed by the predicted motion vector, it also checks at a rood pattern distributed points, as shown in Fig 7.

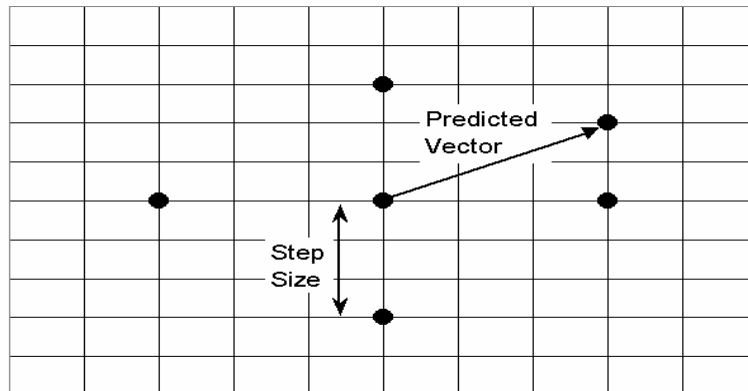


Fig. 7: Adaptive root pattern. The MV is (3, -2)

3. Intern-Block Motion Algorithms:

Another approach to improve the search efficiency is to incorporate the prediction of the inter-block motion information [15]-[16]. Since many blocks in a large homogeneous area of a frame are likely to move in the same direction with similar velocities, the motion vector information between two adjacent reference blocks are highly correlated. Hence it is possible to predict the motion vector of a reference block from the motion vectors of the adjacent blocks.

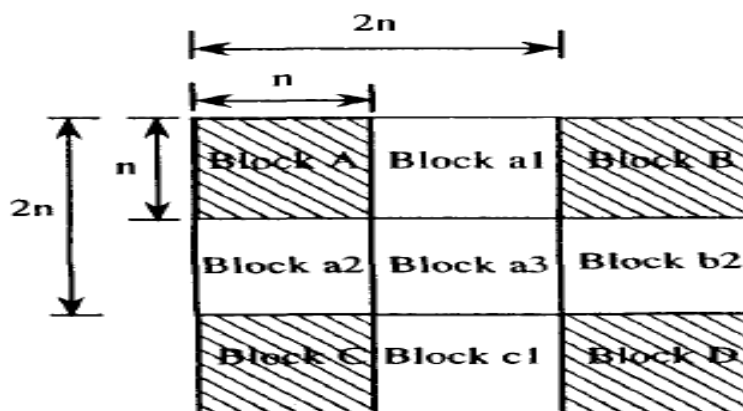


Fig. 8: Illustration of the Inter-Block Motion Prediction

This approach is illustrated in Fig 8 where a $2n$ by $2n$ block is divided into four $n \times n$ blocks (A, a1, a2 and a3). The FSA algorithm is first applied on blocks A, B, C and D. Block a1 is assigned the motion vector of either block A or B depending on whether the motion vector of A or B gives a better match. Likewise, block a2 is assigned the motion vector of either block A or C; and block a3 is assigned the motion vector of either A, B, C or D. Hence the total number of operations is approximately 1/4 of the conventional FSA algorithm. The threshold has a major impact on the PDC criterion, i.e. a smaller threshold value performs better than a larger threshold value for a slow motion sequence while the situation is reversed for a fast motion sequence.

B. VARIABLE SIZE BLOCK MATCHING METHODS

Block Motion Compensation using fixed sized blocks has disadvantages that large blocks may fail to match the actual motion in a sequence, particularly along the moving edges, while small blocks require more overhead information. Most of the time, the objects in motion have different size and shape. Based on this characteristic of motion, in order to overcome the limitation of the traditional fixed-size block motion compensation (FSBMC), several techniques were proposed:

- Variable-Size Block Motion Compensation



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

(VSBMC) [17]- [18] - [19]

- Region-Wise block Motion Compensation (RWMC)

1. Variable Size Block Matching Methods

Achieving, both good video compression ratio and image quality is a conflicting requirement, particularly with fixed-size block matching (FSBM), where the size of all the blocks is the same. The above problem can be overcome by variable size block matching motion estimation technique (VSBME). In variable-size block matching (VSBM) [17]-[18] smaller blocks can be used to describe complex motion while larger blocks can be used in areas where the image content is stationary or undergoing uniform motion while the no of blocks and bit rate are fixed.

