

Linear Hashtable Method Predicted Hexagonal Search Algorithm with Spatial Related Criterion

Yunsong Wu¹, Graham Megson¹, Zhengang Nie², and F.N. Alavi³

¹ Computer Science, Reading University,
Reading, UK, RG6 6AA
{sir02yw, G.m.Megson}@rdg.ac.uk

² Beihang University,
zhengang.nie@ee.buaa.edu.cn

³ Computer Science, Queen Mary, University of London,
fna@dcs.qmul.ac.uk

Abstract. The paper presents a novel Linear Hashtable Method Predicted Hexagonal Search (LHMPHS) method for block base motion compensation. It bases on the edge motion estimation algorithm called hexagonal search (HEXBS). Most current variances of hexagonal search are investigated. On the basis of research of previous algorithms, we proposed a Linear Hashtable Motion Estimation Algorithm (LHMEA). The proposed algorithm introduces hashtable into motion estimation. It uses information from the current frame. The criterion uses spatially correlated macroblock (MB)'s information. Except for coarse search, the spatially correlated information is also used in inner search. The performance of the algorithm is evaluated by using standard video sequences and the results are compared to current algorithms such as Full Search, Logarithmic Search etc. The evaluation considers the three important metrics: time, compression rate and PSNR.

1 Introduction

In this paper, we propose a Linear Hashtable Motion Estimation Algorithm (LHMEA) to predict motion vectors for intra-coding. The objective of our motion estimation scheme is to achieve good quality video with very low computational complexity. Our method attempts to predict the motion vectors using linear algorithm. It uses hashtable method into video compression. After investigating of most traditional and on the edge motion estimation methods, we use latest optimization criterion and prediction search method. Spatially MBs' information is used to generate the best motion vectors. We also combine the LHMEA with HEXBS by motion predictor method. HEXBS is one of the best motion estimation methods currently. The new method improved by us achieves the best results so far; related statistics has been listed in this paper. The main contributions of this paper are (1) Megson introduced hashtable concept into video compression which uses several variables to represent whole MB.¹ This shows a direction for future research. (2) Linear Algorithm is used in video

¹ Graham Megson & F.N.Alavi Patent 0111627.6 -- for SALGEN Systems Ltd.

compression. This will improve speed; also leave space for parallel implementation. (3) LHMEA is combined with HEXBS. A new LHMEA predicted hexagonal method is proposed, which makes up for drawback of coarse search of HEXBS. It can also be used for Diamond Search etc. nearly all kinds of similar fast algorithms. (4) Spatially related MB's information is used not only in coarse search but also inner fine search.

There are a large number of motion prediction algorithms. We only focus on one class of such algorithms, so called the Block Matching Algorithms, which is widely used in MPEG2, MPEG4, and H.263. By partitioning a current frame into non-overlapping macroblocks with equal size, block-matching method attempts to find a block from a reference frame (past or future frame) that best matches a predefined block in the current frame. Matching is performed by moving and comparing with a criterion, which is called the MAE mean absolute error. The MB (macroblock) in the reference frame moves inside a search window centered on the position of the current block in the current frame. The best matched block producing the minimum distortion is found within the search window in the reference frame. However, the motion estimation is quite computationally intensive and can consume up to 80% of the computational power of the encoder if the full search is used. It is highly desired to significantly speed up the process without sacrificing the distortion seriously. Many computationally efficient variants were developed, among which are typically Two Level Search(TS), Two Dimensional Logarithmic Search(DLS), Subsample Search(SS)[1], the Three-Step search (TSS), Four-Step Search (4SS) [2], Block-Based Gradient Descent Search (BBGDS) [3], and Diamond Search (DS) [4][5] algorithms. A very interesting method called HEXBS has been proposed by Ce Zhu, Xiao Lin, and Lap-Pui Chau [6] in 2002 on IEEE. There are some variant HEXBSs, such as Enhanced Hexagonal method [7], Hexagonal method with Fast Inner Search [8].

