Motion Estimation Algorithms for Baseline Profile of H.264 Video Codec

Dharmendra Jha[#1], Fremin Kannampuzha[#2], Justin Joseph[#3], Stevew Possa[#4],
Dr. Deepak Jayaswal[*5], Santosh Chapaneri[+6]

[#]B.E. Students, [*]Professor, [+]Assistant Professor,
Department of Electronics and Telecommunication Engineering,
St. Francis Institute of Technology, University of Mumbai,
Mumbai, India

***Abstract*— The H.264 video compression standard is the most efficient video compression techniques available today. H.264 encoder can reduce the size of a digital video file by more than 80% compared with the Motion JPEG format and as much as 50% more than with the MPEG-4 Part 2 standard, thus ensuring less network bandwidth and storage space. H.264 supports video compression through motion estimation. Various algorithms have been developed for the process of motion estimation and to meet the requirement of better compression quality and less computational time. The block matching algorithms (BMA) are the simplest method to obtain motion vectors. Full Search (FS) is an optimal search algorithm, but it suffers from high computational time. Motivated by the need for fast and accurate motion estimation, we present sub-optimal and adaptive search techniques for motion estimation, which have good motion estimation time (MET) with fewer number of search points and have almost similar compression quality as that of the full search technique.**

***Keywords*— H.264/AVC, Block Matching Algorithms, Motion Estimation, Optimal Algorithms.**

I. Introduction

With the widespread adoption of technologies such as digital television, Internet streaming video and DVD-Video, video compression has become an essential component of broadcast and entertainment media. Video compression is useful because it helps to reduce the consumption of expensive resources such as data storage on hard disks/servers and transmission bandwidths. Video sequences contain a significant amount of statistical and subjective redundancies within and between frame. The ultimate goal of a video source coding is bit rate reduction for storage and transmission by exploring both statistical (spatial) and subjective (temporal) redundancies. Motion Estimation techniques exploit these unwanted redundancies and achieve video compression [1]. Motion Estimation using block matching techniques is the most widely used method to find motion vector (MV). In this paper, we present three categories of the search algorithms for motion estimation using block matching method: optimal, sub-optimal and adaptive search techniques; alongwith few algorithms of each category.

The optimal algorithm provided the foundation to the development of motion estimation algorithms. However, its high computational cost resulted in the research of algorithms with reduced computations. The development of suboptimal algorithms [3-11] greatly reduced the computational load on the optimal algorithms. The suboptimal algorithms can be classified as follows:

1) *Heuristic Search Technique*: Instead of searching all candidates within search area, it looks for less number of candidate MVs in search area. The choice of the position is driven by some heuristic criterion inorder to find the absolute minimum of cost function [TSS, FSS, and DS].
2) *Fast Exhaustive Search Technique*: In this technique, search points are reduced by removing non-optimal positions without affecting the visual quality.
3) *Hierarchical or Multiresolution Technique*: The MVs are searched for low resolution image and are refined in the normal resolution.
4) *Spatio-Temporal Correlation Technique*: The MVs are selected using the vectors already calculated in the current frame and in the previous frame.

Recent developments have seen that motion estimation through motion vector prediction (MVP) is far more efficient than existing exhaustive search, as well as the sub-optimal search strategies. These algorithms [13-18] provide better PSNR as compared to the suboptimal algorithms.

This paper is organised into four sections. Section II gives the overview of optimal search techniques. The suboptimal search algorithms are explained in section III, while the adaptive search algorithms are discussed in section IV. Section V mentions the comparative overview of optimal, sub-optimal and adaptive search algorithms. Finally, section VI provides the experimented results obtained and the conclusion drawn from them is presented in section VII.

II. Optimal Search Algorithms

The optimal search techniques are the simplest block matching techniques for motion estimation. We have two optimal search techniques, viz. full search (FS) algorithm and successive elimination algorithm (SEA).

A. Full Search Algorithm

Full Search (FS) or Exhaustive Search (ES) algorithm is a brute force method to find motion vectors from the given block. Within the search window, the algorithm starts by identifying the best matching block from the previous frame for a given block in the reference frame in a pixel-by-pixel manner and then estimating the motion vectors for the corresponding best matching block. The algorithm searches all the $(2W+1)^2$ candidate block positions in the search window of size W. This algorithm gives the optimum quality of compression, but it is quite slow and requires more computational time.