Another Quad tree-Structured Variable-Size Block-Matching Motion Estimation [19] presents two algorithms with Minimal Error. The first algorithm computes the optimal selection of variable-sized blocks to provide the best-achievable prediction error under the fixed number of blocks for a Quad tree-based VSBM technique. Although this algorithm is computationally intensive, it does provide a yardstick by which the performance of other more practical VSBM techniques can be measured.

2. Optical Flow Computation Method:

Optical flow estimation may be derived using an inverse finite element approach. It carries out the motion estimation by calculating the gradient, Laplacian, and subsequently the velocities of each pixel in a parallel design which aids in the speed of the computation. Optical flow computation has been proposed as a Pre-processing step for many high-level versions Algorithms. It is an Intensity-based differential technique. Optical flow is calculated by sampling the image intensity at point (x, y) in the image plane at time t ; this is denoted by $u(x, y, t)$. Horn and Schunck [20] initially considered an element of the intensity pattern which has moved a distance of δx and δy in the x and y directions respectively in time δt . They assume that the image intensity of the area remains at a constant value between images.

The Lucas–Kanade method [21] is a widely used differential method for optical flow estimation It assumes that the flow is essentially constant in local neighbourhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion. The Lucas–Kanade method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighbourhood of the point p under consideration. Thus the optical flow equation can be assumed to hold for all pixels within a window centered at p . namely the local image flow (velocity) vector (V_x, V_y) must satisfy the equations (5), (6) and (7).

$$I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1) \quad (5)$$

$$I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2) \quad (6)$$

$$I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n) \quad (7)$$

where q_1, q_2 are the pixels inside the window, and $I_x(q_1), I_y(q_2)$, are the partial derivatives of the image I with respect to position x, y and time t , evaluated at the point q_i and at the current time.

3 .Phase Correlation Method:

[22]These motion estimation techniques operate in the frequency domain and are commonly based on the technique of cyclic correlation. It uses a method by which we can obtain sub pixel motion estimation of high accuracy by using phase correlation. The local estimator is directly applied to signals, without the need to process a complex cross-correlation function, as is done with most of the phase shift estimators. [23] The method is applied to electrography. They perform down sampling of axial and lateral modulated signals leads to very little change in the accuracy and in the spatial resolution. It uses the principle of curve fitting on phase correlation surface and introduction of variable separable fitting using a modified sinc function. Sub-pixel motion estimation is achieved by straightforward extensions to the basic integer-pixel block-matching algorithm mainly through the use of bilinear interpolation. Fitting prototype functions in the vicinity of the maximum peak of the phase correlation [24].



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

C. EFFICIENT LOW COMPUTATION COMPLEXITY VSBME TECHNIQUES

The second algorithm in [19] adopts a heuristic way to select variable - sized square blocks. It relies more on local motion information than on global error optimization. Experiments suggest that the effective use of local information contributes to minimizing the overall error. Heuristic algorithm. is a more computationally efficient VSBM technique than the optimal algorithm but with a comparable prediction error. A fast variable block-size motion estimation algorithm based on merge and split procedures for H.264/MPEG-4 AVC video encoding is proposed in [25].The algorithms take advantage of the correlation of the Motion Vectors (MVs) of the different block-size modes, to achieve good computation reduction. In this method the number of search points can be reduced to about 4% of that using full-search motion estimation, with negligible quality degradation. The problem with VSBM method is its computational complexity where motion vectors and distortions must be calculated for every candidate block.

D. RATE-DISTORTION OPTIMIZED LOW BIT RATE TECHNIQUES:

Low bit rate can be achieved in video coding by the cost of reduced quality (with high distortion). Without losing much the quality several techniques are proposed in rate-distortion optimized manner. Multi-frame prediction [26] and variable block-size motion compensation in a rate-distortion optimized motion estimation and mode selection process are powerful tools to improve the coding efficiency of today's video coding standards like MPEG-4 Visual and ITU-T H.263.