2 Hexagonal Search Algorithm

HEXBS is an improved method based on DS (Diamond Search). It has shown the significant improvement over other fast algorithms for example DS. In contrast with the DS that uses a diamond search pattern, the HEXBS adopts a hexagonal search pattern to achieve faster processing due to fewer search points being evaluated. The motion estimation process normally comprises two steps. The low-resolution coarse search to identify a small area where the best motion vector is expected to lie, and then followed by fine-resolution inner search to select the best motion vector in the located small region. Most fast algorithms focus on speeding up the coarse search by taking various smart ways to reduce the number of search points in identifying a small area for inner search. There are two main directions to improve the coarse search, first is usage of predictors [8] [9], second is early termination [9]. In [8] a new algorithm was introduced on HEXBS, which is similar as Motion Vector Field Adaptive Search Technique (MVFAST) [10] based on DS. The algorithm has significantly improved the preexisting HEXBS algorithm both in image quality and speed up by initially considering a small set of predictors, namely the (0,0) motion vector and the motion vectors of the three spatially adjacent blocks(left, top, top-right) as possible motion vector predictor candidates.

2.1 Enhanced Hexagonal Algorithm with Variant Spatial Related Predictors

Modified Hexagonal pattern used the best motion vector predictor candidate as the center of search. In [9, 10] it was proposed a prediction set. In general, we can state that the blocks correlated with the current one, which are likely to undergo the same motion, can be divided into three categories as in Fig.1.

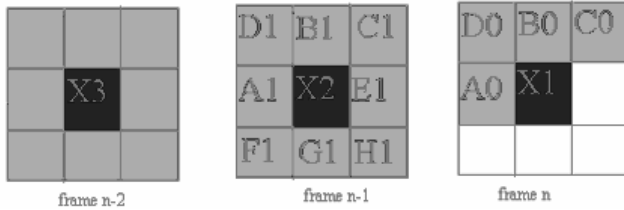


Fig. 1. Blocks correlated with the current one

- (1) Spatially correlated blocks (A0, B0, C0, D0),
- (2) Neighboring blocks in the previous frame (A1, B1, C1, D1, E1, F1, G1, H1)
- (3) Co-located blocks in the previous two frames (X2 and X3), which provide the acceleration motion vector (MV).

This last one can enhance temporal prediction in sequence with fast and non-uniform motion. So the MV set composed of the (0,0) vector; the median of the MVs of the left- up- and upright blocks (respectively named A0, B0 and C0 in Fig. 1; 4 neighboring blocks in the current frame; the ones of the co-located block (X2) and of the four vertically (B1, G1) and horizontally (A1, E1) adjacent blocks in the previous frame, and the acceleration motion vector $MV_{acc} = MV_{X2} + (MV_{X2} - MV_{X1})$

2.2 Enhanced Hexagonal Algorithm with Variant Spatially Related Pixels Inner Search

In the original HEXBS algorithm [6], it uses the large hexagon search pattern consisting of six endpoints. After coarse search procedure first locates a region where the optimal motion vector is most expected. The coarse search continues based on a gradient scheme until the center point of the hexagon has the current smallest distortion. After a hexagonal area is located in the coarse search, then the following fine-resolution search looks into the small area enclosed by the large hexagon for focused inner search. The original HEXBS inner search [6] used 4 points. Normal hexagonal inner search now uses 8 points inner search. If full search is required for the inner search, eight search points inside the large hexagon will be evaluated, which is computationally inefficient. Based on the monotonic distortion characteristic in the localized area around the global minimum, it is proposed to check only a portion of the inner search points that are nearer to the checked points with smaller distortions, which can save more than half of the eight search points inside. [8] This is based on knowledge that spatial coherence. Spatially related pixels have similar information and tend to be same in not only Sum of Absolute Difference (SAD) in motion

estimation but also similar motion vectors. It considers forming the 6 endpoints into 6 groups. For each group, a group distortion is defined by summing the distortions of all the points within the group. The area near to the group with the smallest group distortion is considered where the minimum distortion is most likely to be found, as explained in the figure 2.