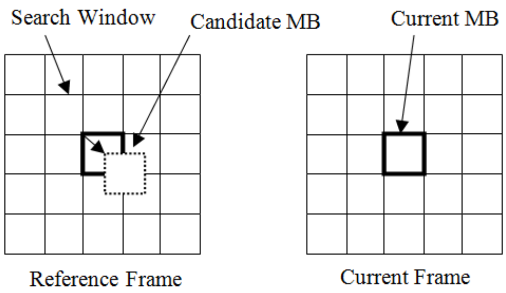

Fig. 1: Full Search Algorithm

B. Successive Elimination Algorithm

Successive elimination algorithm (SEA) [2] is a type of fast full search algorithm. This algorithm reduces the complexity and improves the speed of the search. It uses the Minkowski inequality in order to calculate the best match faster. The Minkowski Inequality states that:

$$| (x_1 + x_2) - (y_1 + y_2) | \leq | (x_1 - y_1) | + | (x_2 - y_2) | \qquad (1)$$

Eq. (1) implies that if the difference between the block sum (summing all pixels value inside a block) of candidate Y and the block sum of reference block X is greater than the minimum sum of absolute difference SAD(X, Y), block Y must not be the best match, since it's SAD must be greater than the minimum SAD(X, Y) based on the inequality. Calculating the block sum difference is much faster than calculating the SAD, thereby increasing the speed of the algorithm.

III. SUB-OPTIMAL SEARCH ALGORITHMS

The sub-optimal search algorithms existing in literature are three-step search (TSS) [3], four-step search (4SS) [5], diamond search (DS) [6], cross-search (CSA) [8] among others. These algorithms use geometric search patterns to determine the motion vector from a given macro-block. The performance of these algorithms depends on the search pattern used and the complexity of video sequence.

A. Three Step Search Algorithm

The three-step search algorithm (TSS) [3], developed by Koga et al, uses a 9 x 9 grid with eight checking points and an initial step size. This algorithm searches for the point having minimum block distortion measure (BDM) among all the eight points at a distance of step size around the center. After every search the center is moved to the point of minimum BDM and the step size is halved. The search terminates when the minimum BDM point happens to be the center, in which case the motion vectors corresponding to the center are the required vectors for the best matching block.

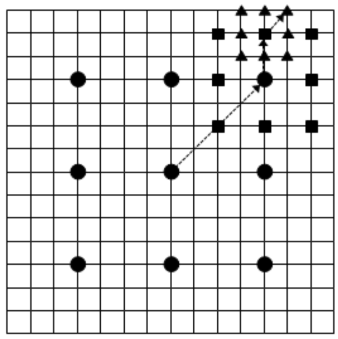

Fig. 2: Search Path for TSS

B. Four Step Search Algorithm

The four-step search algorithm (4SS) was developed by Lai-Man Po and Wing-Chung Ma [5], to resolve the problem encountered by the three-step search algorithm. The algorithm utilizes the center-biased search pattern with a nine checking points in a 5 x 5 window. The center of the search window is then shifted to minimum BDM point. If the minimum BDM point is found at the center of the search window, the search will go to the final step, where a 3 x 3 search window is used to obtain the motion vector corresponding to the minimum BDM point. Otherwise, the search window size is maintained at 5 x 5 and the search pattern is changed based on the location of minimum BDM point on the search window.

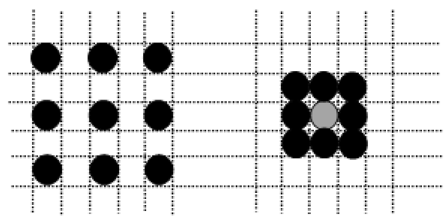

Fig. 3: 5 x 5 and 3 x 3 Grid Search Pattern for 4SS

C. Diamond Search Algorithm

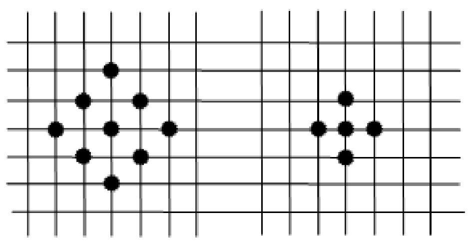

Fig. 4: Diamond Search Patterns (LDSP & SDSP)

The diamond search (DS) algorithm [6] uses two search patterns, viz. large diamond search pattern (LDSP) with nine checking points and small diamond search pattern (SDSP) with five checking points. In the searching procedure, the LDSP pattern is used repeatedly until the minimum BDM point is obtained at the center. Once the minimum BDM occurs at the center of the pattern, the LDSP switches to SDSP. The point with minimum BDM in SDSP provides the motion vector of the best matching block.

