

Semi-Hierarchical Based Motion Estimation Algorithm for the Dirac Video Encoder

M. TUN, K. K. LOO, J. COSMAS
School of Engineering and Design
Brunel University
Kingston Lane, Uxbridge, UB8 3PH
UNITED KINGDOM

myo.tun@brunel.ac.uk, jonathan.loo@brunel.ac.uk, john.cosmas@brunel.ac.uk
<http://www.brunel.ac.uk/>

Abstract: - Having fast and efficient motion estimation is crucial in today's advance video compression technique since it determines the compression efficiency and the complexity of a video encoder. In this paper, a method which we call semi-hierarchical motion estimation is proposed for the Dirac video encoder. By considering the fully hierarchical motion estimation only for a certain type of inter frame encoding, complexity of the motion estimation can be greatly reduced while maintaining the desirable accuracy. The experimental results show that the proposed algorithm gives two to three times reduction in terms of the number of SAD calculation compared with existing motion estimation algorithm of Dirac for the same motion estimation accuracy, compression efficiency and PSNR performance. Moreover, depending upon the complexity of the test sequence, the proposed algorithm has the ability to increase or decrease the search range in order to maintain the accuracy of the motion estimation to a certain level.

Key-Words: - Semi-Hierarchical, Motion Estimation, Dirac Wavelet Video Codec

1 Introduction

Motion estimation is extensively used in most standard video encoder as a means to exploit temporal redundancy by removing the redundant pixels in temporal domain between frames of video. The key step in removing temporal redundancy is the prediction of motion vector (MV) between the current frame and the reference frame. The most reliable Motion Estimation (ME) algorithm is the Full Search Block Matching Algorithm (FS-BMA) where prediction of motion vector is performed under block-by-block basis and it is widely used in the reference software as a benchmark. However, FS-BMA requires huge computational load since it attempts to match every possible candidate using a certain type of cost function in the given search window size making it impractical for a real-time video encoding. The larger the search window size, the higher number of computation would be required. The total number of computation required is $(2w+1)^2$ where w is the size of the search window. Over the past decade, many fast and efficient block matching algorithms (BMA) have been proposed in order to achieve the accuracy and speed at the same time. Among them, some of the well known algorithms are the three-step search (TSS) [1], the new three-step search (NTSS) [2], the four-step

search (FourSS) [3], the Diamond Search (DS) [4], the Adaptive Rood Pattern Search (ARPS) [5], the Adaptive Irregular Pattern Search (AIPS) [6] and Fast and Robust Search [7] etc.. Most recently, Multi-Direction Cross-Hexagonal Search Algorithms was proposed [8], in which search pattern is based upon the hexagonal shape instead of using traditional square and diamond shape patterns. The goal of these algorithms is to reduce the number of search points at the expense of motion estimation accuracy and compression efficiency. With the use of these algorithms, the number of cost function calculation is greatly reduced with a certain level of accuracy. However, most of these algorithms were tested on the platform independent IPPPP... Group of Picture (GOP) structure and a complete algorithm which can be applied to a functional video encoder with any types of video formats and GOP structure is still required. From among the fast BMAs, the adaptive search algorithms [5][6][9] become increasingly popular because of their flexibility in choosing the center of search location adaptively. Since there is no limitation on their search range, adaptive based search algorithms can tack the global minimum quite accurately. There are several methods in predicting the center point of search location. Most commonly used methods are the

spatial prediction where center search point of current block is predicted from the adjacent left, top and top left blocks, the temporal prediction where center point is predicted from the corresponding block of the previous encoded frame and the hierarchical prediction where center point is predicted from the corresponding block location of the previous hierarchical layer.

The current release of the Dirac encoder [10] employs fully hierarchical motion estimation for all types of inter frames (both P and B) coding causing the encoder practically impossible to apply in real time encoding especially for High Definition (HD) video sequences. The main objective of this paper is to propose the fast and efficient motion estimation strategy which combine modified adaptive search algorithm and semi-hierarchical approach where hierarchical motion estimation is considered only for a certain type of inter frame encoding.

The rest of the paper is organized as follows. Section 2 describes the motion estimation strategy in current release of Dirac and the proposed fast motion estimation algorithm using semi-hierarchical approach is detailed in Section 3. Results and discussion followed by conclusion are presented in Section 4 and 5 respectively.

2 Motion Estimation in DIRAC

In its hierarchical motion estimation, Dirac first down converts the size of the current and reference of all types of inter frames (both P and B) using the 12 taps down conversion filter. The number of down conversion levels depends upon the frame format and can be calculated using equation 1 as follow.

$$level = \min \left(\log_2 \left(\frac{width}{12} \right), \log_2 \left(\frac{height}{12} \right) \right) \quad (1)$$

According to equation 1, the number of down conversion levels is 4 and 6 for the frame format CIF and HD (1920×1080) respectively. In the down conversion process, the dimension (both height and width) of the frames are reduced by the factor of two in each levels. The motion estimation is performed first in the lowest resolution (smallest frame) level and gradually increased to the higher resolution levels until it reaches the original frame size. The search pattern used in lowest level is Diamond shape with the search range 5 and all other levels higher than lowest use square shape search pattern with search range 1. Fig. 1 shows both search patterns where there are altogether 61 search points in Diamond shape and 9 points in square shape.