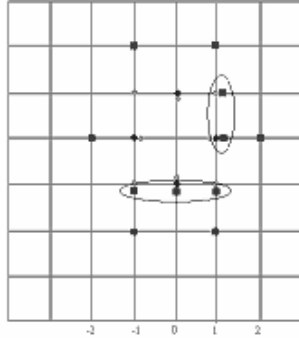


Fig. 2. 8 points hexagonal inner search

In figure 2, (a) Three inner points (1,-1) (0,-1) (-1,-1) nearest to bottom two endpoints (-1,-2) (1,-2) of hexagon with the smallest group distortion are to be checked. If the smallest distortion group is bottom two points, three checking points nearest to them will be used in the focused inner search. The top 3 inner points are the same. (b) Two inner points (1,0) (1,1) nearest to two endpoints (2,0) (1,2) of hexagon with the smallest group distortion are to be checked. If the smallest distortion group is (2,0) (1,2) , two inner points nearest to the smallest distortion group will be evaluated in the focused inner search.

The following table is for the statistics to show that inner group search is faster than the other inner search methods when PSNR does not change. In this figure inner group is 6-side-based fast inner search; inner normal is 8 points check; inner square is 4 points check. Normal inner search is the slowest of all.

Table 1. Comparison of inner search methods (based on 100 frames of Table Tennis)

Hex predictor	pred_median	pred_median	pred_median
Inner search	Inner group	Inner normal	inner square
Early termination	hex_near	hex_near	hex_near
compression time(P)	1	1	1
Frame per second(fps)	17.5325	15.9763	16.7702
compression Rate(P)	42	42	42
average P frame PSNR	21.2	21.2	21.2

2.3 Enhanced Hexagonal Algorithm with Early Termination Using SAD from Spatially Related MBs

Sequences with low or global motion usually have more predictors close to the optimum and provide an acceptable distortion. To take advantage of this situation, early termination criteria can be applied to minimize the number of matches. The chosen threshold takes into account the minimum SAD found for the adjacent blocks and for the current block in the last frame:

$$T = \min(MSAD_{A0}, MSAD_{B0}, MSAD_{C0}, MSAD_{X1}) + npel.$$

Where $MSAD_i$ is the minimum SAD found for block i ; $A0, B0$ and $C0$ refer to the left, up and upright block respectively, $X1$ is the current block in the previous frame and $npel$ is the number of pixels in the block. As in the figure below, the early termination can increase about 10% of compression speed. In the Table 2, hex max means without early termination, hex near is with early termination.

Table 2. Comparison of normal and early termination method (100 frames of Table Tennis)

Hex predictor	pred_hashtable	pred_hashtable
Inner search	inner group	inner group
Early termination	hex_max	hex_near
compression time(P)	2	2
Frame per second(fps)	10.9312	12.1076
Compression rate(P)	14	14
average P frame PSNR	27.5	27.5

3 Linear Hashtable Method Predicted Hexagonal Search (LHMPHS)

Most of current hexagonal search algorithms predictive methods focus on relations between current frame and previous frames. What we want to do is to find a fast method which discovers the predictors from current frame information. It uses spatially related MB or pixels' information. It is fast, accurate and independent on finding right predictors. So we designed a vector hashtable lookup matching algorithm which is more efficient method to perform an exhaustive search: it considers every macroblock in the search window. This block-matching algorithm calculates each block to set up a hashtable. It is a dictionary in which keys are mapped to array positions by a hash function. We try to find as few as possible variables to represent the whole macroblock. Through some preprocessing steps, "integral projections" are calculated for each macroblock. These projections are different according to different algorithm. The aim of these algorithms is to find best projection function. The

algorithms we present here has 2 projections, one of them is the massive projection, which is a scalar denoting the sum of all pixels in the macroblock. It is also DC coefficient of macroblock. Another is A of $Y=Ax+B$ (y is luminance, x is location.) Each of these projections is mathematically related to the error metric. Under certain conditions, the value of the projection indicates whether the candidate macroblock will do better than best-so-far match. The major algorithm we discuss here is linear algorithm