IV. Adaptive Search Algorithms

Over the years, developing optimum algorithm for motion estimation has seen major boost in the video coding world. Variations and modifications in sub-optimal techniques were carried out to overcome their shortcomings [4], [7], [9]. New search patterns [10], [11] and new search strategies were developed to meet the requirement of an optimum algorithm. This led to the development of adaptive search techniques, which use predictive methods, threshold limits, stationary block determination and many other features to reduce the complexity of search algorithm.

A well-known adaptive technique called the motion vector field adaptive search technique (MVFAST) had a significant improvement over the sub-optimal techniques. It considered predictors in the initial step and a fixed early-stop criteria combined with search pattern selection method, to enhance the process of motion estimation. The predictive motion vector field adaptive search technique (PMVFAST) further improved upon the MVFAST technique, by considering additional set of predictors and adaptive thresholding. On the same lines, the advanced predictive diamond zonal search (APDZS) was developed, which achieved better quality at an insignificant cost in speed-up. The most efficient algorithms developed from the above ideas are discussed in this section.

A. Unsymmetrical-cross Multi-hexagon Grid Search Algorithm

The unsymmetrical-cross multi-hexagon grid search (UMHexagonS) algorithm [12] was drawn from the basic idea that the movement in the horizontal direction is much heavier than that in the vertical direction and the distribution of motion vectors is zero centered. The algorithm consists of four search patterns: uneven cross search, multi-hexagon grid search, iterative hexagon search and diamond search.

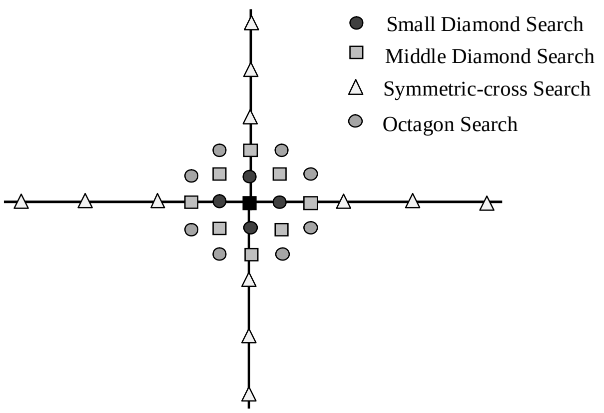

Fig. 5: UMHexagonS Search Pattern – Step 1

The algorithm first predicts the initial search point from the candidate motion vector list. Two threshold limits are defined for early termination scheme. A small diamond search is applied to determine termination based on threshold$_1$, while a middle diamond search is used for termination based on threshold$_2$. Then a symmetric-cross search and an octagon search are applied to find minimum BDM point. Search stops if the minimum BDM found in this step coincides with the minimum BDM found in previous step.

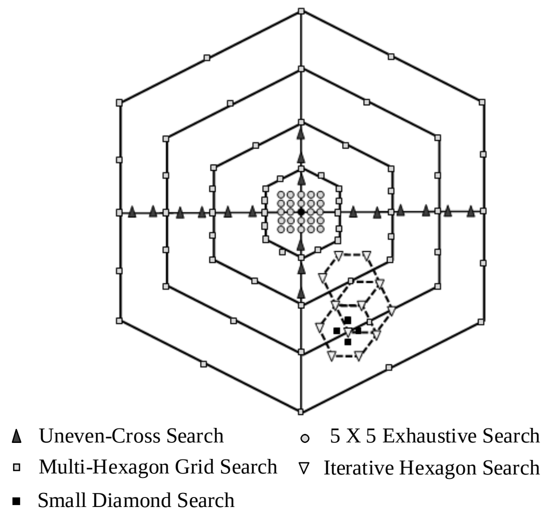

Fig. 6: UMHexagonS Search Pattern – Steps 2 to 4

If the above steps fail, then the algorithm proceeds through an uneven-cross search. The process is continued by a 5 x 5 exhaustive search and multi-hexagon grid search. Finally, the iterative hexagon search and small diamond search is followed around the minimum BDM point to obtain the motion vector. The above sequence of operation is illustrated in Fig. 6.