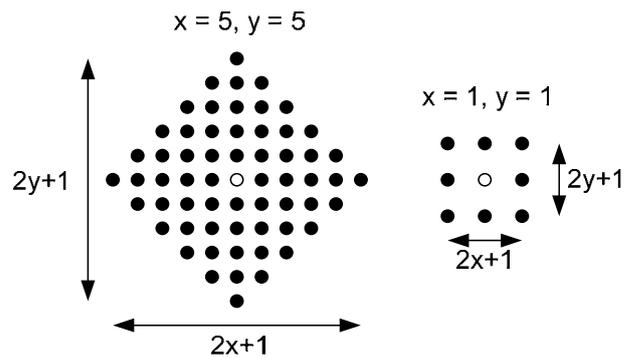


Fig. 1. Search Patterns of Dirac

First of all, candidate lists which are the lists to be searched are generated. A candidate list consists of a number of points to be searched, which follows a certain pattern either diamond or square as shown in Fig. 1 and centered at a predicted MV. The predicted MV can be either zero, spatially predicted or guide MV. Spatially predicted motion vector is the medium vector of block number 1, 2 and 3 or mean vector of block 1 and 2 as shown in Fig. 2, depending upon the location of the current block where motion estimation is carried out. Guide vector is the best motion vector at the corresponding block location of the adjacent lower hierarchical level and it is not available for the lowest level.

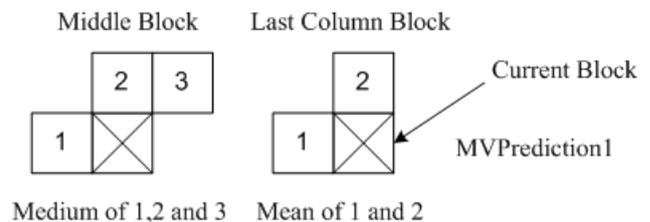


Fig. 2. Spatially Predicted Motion Vector of Dirac, the current block is the block where ME is being performed

In Fig. 3, for lowest level search, two candidate lists are generated centered at zero motion vector and spatially predicted motion vector respectively with the diamond search pattern. Sum of the Absolute Difference (SAD) is used here as the cost function. At the initial search step, the SAD calculation is carried out only for the center point of diamond pattern in each list and finds the list which gives the minimum cost. The candidate lists to be searched are chosen by multiplying the minimum cost with 1.5 and choose all the lists which give the cost less than 1.5 times minimum costs. So, there can be at most two candidate lists and 122 search points can be involved in lowest level search if there

is no overlapping between the two lists. In the refine step, SAD calculation is carried out for all chosen candidate lists on their corresponding search points and the coordinate of the point which gives the minimum cost, is recorded as the best MV.

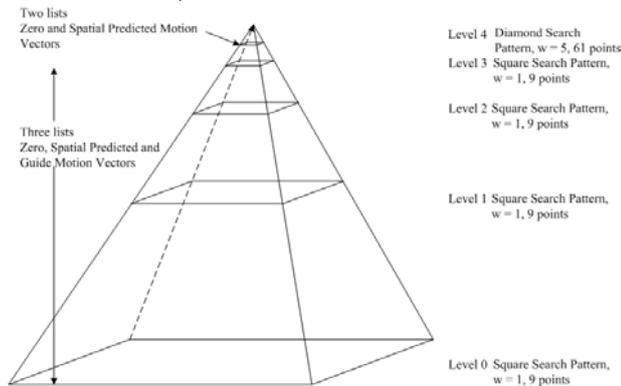


Fig. 3. Dirac's Four levels Hierarchical Motion Estimation for CIF video format, where w is search range

The search procedure is basically the same for all other levels except the addition of one more candidate list which is centered at the guide vector. So, there are three candidate lists in these levels with the square search pattern as shown in Fig. 1 and the maximum number of search points can be at most 27 in each level if there is no overlapping between the lists.

After going through all these levels, the pixel accuracy motion vectors for each block are obtained. Dirac provides the option to find the motion vectors up to 1/8 pixel accuracy. In order to achieve this, motion estimation undergoes subpel refine process where the current and reference pictures are up converted by 2, multiply the pixel accuracy motion vector by 2 and search around the pixel accuracy motion vector block to get 1/2 pel accuracy motion vector. The above procedure is repeated until it gets the require accuracy.

After getting the required accuracy motion vectors for each block, the last stage of motion estimation, mode decision is carried out by using RDO motion estimation matrix. The metric consists of a basic block matching metric which is SAD plus some constant times a measure of the local motion vector smoothness. The smoothness measure is based on the difference between the candidate motion vector and the median of the neighboring previously computed motion vectors. The total metric is a combination of these two metrics. Given a vector V which maps the current frame block P to a block $R=V(P)$ in the reference frame, the metric is given by,

$$SAD(P, R) + \lambda \cdot \max(|V_x - M_x| + |V_y - M_y|, 48). \quad (2)$$

In mode decision, Dirac considers the total of 12 modes which includes the combination of 3 Macro Block (MB) splitting levels as shown in Fig. 4 and 4 prediction modes. A MB consists of a 4×4 array of blocks, and there are three possible ways of splitting a MB:

Splitting level 0: no split, a single MV per reference frame for the MB;

Splitting level 1: split into four sub-macroblocks (sub-MBs), each a 2×2 array of blocks, one MV per reference frame per sub-MB;

Splitting level 2: split into the 16 constituent blocks.

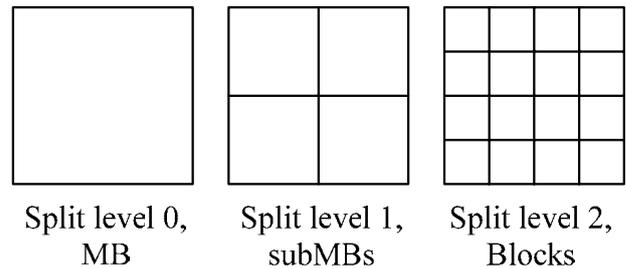


Fig. 4. MacroBlock Splitting Modes

At the same time, the best prediction mode for each prediction unit (block, sub-MB or MB) is chosen. There are four prediction modes available: INTRA: intra coded, predicted by DC value; REF1_ONLY: only predict from the first reference; REF2_ONLY: only predict from the second reference (if one exists); REF1AND2: bi-directional prediction.