3.1 Linear Hashtable Motion Estimation Algorithm (LHMEA)

Linear Algorithm is most beautiful, easy and fast to calculate on computer because the construction of computer calculator bases on additions. So if most of calculations of video compression are done by linear algorithm, we can save lots of time on compression. It is also very easy to put on parallel machines in the future, which will benefit real time encoding. In the program, we try to use polynomial approximation to get such result $y=mx+c$; y is luminance value of all pixels, x is the location of pixel in macroblocks. The way of scan y is from left to right, from top to button. Coefficients m and c are what we are looking for. As in the figure 3

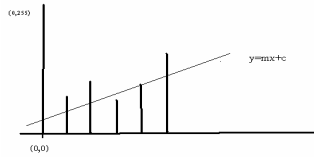


Fig. 3. Linear algorithm for discrete algorithm

In this function $y=f(x)$, x will be from 0 to 255 in a macroblock, $y=f(x)=mx+c$

$$m = \frac{N * \sum_{i=0}^N (x_i * y_i) - \sum_{i=0}^N x_i * \sum_{i=0}^N y_i}{N * \sum_{i=0}^N x_i^2 - \sum_{i=0}^N x_i * \sum_{i=0}^N x_i}$$

$$c = \frac{\sum_{i=0}^N y_i * \sum_{i=0}^N x_i^2 - \sum_{i=0}^N x_i * \sum_{i=0}^N x_i * y_i}{N * \sum_{i=0}^N x_i^2 - \sum_{i=0}^N x_i * \sum_{i=0}^N x_i}$$

In this way, we initially realize the way to calculate the hashtable. In previous research methods, when people try to find a block that best matches a predefined block in the current frame, matching was performed by SAD (calculating difference

between current block and reference block). In Linear Hashtable Motion Estimation Algorithm (LHMEA), we only need compare two coefficients of two blocks. In current existing methods, the MB moves inside a search window centered on the position of the current block in the current frame. In LHMEA, the coefficients move inside hashtable to find matched blocks. If coefficients are powerful enough to hold enough information of MB, motion estimators should be accurate. So LHMEA increases lots of speed, accuracy and will make a new era of video encoding.

3.2 Linear Hashtable Method Predicted Hexagonal Method

After motion estimators are generated by LHMEA, they will be used as predictors for HEXBS for coarsely search. These predictors are different from all previous predictors. They are based on full search and current frame only. Because LHMEA is linear algorithm, it is fast. Because the predictors generated are accurate, it will improve HEXBS without too much delay in speed.

In the Figure below, we compared Full Search (FS), Linear Hashtable Motion Estimation Algorithm (LHMEA), Subsample Search(SS), Two Level Search(TLS), Logarithmic Search(LS) and three kinds of HEXBS Algorithms. Three HEXBS Algorithms are Hexagonal Search without predictor(HEXBS), LHMPHS and Hexagonal with median predictors of spatially adjacent blocks (left, up and upright blocks what are respectively named A0, B0 and C0 in Fig. 1) (HSM)[10]. All HEXBS algorithms used 6-side-based fast inner search [9] and early termination criteria [10] mentioned in our paper. All the data here refer to P frames only. HEXBS can achieve nearly the same PSNR as FS and only takes 10% time of FS. The LHMPHS is better than HEXBS without predictor on compression rate when time and PSNR are the same. HSM is the best algorithm of three. But if we can find better coefficients in the hashtable to represent MB, the hashtable will have a wonderful future.