B. Enhanced Predictive Zonal Search Algorithm

Enhanced predictive zonal search (EPZS) algorithm [15] is an improvement upon predictive motion vector field adaptive search technique (PMVFAST) [17] and advanced predictive diamond zonal search (APDZS) [14]. It considers several other additional predictors in the generalized predictor selection phase of PMVFAST and APDZS algorithms. The algorithm also selects a more robust and efficient adaptive thresholding calculation whereas, due to the high efficiency of the prediction stage, the pattern of the search can be considerably simplified.

There are three important phases in EPZS algorithm. The prediction selection phase is the most important and key feature of EPZS algorithm. The predictors involved in the prediction selection phase are as follows: spatial and temporal adjacent predictors, the median predictor, the (0, 0) motion vector predictor and the accelerator motion vector predictor. EPZS uses an improved adaptive threshold phase, in which it calculates the thresholding parameters by considering minimum SAD of the spatially located three adjacent blocks and the co-located block in the previous frame. The thresholding parameter could in general be seen as:

$$T_k = a_k + \min(MSAD_1, MSAD_2, \ldots\ldots, MSAD_n) + b_k \qquad (2)$$

The search pattern phase in EPZS uses the small diamond or square search pattern as illustrated in Fig. 7.

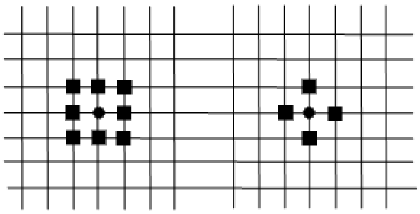

Fig. 7: Square and small diamond pattern used in EPZS

C. *Fast Adaptive Motion Estimation Algorithm*

The fast adaptive motion estimation (FAME) algorithm [18] is an adaptive search technique to determine the motion vector for a given block in current frame. Actually, the FAME algorithm is the advanced version of the motion vector field adaptive search technique (MVFAST) [16] and the predictive motion vector field adaptive search technique (PMVFAST) [17]. The predictor selection phase in FAME algorithm is the same as that of PMVFAST algorithm. While both the search techniques, MVFAST and PMVFAST, use single threshold i.e. threshold for early termination, the FAME algorithm uses two adaptive thresholds, viz. the threshold for stationary block (TSB) and the threshold for half stop (THS), to determine the stationary block in the current frame. The TSB is computed as follows:

$$\begin{aligned} TSB &= \max(SAD_1, SAD_2, SAD_3), \text{ if all adj. MBs have MVs at } (0, 0) \\ &= \min(SAD_1, SAD_2, SAD_3), \text{ otherwise} \end{aligned} \qquad (3)$$

The THS is computed as:

$$\begin{aligned} THS &= \text{mean}(SAD_1, SAD_2, SAD_3), \text{ if } LMA < 4 \\ &= \min(SAD_1, SAD_2, SAD_3), \text{ otherwise} \end{aligned} \qquad (4)$$

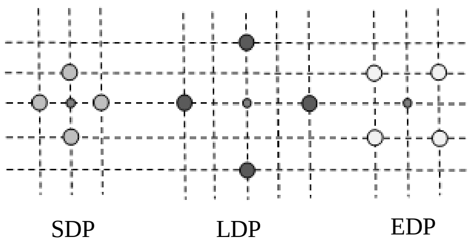

Fig. 8: Various Search Patterns Used in FAME

MVFAST and PMVFAST techniques employs two search patterns: the small diamond pattern (SDP) and the large diamond pattern (LDP). On the other hand, the FAME algorithm employs two more search patterns in addition to the above search patterns: the elastic diamond pattern (EDP) and the motion adaptive pattern (MAP).

Depending on the motion activity, the FAME algorithm chooses the appropriate search center and adaptively changes between the patterns to accelerate the motion estimation process. Also, the FAME algorithm has a limiting factor for every type of motion activity called the elastic factor (k), so that the search does not trap in local or global minima.