3 Semi-Hierarchical Fast ME

In the existing ME search strategy of Dirac, even though the algorithm could locate the minimum SAD point with a certain level of accuracy, the whole process takes too long to complete because of the usage of multiple levels of hierarchies for all types of inter frames, i.e. for both P and B frames. For example, encoding a CIF format video sequence would require the algorithm to generate four levels of hierarchy for both current and reference frames. The algorithm search the optimum motion vector in each by calculating the SAD of each point using the corresponding pattern as shown in Fig. 1. After completing these four levels, the final search is carried out again in the original frame level itself with the square search pattern. Obviously, it is the most time consuming stage and requires approximately 80% of total encoding time. So it is required to find the faster ME search strategy, which

could give the same accuracy or even better if possible.

The proposed strategy is based upon the extension and optimization of the existing method. More predicted MVs (i.e. more candidate lists) are added at the initial search in order to find the most probable minimum point as quickly as possible. So, the initial search now includes six predicted MVs instead of three, which are zero, three spatially predicted (MVPrediction1, MVPrediction2 and MVPrediction3), guide and temporally predicted motion vectors.

MVPrediction1 is the spatially predicted MV used in the existing ME algorithm of Dirac. The other two spatially predicted MVs, MVPrediction2 and MVPrediction3 which are the best MVs from left and top blocks respectively, are added in the proposed method in order to get the better prediction for the horizontal and vertical camera panning. The idea of zero and guide motion vectors are the same as existing algorithm but the last motion vector which is predicted temporally is added to the list in order to exploit the temporal redundancy of the video sequence. This is the vector resulting from the motion estimation of the previous successive frame at the corresponding block location.

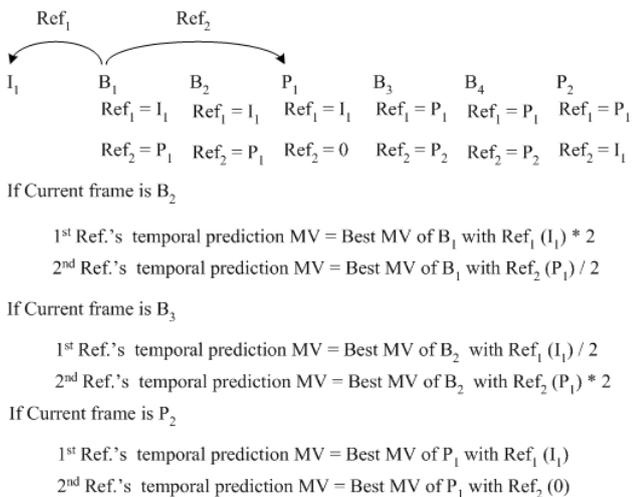
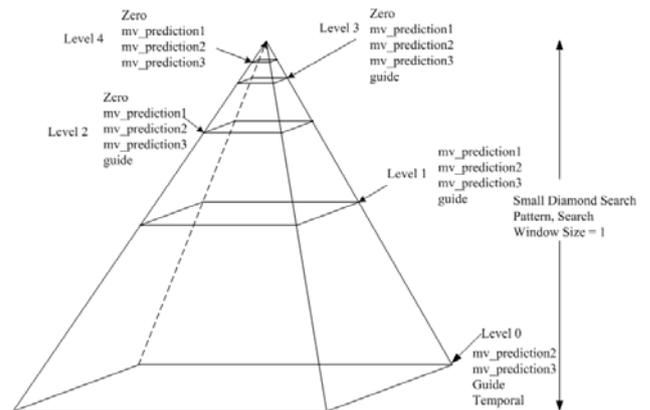


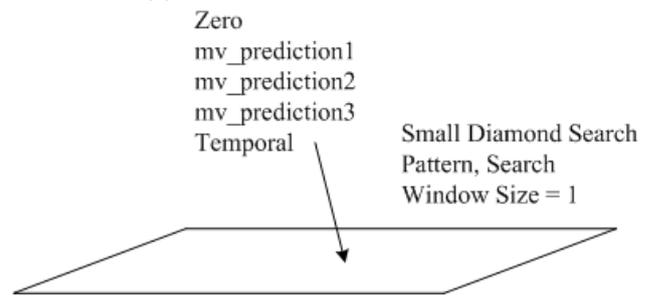
Fig. 5. Temporal MV Prediction for P and B frames

Fig. 5 shows the prediction of temporal MV for the different types of frames either P or B. Note that there is no temporal predicted MV available for the first P and B frames. Again, the best MVs from the P frame cannot be used as the temporal predicted MV for the successive B frame since the prediction structure of the P and B are different as shown in Fig. 5. Furthermore, the prediction of the temporal MV for B frames requires scaling up or down since the temporal distances to the references for a particular reference type (either reference 1 or 2) are

different for B frames. For example, in the temporal MV prediction of B₂ frame, the temporal distance for B₁ to its 1st reference, which is I₁, is 1 but the distance for B₂ to its 1st reference, which is also I₁, is 2. So, it is required to multiply the best MV of B₁ to its 1st reference by 2 in order to get the proper predicted temporal MV for the 1st reference of B₂. The same reason applies for the 2nd reference. It is important to note that the temporal predicted MV is available only for the level 0 motion estimation.



(a) Motion Estimation for P frame



(b) Motion Estimation for B frame

Fig. 6. Proposed Semi-Hierarchical Fast Motion Estimation for CIF video format

In order to reduce the level of complexity, the hierarchical motion estimation is employed only for P frame. The idea of unequal level of motion estimation for P and B frames or semi-hierarchical motion estimation comes from the following facts. According to the nature of GOP structure, the reference for P frames are typically far away (e.g. three to six frames in temporal separation for IBBP GOP structure) from the current frame and clearly it is unlikely to find the best match near the vicinity of the current block. So, it is required either to increase the search range or to introduce the hierarchical way of motion estimation. Another factor is that the quality of the P frame plays an important role in getting the lower residual error weight in the motion estimation of B frames. The only way to maintain

the quality of the P frame without losing the compression efficiency is to increase the accuracy of the motion estimation in order to reduce the residual error weight.