Table 3. Comparison of compression rate, time and PSNR between FS, LS, SS, TLS, HEXBS, LHMPHS, HSM (based on 100 frames of Table Tennis)

Search Method	EXHAUSTIVE	LOGARITHMIC	SUBSAMPLE	TWO LEVEL	HexNo Predictor	Pred_Hashtable	Pred_Median
Inner Search					Inner Group	Inner Group	Inner Group
Early Termination					Hex_near	Hex_near	Hex_near
Compression Time(s)	11	1	4	3	1	1	1
(fps)	2.3726	24.7706	6.4286	7.3171	19.4245	14.2857	17.5325
Compression Rate	48	42	48	48	37	39	42
PSNR	21.3	21.1	21.3	21.3	21.2	21.2	21.2

The figure below shows the PSNR from 5 to 100 frames of flow garden data stream among FS, HEXBS, LHMPHS, HSM. It shows the LHMPHS has better PSNR than the other algorithms.

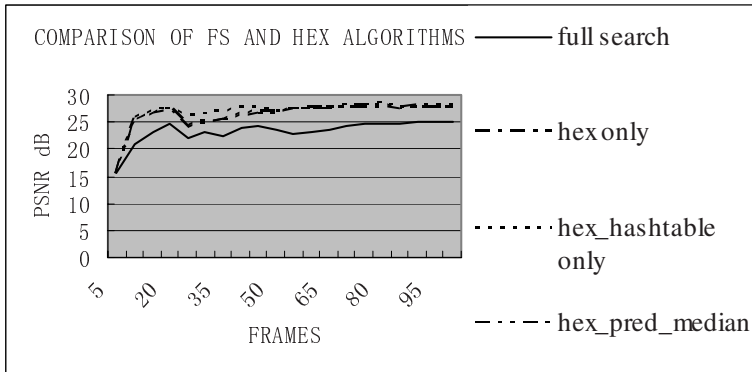


Fig. 4. Comparison of PSNR between FS, HEXBS, LHMPHS, HSM (based on 5-100 frames of Flower Garden)

But in the following figure about PSNR from 5 to 100 frames of table tennis data stream, all algorithms' PSNR are the same. It means LHMEA works better on large motion vector video stream.

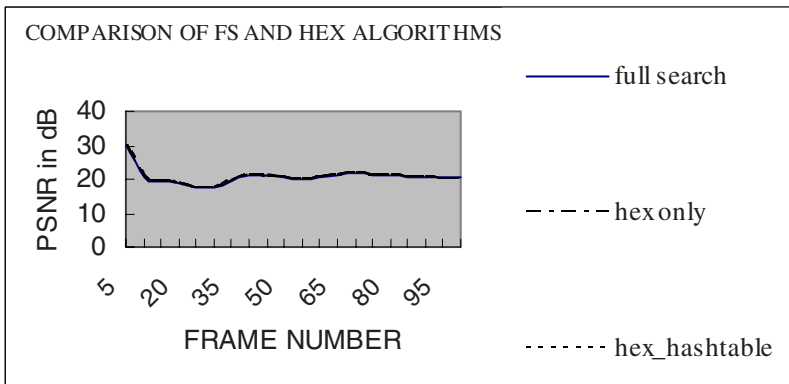


Fig. 5. Comparison of PSNR between FS, HEXBS, LHMPHS, HSM (based on 5-100 frames of Table Tennis)

The FS, HEXBS, LHMPHS are certain center biased algorithms. This is also basis of several other algorithms. It was based on the fact that for most sequences motion vectors were concentrated in a small area around the center of the search. This can also be seen in the figures below. Unfortunately for some sequences this is not always true as can be seen in the flower garden figure 6 (a), which implies that these algo-

rithms will have reduced performance in such cases. From additional simulations we can observe that predictor seems to have much higher correlation with the current motion vector than (0,0) even for non-center biased sequences such as the flower garden mentioned previously. This suggests that, instead of initially examining the (0,0) position, we can achieve better results if the linear hashtable predictor is examined first and given higher priority with the usage of an early termination threshold.