V. Analysis of Optimal, Sub-optimal and Adaptive Search Algorithms

The optimal search techniques like full search algorithm gives the best quality compression, but suffer from high computation time. In SEA algorithm, the calculations involve less computation operations, thereby improving the speed of the search. Also, the block sum calculated for the previous frame was reused every time the matching criterion is computed; thereby further reducing the number of computations. However, the disadvantage of fast calculation of block sum method in SEA is that it increases the memory storage requirement.

The sub-optimal techniques did overcome the computation load over the optimal techniques. The performance of three-step search algorithm was better as compared to the full search for video sequences with large motion. However, the TSS algorithm uses a uniformly allocated checking point pattern in the first step, which becomes inefficient for small motion videos. The 4SS reduced the computational complexity of the TSS and its performance was similar to TSS in terms of quality of compression. It was also more robust than the TSS and it maintained its performance for image sequences with complex movements like fast motion. However, 4SS proved ineffective for video sequences with slow motion. The DS performed better than both, the TSS and 4SS, but it encountered loss of visual data in the process.

The adaptive techniques, viz. UMHexagonS, MVFAST, PMVFAST, EPZS, FAME, preserved the visual quality of compressed video, while reducing the compression time. Thus, adaptive search techniques are the evolving methods, which give optimum results i.e. as good video quality as full search, with reduction in compression time to achieve motion estimation process.

VI. Experimental Results

The Joint Model [19] reference software i.e. JM 16.1 developed by the joint team of ITU-T and ISO/IEC, was used to analyse the above mentioned search algorithms. The encoder was configured to work in the baseline profile with the first 100 frames to be encoded. The block distortion measure used was sum of absolute difference (SAD). The quantization parameter (QP) for the I-Slice and P-Slice was set at 28. Only the macro-block size of 16 x 16 was used in the implementation. The test video sequences [20] used were:

1) *football.cif*: Sequence consists of complex object movement with fast camera panning motion (Fig. 9a).
2) *foreman.cif*: Sequence with large motion in all direction (Fig. 9b).
3) *claire.qcif*: Video conference sequence with head and shoulder movement (Fig. 9c).
4) *missamerica.qcif*: Video conference sequence with head and shoulder movement, and static background (Fig. 9d).

(a) (b) (c) (d)

Fig. 9: Test Video Sequences

The results obtained are tabulated in Table I. The performance analysis is based on the motion estimation time (MET) and the PSNR value of the luminance component in the video sequence. It can be observed from Table I that full search give the optimum quality, but consume more computation time. On the other hand, the three-step search, four-step search and diamond search give fast result, but have poor compression quality. A better reduction in computation time is obtained in UMHexagonS, EPZS and FAME algorithm.

The PSNR comparison graphs of different algorithms for the video sequences used in the experimentation are shown in Figs. 10, 11, 12 and 13. It can be observed from the results that FS has the highest PSNR than any other algorithm. Thus, FS is the best algorithm if we require optimum visual quality regardless of compression time. Since *football.cif* is a sequence with complex movement, we can see from Fig. 10 that the PSNR of TSS is slightly better than 4SS. This is due to the fact that TSS performs better than 4SS for video sequences with large motion. For *foreman.cif*, the object motion affects the performance of 4SS, but the TSS performs comparatively better as shown in Fig. 11. On the other hand, the graph in Fig. 12 shows that the 4SS has better performance than TSS, since *claire.qcif* has small object movement. Similar results were obtained for *missamerica.qcif* shown in Fig. 13. Also, in all the results obtained, adaptive algorithms, viz. UMH, EPZS, FAME, outperform the suboptimal algorithms. In other words, adaptive algorithms are efficient than suboptimal algorithms in terms of both, the compression quality and the computation time.