Fig. 6 shows the proposed semi-hierarchical fast motion estimation scheme for P frame and B frame in CIF format. There are altogether four candidate lists to be searched initially (except guide and temporal MV) in level n (where $n = 4$ for CIF video format) of P frame since the MVs of the rest two candidate lists are not available. In addition to this, the guide MV is added in levels $(n-1)$ to 2 (which are level 3 and 2 in CIF format) so that the total number of lists in these levels becomes five. Zero MV is removed in level 1 since it is not required to search the MV of stationary object in all levels. In level 0, MVPrediction1 is replaced with temporal MV leaving only four essential candidate lists in highest resolution frame level. For one level B frame search, it includes altogether five candidate lists except guide MV since there is no hierarchical way of motion estimation.

Fig. 7 shows the detail algorithm flow chart. The search pattern used in proposed method is small diamond (SD) with search window, w which is set to 1 for all cases. Depending upon the level of hierarchy, the number of candidate lists to be searched initially for P frame can be varied as shown in Fig. 6 (a). But the initial number of candidate list to be searched in B frame is constant, i.e. 5 candidate lists excluding temporal. Similar to the existing algorithm of Dirac, at the initial search stage, the cost is calculated only at the center point of SD search pattern in each candidate list and finds the minimum list i.e. the list which gives minimum cost. The group of lists to be searched at the next stage can include more than one list if two or more lists give the same minimum cost. An early stopping criterion is incorporated in all cases, which allows the algorithm to skip the refine search stage when the minimum SAD cost at the initial search is less than the number of coefficients in a block (1^{st} threshold).

At the refine search stage, the center MVs from each list are extracted one by one, set as the center MV and find the best cost which is minimum cost, around this MV by using SD search pattern. If the best cost point is not at the center, set the current best cost point as the center, increase the number of loop by one and search again its surrounding 4 points. The same procedure continues until the stopping criteria is met, which is either the best cost point is at the center or best cost is less than two times the multiplication of width and height of the block (2^{nd} threshold) or the number of looping is

more than five. Once the stopping criteria is met, the corresponding MV is saved to the best_MV_list[] array. The looping continues until there are no more MVs to be set as the center MV in the list to be searched. Finally, choose the best MV from the best_MV_list[] by comparing their corresponding cost.

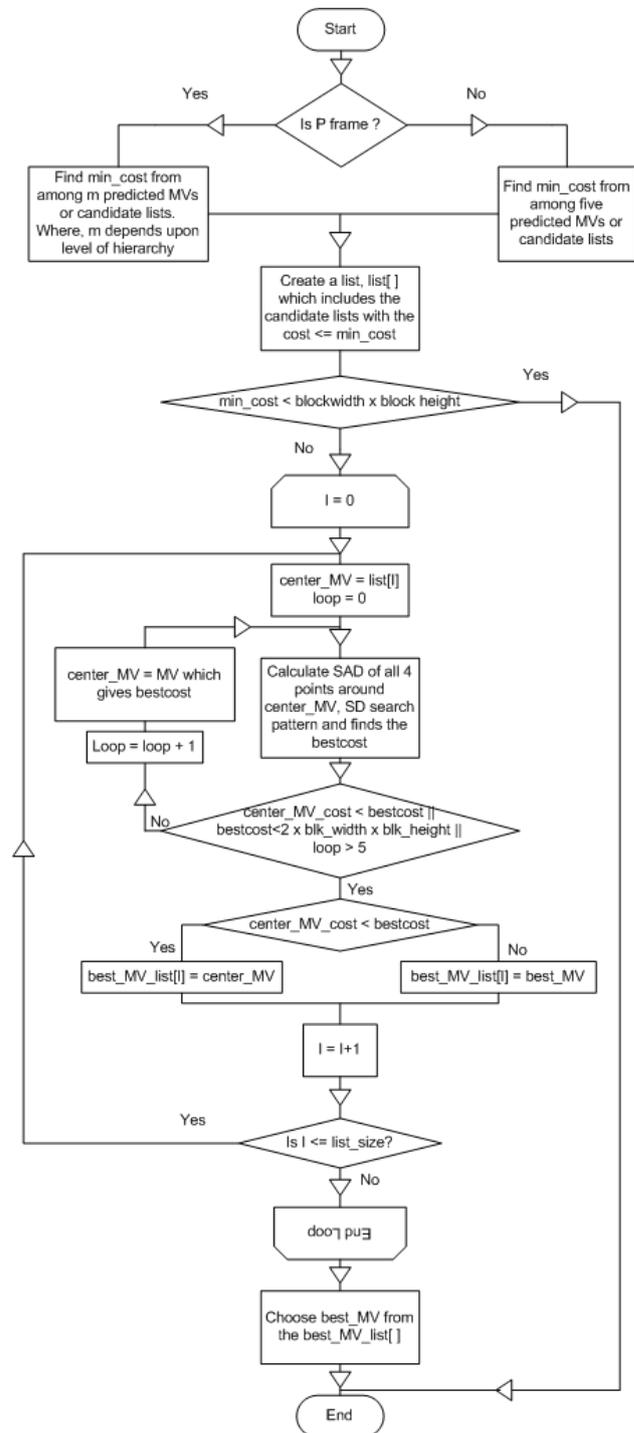


Fig. 7. The Proposed Semi-Hierarchical Fast ME Algorithm flow chart

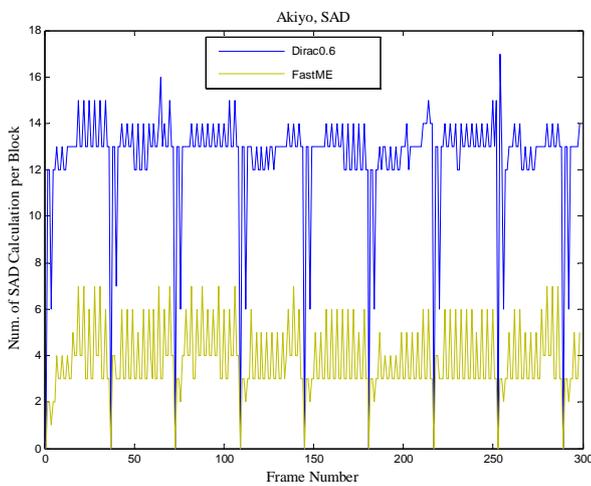
4 Results and Discussions

In order to evaluate the performance of the proposed algorithm, several test sequences ranges from slow, medium to high motion in CIF formats were used. As for the test platform, Dirac version 0.6 from [10] has been employed. The GOP length is set to 36 which means the number of P frames is 11 and P frame separations is 3 forming IBBPBBP GOP structure.