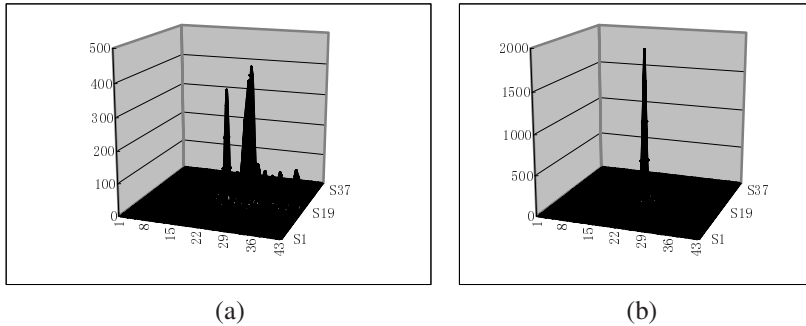


Fig. 6. Motion vector distribution in (a) Flower Garden and (b) Table Tennis by LHMEA

4 Summary

In the paper we proposed a new algorithm called Linear Hashtable Motion Estimation Algorithm (LHMEA) in video compression. It uses linear algorithm to set up hashtable. The algorithm searches in hashtable to find motion estimator instead of by FS. Then the motion estimator it generates will be sent to HEXBS, which is best motion estimation algorithm, as predictor. No matter in coarse search or fine inner search, new method used lots of spatial related MB or pixels' information. In this way, it improves both quality and speed of motion estimation. The key point in the method is to find suitable coefficients to represent whole MB. The more information the coefficients in hashtable hold about pictures, the better result LHMPHS will get. This also leaves space for future development.

References

1. Ze-Nian Li Lecture Note of Computer Vision on personal website (2000)
2. L. M. Po and W. C. Ma: A novel four-step search algorithm for fast block motion estimation," *IEEE and systems for video technology*, vol. 6, pp. 313–317, (June 1996.)
3. 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.)
4. S. Zhu and K.-K. Ma: A new diamond search algorithm for fast blockmatching motion estimation: *IEEE Trans. Image Processing*, vol. 9, pp. 287–290, (Feb. 2000.)

5. 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 and systems for video technology, vol. 8, pp. 369–377, (Aug. 1998)
6. Ce Zhu, Xiao Lin, and Lap-Pui Chau: Hexagon-Based Search Pattern for Fast Block Motion Estimation: IEEE Trans on circuits and syst. for video technology, Vol. 12, No5, (May 2002)
7. C. Zhu, X. Lin and L.P. Chau: An Enhanced Hexagonal Search Algorithm for Block Motion Estimation: IEEE International Symposium on Circuits and Systems, ISCAS2003, Bangkok, Thailand, (May 2003)
8. Ce Zhu, Senior Member, IEEE, Xiao Lin, Lappui Chau, and Lai-Man Po: Enhanced Hexagonal Search for Fast Block Motion Estimation: IEEE Trans on circuits and systems for video technology, Vol. 14, No. 10, (Oct 2004)
9. Paolo De Pascalis, Luca Pezzoni, Gian Antonio Mian and Daniele Bagni: Fast Motion Estimation With Size-Based Predictors Selection Hexagon Search In H.264/AVC encoding: EUSIPCO (2004)
10. Alexis M. Tourapis, Oscar C. Au, Ming L. Liou: Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) Enhancing Block Based Motion Estimation: proceedings of Visual Communications and Image Processing, San Jose, CA, January (2001)
11. A. M. Tourapis, O. C. Au and M. L. Liou: Highly Efficient Predictive Zonal Algorithms for Fast Block. Matching Motion Estimation: IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, No.10, pp 934-947, (October 2002)