TABLE I
COMPARISON OF VARIOUS MOTION ESTIMATION ALGORITHMS

Video Sequence	Search Algorithm	Average Y-PSNR (dB)	Total MET (sec)
football.cif (GOP = 9)	FS	37.427	831.738
	TSS	37.160	7.841
	4SS	37.171	10.112
	DS	37.154	157.457
	FAME	37.231	30.267
	EPZS	37.064	9.174
	UMH	37.137	24.042
foreman.cif (GOP = 15)	FS	37.302	673.861
	TSS	36.971	6.971
	4SS	36.973	8.736
	DS	36.978	145.478
	FAME	36.949	16.681
	EPZS	36.961	3.855
	UMH	36.971	11.608
claire.qcif (GOP = 15)	FS	40.379	135.688
	TSS	39.969	1.390
	4SS	39.964	1.789
	DS	39.976	32.770
	FAME	39.954	0.707
	EPZS	39.946	0.551
	UMH	39.962	1.245
miss america .qcif (GOP = 15)	FS	40.584	223.313
	TSS	40.159	1.687
	4SS	40.160	2.191
	DS	40.170	37.620
	FAME	40.119	1.120
	EPZS	40.159	0.726
	UMH	40.166	1.518

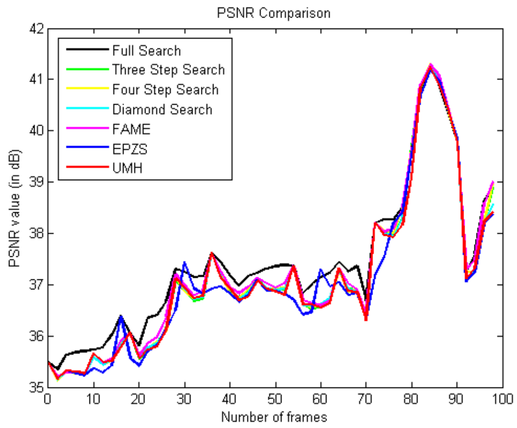

Fig. 10: PSNR comparison for *football.cif* video sequence

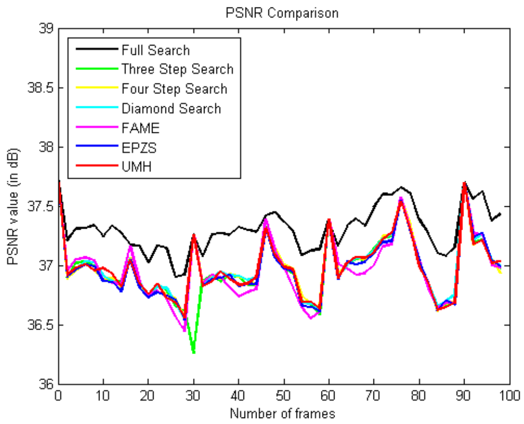

Fig. 11: PSNR comparison for *foreman.cif* video sequence

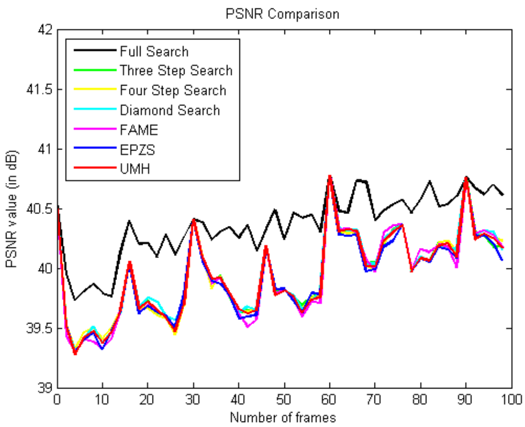

Fig. 12: PSNR comparison for *claire.qcif* video sequence

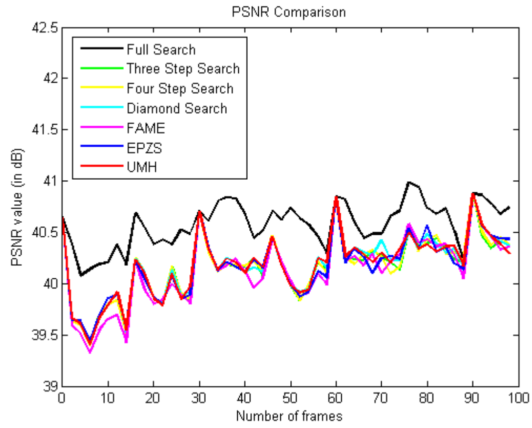

Fig. 13: PSNR comparison for *missamerica.qcif* video sequence

VII. CONCLUSION

In this paper, a comparative analysis is done for the motion estimation algorithms of optimal, sub-optimal and adaptive search techniques. From the experimental results, we conclude that although sub-optimal techniques reduce the computation time of full search technique, they incur a loss of visual data in the process. The adaptive search techniques are suitable for achieving good performance in terms of both compression quality and computation time. However, they do not provide as optimum compression as the full search. Recent developments have aided to improving the efficiency of adaptive search techniques. The rate-distortion optimization (RDO) has emerged as the efficient solution for improving video quality in compression. It refers to the optimization of the amount of distortion against the amount of data required to encode the video. RDO results are as efficient as full search, thereby improving visual quality of other motion estimation algorithms.

REFERENCES

[1] Iain E.G. Richardson, "H.264 and MPEG-4 video compression" Wiley Press, ISBN 0-470-84837-5, 2003

[2] W. Li and E. Salari, "Successive elimination algorithm for motion estimation", *IEEE Trans. Image Processing*, vol. 4, no.1, pp. 105-107, Jan 1995

[3] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for fast motion estimation", *IEEE Trans. Circuits Systems Video Technology*, vol. 4, no.4, pp. 438-442, Aug 1994