Multi-frame prediction [26] and variable block-size motion compensation in a rate-distortion optimized motion estimation and mode selection process are powerful tools to improve the coding efficiency of today's video coding standards like MPEG-4 Visual and ITU-T H.263. Multi-frame motion compensation is an effective means of improving the quality of the temporal prediction. A Wavelet-Based Video Coding Using Variable Block-Size Motion Compensation and Adaptive Arithmetic Coding is presented as design of a video coder (DVC) that demonstrates how these state-of-the-art coding tools as implemented in the current test model TML- 8 of the ITU-T H.26L, [26]-[27] video compression standardization project can be successfully integrated in a blocking artifact free video coding environment. The temporal prediction is performed by using block motion estimation (BDM) and overlapped block motion compensation (OBMC), where the reference of each predicted block can be obtained from a long-term reference frame memory. A rate-distortion framework to define a very low bit rate coding scheme [19] based on quad tree segmentation and optimized selection of motion estimators. This technique achieves maximum reconstructed image quality under the constraint of a target bit rate for the coding of the vector field and segmentation information. The method to optimize Quad tree object segmentation for hybrid motion estimation for 2D and 3D motion estimation and compensation is presented in the rate distortion sense. This scheme adapts to the depth of the Quad tree and the technique used for motion estimation for each leaf of the tree. A more sophisticated technique, adapted to the requirements of a very low bit rate coder is also proposed which considers also the transmission of the prediction error corresponding to the particular choice of the motion estimator. Based on these coding schemes two versions of a very low bit rate image sequence coder are given.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

SECTION IV

TABLE I Comparisons between Block Matching Methods

Title	Complexity(worst case)	Advantage	Disadvantage
ES	$2*(2P+1)^2$	Best picture quality and highest PSNR	Very high computational cost
TSS	$[1+8\text{Log}_2(p+1)]$	Optimum performance ,less complexity, recommended in MPEG2	can't detect small motion
NTSS	$[1+8\text{Log}_2(p+1)]+8$	More efficient then TSS for small motion	More complex then TSS
4SS	$[18\text{log}_2(p+1)/4+9]$	Initial small step size so more efficient for small MV	More complex then TSS
OSA	$[1+4\text{Log}_2(p+1)]$	More effective in complexity then TSS	Very inefficient in small motion, high MSE error
HEXAGON	$N=7+3*n+4$	More search points as compare to other methods.	Improve speed rate as compare to other methods.
CSA	$[5+4\text{Log}_2(p)]$	Less complex the TSS	Some potential locations are not searched.
CDS	$2*(2*(P+1))$	More efficient in picture quality	More complex
HBMA	Depends on algorithm used	Can be used to get high quality picture with high spatial scalability.	More complex then TSS
BBGDS	9 check point at each step	Works better in small motion	Very large MV for large motion

IV. CONCLUSION

A review of the various block matching based motion estimation algorithms has been presented. These algorithms are classified into three categories, namely, fast algorithms, true motion or good quality oriented methods and low computation complexity VSBME techniques algorithms. The features, advantages and disadvantages of these algorithms have been presented. In addition, we have also briefly reviewed the various matching, criterion such as MAE, MSE, or SAD.

Algorithms are reviewed, in terms of both coding efficiency and computational complexity. Additionally, in order to design a low complexity ME algorithm, the researchers should jointly use the powerful techniques that are listed above to reduce the computational cost in different aspects. When used well together, the new design will significantly reduce the computational cost, as compared with the traditional fast ME algorithms.

REFERENCES

- [1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in inter frame image coding," IEEE Trans. Commun., vol.29, pp. 1799- 1808, Dec. 1981.
- [2] Iain E. G. Richardson, Video Codec Design, West Sussex: John Wiley& Sons Ltd., 2002, Ch. 4, 5, & 6.
- [3] Overview of the H.264/AVC Video Coding Standard, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003.
- [4] Jianhua Lu, and Ming L. Liou, "A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 7, no. 2, pp. 429-433, April 1997
- [5] Renxiang Li, Bing Zeng, and Ming L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 4., no. 4, pp. 438-442, August 1994.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