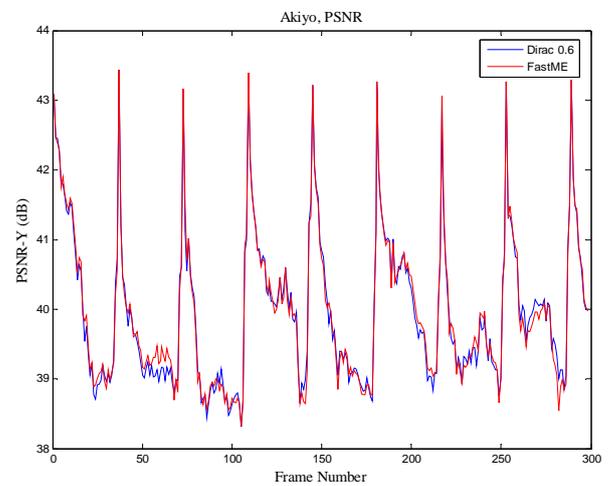
Table 1 shows the motion estimation results from both Dirac 0.6 and proposed fast ME algorithm for different test sequences in CIF format. In table 1, weight refers to the average residual error frame's weight for the whole sequence and the lower weight reflects the higher accuracy in the corresponding motion estimation algorithm.

Table 1. The Comparison of ME Results for CIF Video format

| Sequence | Algorithm | SAD | Weight | File Size (bytes) | PSNR-Y (dB) |
|----------|-----------|-------|--------|-------------------|-------------|
| Akiyo | Dirac 0.6 | 12.48 | 4.45 | 237959 | 39.97 |
| | Fast ME | 3.76 | 4.45 | 238538 | 39.97 |
| Foreman | Dirac 0.6 | 32.46 | 8.24 | 614303 | 34.63 |
| | Fast ME | 14.78 | 8.18 | 609187 | 34.67 |
| Bus | Dirac 0.6 | 34.33 | 9.10 | 636695 | 31.50 |
| | Fast ME | 15.04 | 8.69 | 590205 | 31.68 |
| Football | Dirac 0.6 | 40.39 | 9.46 | 275735 | 33.03 |
| | Fast ME | 23.28 | 9.43 | 274491 | 32.91 |

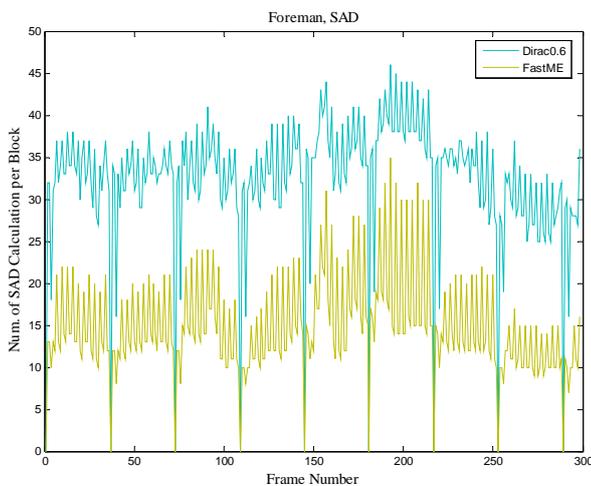


(a) Comparison of Average Number of SAD Calculation

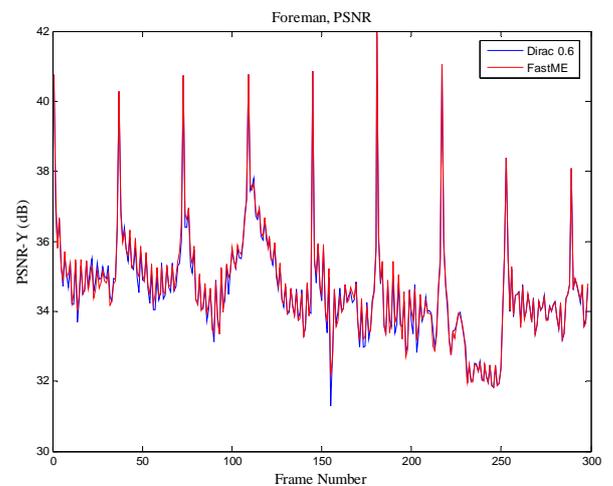


(b) Comparison of PSNR-Y

Fig. 8. Comparison of ME results for Dirac 0.6 and Proposed Fast ME, Akiyo in CIF format