[4] X. Jing and L. Chau, "An efficient three-step search algorithm for block motion estimation", *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 435-438, Jun 2004

[5] L. Po and W. Ma, "A novel four-step search algorithm for fast block motion estimation", *IEEE Trans. Circuits Systems Video Technology*, vol. 6, no. 3, pp. 313-317, Jun 1996

[6] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation", *IEEE Trans. Circuits Systems Video Technology*, vol. 8, no. 4, pp. 369-377, Aug 1998

[7] S. Zhu and K. Ma, "A new diamond search algorithm for fast block-matching motion estimation", *IEEE Trans. Image Processing*, vol. 9, no. 2, pp. 287-290, Feb 2000

[8] M. Ghanbari, "The cross-search algorithm for motion estimation", *IEEE Trans. Communications*, vol. 38, no. 7, pp. 950-953, Jul 1990

[9] C.H. Cheung and L.M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Systems Video Technology*, vol. 12, no. 12, pp. 1168-1177, Dec 2002

[10] C. Zhu, X. Lin, and L. Chau, "Hexagon-based search pattern for fast block motion estimation", *IEEE Trans. Circuits Systems Video Technology* vol. 12, no. 5, pp. 349-355, May 2002

[11] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding", *IEEE Trans. Circuits Systems Video Technology*, vol. 6, no. 4, pp. 419-422, Aug 1996

[12] X. Lifen, H.Chunqing and C. Bihui, "UMHexagonS search algorithm for fast motion estimation", *3rd International Conference on Computer Research and Development* (*ICCRD*), vol. 1, pp. 483-487, Mar 2011

[13] A.M. Tourapis, O.C.L. Au and M.L. Liou, "Fast motion estimation using circular zonal search," in *Proc. on Visual Communications and Image Processing*, Dec1998

[14] A. M. Tourapis, O.C. Au, M. L. Liou, G. Shen and I. Ahmad, "Optimizing the MPEG-4 encoder - advanced diamond zonal search," in *Proc. Int. Symp. Circuits Syst. (ISCAS)*, Geneva, Switzerland, pp. 674-680, Jun 2000

[15] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation", in *Proc. on Visual Communications and Image Processing*, Jan 2002

[16] P. Hosur and K. Ma, "Motion vector field adaptive fast motion estimation", *2nd Intl. Conference on Information, Communications and Signal Processing (ICICS '99)*, Singapore, 7-10, Dec 1999

[17] A.M. Tourapis, O.C.L. Au and M.L. Liou, "Predictive motion vector field adaptive search technique (PMVFAST): enhancing block based motion estimation", in *Proc. On Visual Communications and Image Processing*, Dec 2000

[18] I. Ahmad, W. Zheng, J. Luo, and M. Liou,"A fast adaptive motion estimation algorithm", *IEEE Trans. Circuits Systems Video Technology*, vol. 16, no. 3, pp. 420-438, Mar 2006

[19] http://iphome.hhi.de/suehring/tml/, JM reference software

[20] http://trace.eas.asu.edu/yuv/index.html, YUV video sequences