- [6] Lai-Man Po, and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 6, no. 3, pp. 313-317, June 1996.
- [7] Shan Zhu, and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol 9, no. 2, pp. 287-290, February 2000.
- [8] Yao Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol 11, no. 12, pp. 1442-1448, December 2002
- [9] Chun-Ho Cheung, and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 12., no. 12, pp. 1168-1177, Dec. 2002.
- [10] P. Thambidurai, M. Ezhilarasan, D. Ramachandran, M. Ezhilarasan, "Efficient Motion Estimation Algorithm for Advanced Video Coding" International Conference on Computational Intelligence and Multimedia Applications 2007
- [11] Research and optimization of a H.264 AVC motion estimation algorithm based on a 3G network, 2013 UPPSALA University.
- [12] Xie Liyin, Su Xiuqin, Zhang Shun "A Review of Motion Estimation Algorithms for Video Compression" 2010 International Conference on Computer Application and System Modeling (ICCA SM 2010).
- [13] M. Bierling, "Displacement estimation by hierarchical block matching: in Preceding of the SPIE Ksriual Communications and Image Processing (VCIPJ), 1988, vol.1001, pp. 942-951.
- [14] Yao Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol 11, no. 12, pp. 1442-1448, December 2002.
- [15] B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors", IEEE Trans. on Circuits and Systems for Video Technology, Vol.1.3, No.2, pp.148-157, 1993
- [16] S. Zafar, Y.Q. Zhang and J.S. Baras, "Predictive Block-Matching Motion Estimation for TV Codine -- Part I: Inter-Block Predicti&", IEEE Trans. on Broadcasting: Vol.37, No.3, pp.97-101.1991.
- [17] H. Jelveh and A. Nandi, "Improved variable size block matching motion compensation for video conferencing applications," in Digital Signal Processing, A. Cappellini and A. Constantinides, Eds. Berlin, Germany: Springer-Verlag, 1991
- [18] G.R. Martin, R.A. Packwood and I. Rhee, "Variable Size Block Matching Motion Estimation with Minimal Error, Department of Computer Science, University of Warwick, Oct95 1
- [19] I. Rhee, G. R. Martin, S. Muthukrishnan and R. A. Packwood, "Quad tree-Structured Variable-Size Block- Matching Motion Estimation with Minimal Error," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 10, pp. 42-50, Feb. 2000
- [20] T.A. Browne, J.V. Condell, G. Prasad, T.M. McGinnity "An Investigation into Optical Flow Computation on FPGA Hardware" International Machine Vision and Image Processing Conference
- [21] Francisco Barranco, Matteo Tomasi, Javier Diaz, Mauricio Vanegas and Eduardo Ros "Parallel Architecture for Hierarchical optical Flow Estimation based On FPGA", IEEE Transactions On very large integrated VLSI System.
- [22] Videoprocessing.usssd.edu/~Stanleychan/research/MotionEstimation.html.(webpage)
- [23] Adrian Basarab, Pierre Gueth, Hervé Liebgott, and Philippe Delachartre "Phase-Based Block Matching Applied to Motion Estimation with Unconventional Beam forming Strategies", IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 56, no. 5, May 2009.
- [24] Mr. P. Vijaykumar, Aman Kumar, Sidharth Bhatia "Latest Trends, Applications and Innovations in Motion Estimation Research" International Journal of Scientific & Engineering Research Volume 2, Issue 7, July-2011 ISSN 2229-5518
- [25] Efficient Architecture for Variable block size Motion Estimation of H.264 Video Encoder, 2012 International Conference on Solid-State and Integrated Circuit (ICSIC 2012) IPCSIT vol. 32 (2012) © (2012) IACSIT Press, Singapore
- [26] Ljubomir Jovanov, Aleksandra Pižurica, Stefan Schulte, "Combined Wavelet-Domain and Motion-Compensated Video Denoising Based Video Codec Motion Estimation Methods" IEEE transactions on circuits and systems for video technology, vol.19, no 3, march 2009
- [27] Novel RD-optimization VBSME with Matching Highly Data Re-Usable Hardware Architecture, 2011 IEEE Transaction on Circuits and System for video technology, Vol-21 .