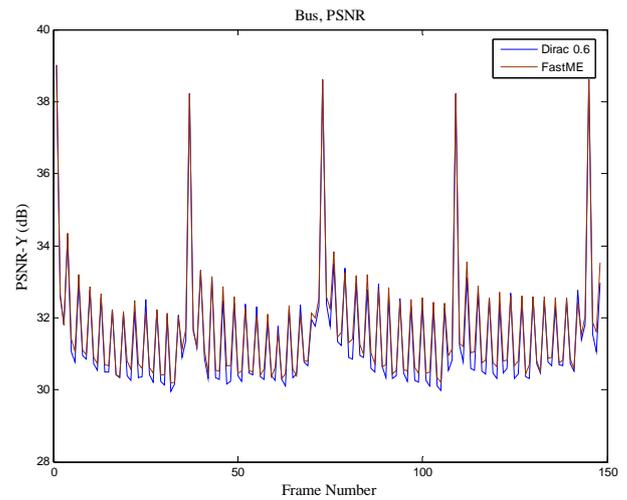
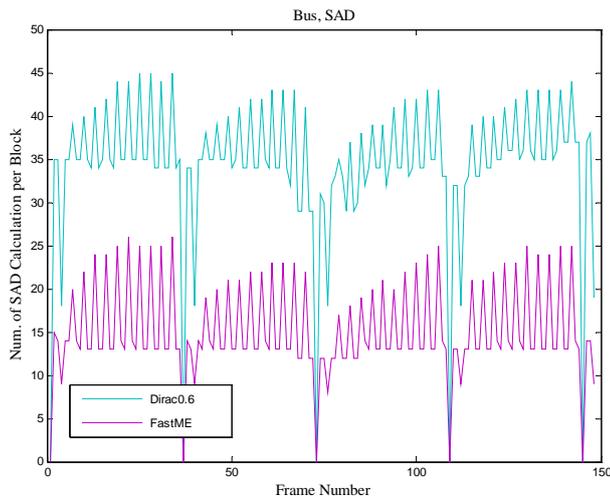


(a) Comparison of Average Number of SAD Calculation



(b) Comparison of PSNR-Y

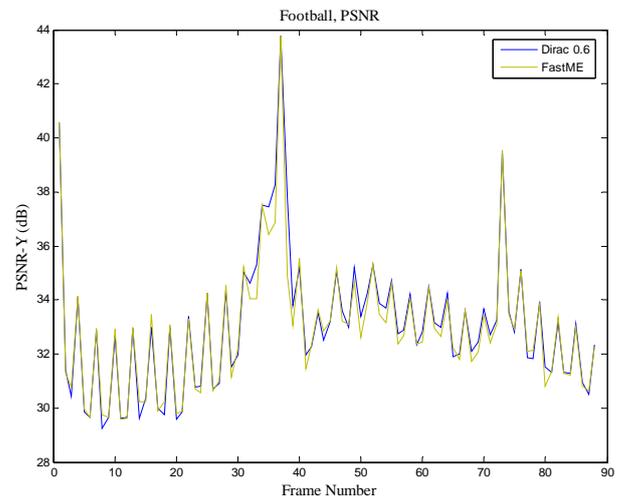
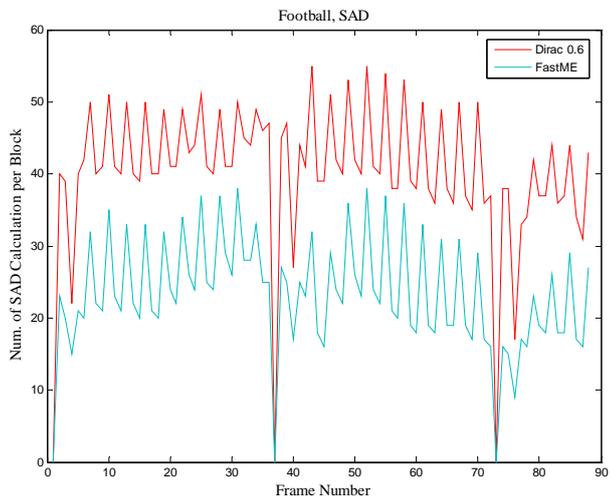
Fig. 9. Comparison of ME results for Dirac 0.6 and Proposed Fast ME, Foreman in CIF format



Comparison of Average Number of SAD Calculation

(b) Comparison of PSNR-Y

Fig. 10. Comparison of ME results for Dirac 0.6 and Proposed, Bus in CIF format



(a) Comparison of Average Number of SAD Calculation

(b) Comparison of PSNR-Y

Fig. 11. Comparison of ME results for Dirac 0.6 and Proposed Fast ME, Football in CIF format

It is the division of the combination of absolute value of the coefficients in residual error frame by frame dimensions and the number of frames in the sequence. Basically, a motion estimation algorithm can be evaluated by determining the accuracy and complexity of the algorithm. In this research, the accuracy and complexity are represented in terms of the residual error frame's weight and the average number of SAD calculation per block, respectively. Fast ME algorithm gives the average weight which is slightly lower than Dirac 0.6 in all sequences except Akiyo. In terms of compression efficiency, again the proposed algorithm gives approximately equal or smaller file size for all test sequences and it

is more significant especially in the Bus sequence. PSNR for Y component is used to compare the objective quality of the reconstructed frames. As shown in table 1, all the test sequences give approximately the same value of PSNR for both algorithms.

But there is significant improvement in proposed one as far as the speed of the algorithm is concerned. There is a huge saving, at least two folds in average number of SAD calculation per block for all test sequences. Reduction in the number of SAD calculation is much more significant in relatively static sequence (e.g. Akiyo) where the required number of SAD calculation per block in proposed

algorithm is more than one third of Dirac 0.6. It is simply because of the application of early termination method after the initial search in proposed algorithm. An early stopping criterion allows the algorithm to skip the refine search stage when the minimum SAD cost at the initial search is less than the number of coefficients in a block. In static sequences, the chances of meeting the early stopping criterion is quite high for most of the blocks since the displacement of both background and foreground objects are not much significant between the adjacent frames. On the other hand, dynamic motions sequences (e.g. Football) require refine search stage since initial search results are not good enough to stop the algorithm for most of the blocks, requiring more number of SAD calculation compared with the less dynamic sequences. In Dirac, the number of reference frames to be searched for motion estimation is 2 and so the average number of required SAD calculation for one block per one reference frame is approximately half of the given values in table 1.

Fig. 8 to 11 show the number of SAD calculation per block and PSNR-Y of each frame for all test sequences shown in Table 1. As expected, the average number of SAD calculation per block in P frames using proposed algorithm is much higher than that of B because of the application of semi-hierarchical motion estimation, resulting the wider band of SAD calculation difference between P and B compared with Dirac 0.6 as shown in figures 8(a) to 11(a). But it is interesting to note that the proposed algorithm has the ability to increase or decrease its search range depending upon the complexity of the test sequence. For example, in Fig. 9(a), the average number of SAD calculation in proposed algorithm is lower while the motion is relatively static in Foreman sequence for the first four GOPs. Then, the algorithm increase its search range once it detects the dynamic motion giving higher number of SAD calculation per block in the fifth and sixth GOP in order to maintain the level of accuracy in motion estimation.

Again, the algorithm reduces its search range for the remaining frames which have less dynamic motions giving lower number of SAD calculation. The application of double thresholds system, one in the initial search and another one in the refine search stage serves as the complexity detection mechanism, controlling the accuracy of overall ME accuracy effectively. Unlike the proposed algorithm, there is no such adaptation in the motion estimation of Dirac 0.6 giving relatively constant number of search in all type of sequences.

Even though the proposed algorithm uses an equal level of motion estimation between different frames types, the PSNR level of each frames are approximately the same and sometime even slightly higher than Dirac 0.6 as shown in Fig. 8 (b) to 11 (b). PSNR results in Fig. 8(b) to 11 (b) confirm that the proposed fast ME algorithm can still maintain the same accuracy even with the non-hierarchical way of coding for intra-non-reference (B) frames. This result can be further confirmed by comparing the weight of the residual error frame for both ME algorithms in Fig. 12. Since weight can be represented as the accuracy of corresponding ME algorithm, getting approximately equal weights for B frames coding in Fig. 12 confirm the above statement.

The above discussion has already proved the importance of P frame coding in the IBBP GOP structure. Getting high level accuracy in P frame coding is crucial since P frame also serves as the reference frame for intra non-reference (B) frames coding. As discussed in section 3, according to the structure of GOP, the reference frames of P are very much further compared with B, requiring wider search window or hierarchical way of searching in order to maintain the optimum accuracy. Again, higher accuracy of ME in P frame coding yields better picture quality which in turn gives lower B frame weight in motion compensation process.

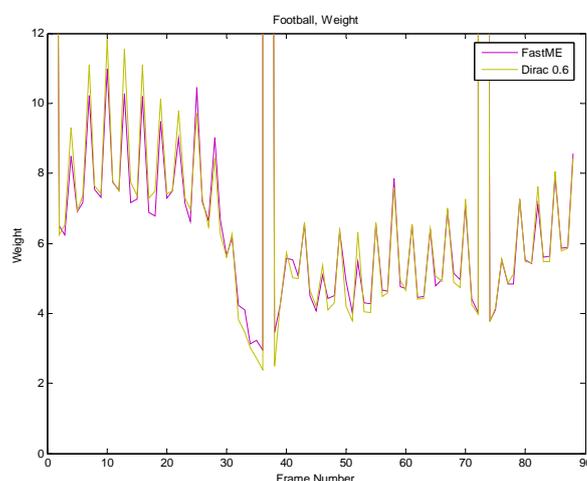


Fig. 12. Comparison of Weights for each frame, Football Sequence, CIF format

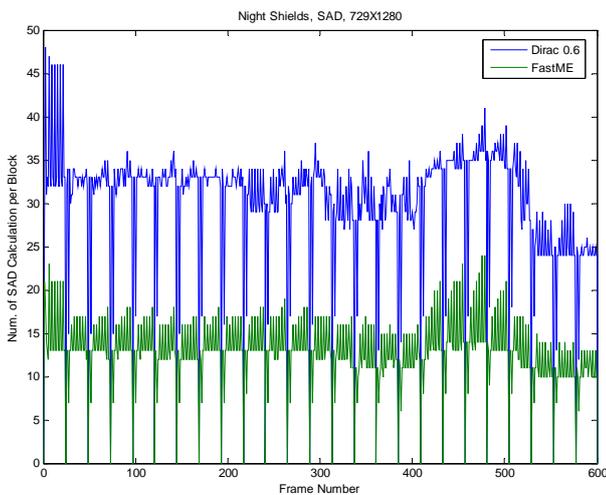
Table 2 shows the motion estimation results from both Dirac 0.6 and proposed algorithm for two types of HD sequences. While the value of PSNR, compression efficiency (in terms of encoded file size) and accuracy of ME (in terms of residual error frame's weight) are approximately the same, the proposed fast ME algorithm requires very much lower number of SAD calculation compared with

existing algorithm in Dirac 0.6. Again, there is at least two fold reductions in the number of SAD calculation for the HD sequences. The results in Table 2 also show that the reduction in SAD calculation is higher in the relatively static sequence (Night Shields) and the more dynamic motion sequence (Pedestrian Area) gives approximately two folds reduction only.

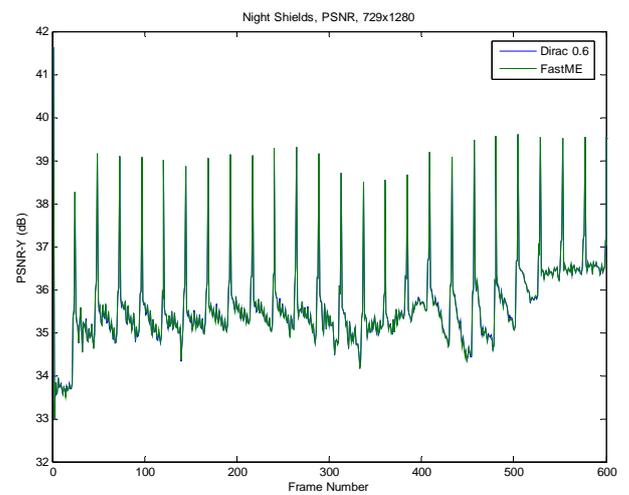
Fig. 13 and 14 show the number of SAD calculation per block and PSNR-Y of each frame for both HD test sequences shown in Table 2. Again, the results in these figures are very much similar to the CIF results in Fig. 8 to 11.

Table 2. The Comparison of ME Results for HD Video format

| Sequence | Algorithm | SAD | Weight | File Size (bytes) | PSNR-Y (dB) |
|---------------------------------|-----------|-------|--------|-------------------|-------------|
| Night Shields 729x1280 | Dirac 0.6 | 29.80 | 5.87 | 7473176 | 35.66 |
| | Fast ME | 12.81 | 5.87 | 7346537 | 35.66 |
| Pedestrian Area 1080x1920 | Dirac 0.6 | 31.09 | 10.22 | 7424520 | 38.09 |
| | Fast ME | 16.98 | 9.98 | 7334674 | 38.16 |

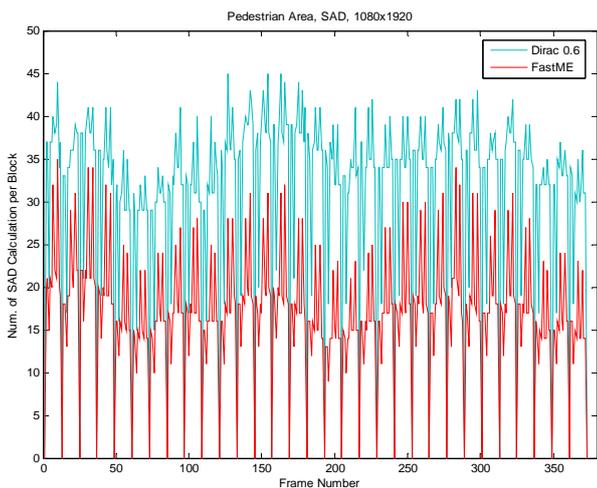


(a) Comparison of Average Number of SAD Calculation

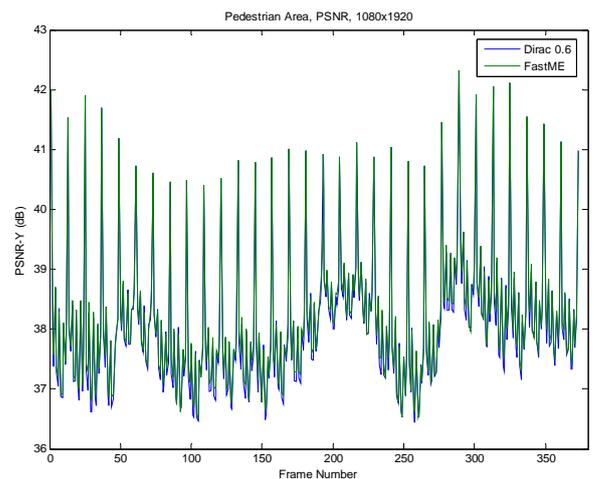


(b) Comparison of PSNR-Y

Fig. 13. Comparison of ME results for Dirac 0.6 and Proposed, Night Shields, HD 729x1280



(a) Comparison of Average Number of SAD Calculation



(b) Comparison of PSNR-Y

Fig. 14. Comparison of ME results for Dirac 0.6 and Proposed Fast ME, Pedestrian Area, HD 1080x1920

5 Conclusion

In this paper, an algorithm using semi-hierarchical way of motion estimation is proposed. It is the modification of the existing one with the addition of more initial search points in both temporal and spatial domain. By using unequal level of hierarchy for the different types of inter frames, the algorithm reduces the overall complexity effectively. The proposed strategy outperforms very well compared with the existing Dirac 0.6 version as shown in Table 1 and 2. It offers huge saving, at least two folds, in terms of the average number of SAD calculation per block for motion ranging from medium to high and more than one third saving for the static motion sequence. Moreover, the proposed algorithm has the ability to increase or decrease the search range depending upon the complexity of the motion in order to maintain the accuracy of the motion estimation to a certain level. Finally, it is obvious to see that the application of the proposed semi-hierarchical way of motion estimation will certainly reduce the complexity of the motion estimation algorithm and can be used in any type of standard video encoder.

Acknowledgments

The authors gratefully acknowledge the financial support from the BBC R&D, UK and Brunel University, UK. The authors would also like to thank Dr Thomas Davies and Dr Tim Borer from the BBC R&D for their insightful and constructive comments on this work.

References:

- [1] 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, pp. G.5.3.1-G.5.3.5, 1981.
- [2] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438-442, 1994.
- [3] L. M. Po, and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313-317, 1996.
- [4] Shan Zhu, and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp.287-290, February 2000.
- [5] Yao Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp.1442-1448, December 2002.
- [6] Yao Nie, and Kai-Kuang Ma, "Adaptive Irregular Pattern Search with Matching Prejudgment for Fast Block-Matching Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp.789-794, June 2005.
- [7] H. Nam, S. Lim, "A New Motion Estimation Scheme Using a Fast and Robust Block Matching Algorithm" *WSEAS Trans. on Information Science & Applications*, Issue 11, Volume 3, November 2006, pp. 2292-2299.
- [8] Y. Shi, C. Yi and Z. Cai, "Multi-Direction Cross-Hexagonal Search Algorithms for Fast Block Motion Estimation," *WSEAS Trans. on Computers*, Issue 6, Volume 6, June 2007, pp. 959-963.
- [9] C. L. Lin, J. J. Leou, "An Adaptive Fast Search Motion Estimation Algorithm for H.264" *WSEAS Trans. on Communications*, Issue 7, Volume 4, July 2005, pp. 396-406.
- [10] BBC Research & Development, "Dirac Video Codec". [Online]. Available: <http://dirac.sourceforge.net/>