A SPATIO-TEMPORAL RECURSIVE SEARCH BASED PREDICTION SCHEME FOR EFFICIENT MULTI-FRAME AND BIDIRECTIONAL MOTION ESTIMATION

Ahtsham Ali, Nadeem A. Khan

Department of Electrical Engineering, Lahore University of Management Sciences
DHA 54792, Lahore, Pakistan
{ahtsham.ali, nkhan}@lums.edu.pk

ABSTRACT

In this paper a new multi-frame/bidirectional motion estimation algorithm based on the concept of recursive-search in spatial and temporal space is proposed that effectively minimizes the number of candidate motion vectors in the prediction set. The algorithm is extremely computationally light compared to full search and outperforms existing fast approaches. The algorithm can operate effectively on reference frames that may be chosen to be consecutive or temporally separated. The average Peak Signal-to-Noise Ratio (PSNR) performance and compression efficiency is virtually not compromised and very close to full search. Results based on a number of video sequences and on different GOP (Group of Pictures) structures are presented to clearly demonstrate the benefits of the proposed motion estimation technique.

Index Terms— Motion Estimation (ME), Multi-frame Motion Estimation, Bi-direction Prediction, DIRAC Wavelet Based Video Codec, 3D Recursive Search

1. INTRODUCTION

Video encoders play an important role in Multimedia Systems and heavily draw upon the hardware resources required for multimedia computing. Modern encoders like H.264/AVC (a DCT based codec) or DIRAC (a wavelet based codec) make use of variable block sizes and multiple reference frames for motion estimation (MRFME) in order to deliver higher coding efficiency. This makes motion estimation a very computationally intensive task. Several fast motion estimation techniques have been proposed for this purpose.

In [6] a fast bidirectional approach has been presented in the context of H.264 encoder supporting four prediction modes based on fast evaluation of Lagrangian cost function but based on full search. In [1], Tun *et. al* proposes a multi-frame/bidirectional semi-hierarchical Fast ME algorithm in the context of DIRAC video encoder which is a motion compensated Wavelet video encoder. This approach is an improvement over the original fully hierarchical motion estimation technique of DIRAC described in [1]. The semi hierarchical motion estimation approach of Tun *et. al* [1] uses fully hierarchical motion estimation only for a certain type of inter frame. The experimental results show that the proposed algorithm reduces the number of SAD calculation to two to three times as compared to the original motion estimation algorithm of DIRAC while providing almost the same PSNR performance.

Use of spatial and temporal prediction can effectively avoid the use of large search windows to develop low complexity approaches based on single resolution. UMHexagonS is one such high performance fast motion estimation (FME) approach introduced in the context of H.264 codec. The UMHexagonS initially evaluates a set of motion vector predictors and then searches using a variety of hexagonal search patterns.

The true MRFME algorithms should be able to exploit the temporal correlation between multiple reference frames for higher performance. One such advanced algorithm designed with this perspective has been presented in [3]. The proposed algorithm uses 3-D search pattern encompassing consecutive neighboring reference frames. This search is centered on the best MV predictor obtained from an earlier step where a set of candidate spatial, temporal and upper layer blocks MV (motion vector) predictors are evaluated to find the best MV predictor. In case of low motion, this 3D search is curtailed to small diamond search. In case of high-motion the regular 3D-search is followed by a Multi-Hexagon-Grid search which is again followed by a second pass of 3D search.

This paper presents a prediction based MRFME approach that exploits spatial and temporal correlations based on recursive search principle and demonstrate how it yields a high performance yet a lower complexity algorithm compared to state-of-art approaches. In comparison with [3] both high and low motion are dealt with the same set of predictors and do not require reference frames to be consecutive. It can cope with GOP structures in which encoding order is different from the actual video frame order for example as in temporally scalable video.

The rest of the paper is organized as follows: Section 2 presents the recursive method for generation of prediction. Section 3 presents the complete variable block size (VBS) Multi-frame/bidirectional approach. In Section 4 results and experiments are presented and the performance is discussed in comparison with other schemes. Section 5 concludes the paper.

2. PROPOSED SCHEME FOR PREDICTION GENERATION

The effectiveness of recursive search approaches in motion estimation has been shown in case of single frame forward motion estimation predictive schemes [4], [5] for both DCT and Wavelet based codec. However, these algorithms are meant for single reference frame motion estimation and also depend on single continuous chain of forward motion estimated frames which is not possible in multi-frame/bidirectional motion estimation.

The proposed scheme is based on the idea of 1-D recursive search [4] method which is explained with reference to Fig 1. It involves recursively optimizing a previously found motion vector of a spatial neighbour termed as ‘spatial predictor’. It assumes that

the discontinuities in the velocity plane are spaced at a distance that enables convergence of a recursive block matcher in between two discontinuities. When this assumption is satisfied, the recursive block matcher yields the correct value at the first side of the object boundary and starts converging at the opposite (second) side. Treating this as the convergence direction, either side of the contour can be estimated correctly depending on the convergence direction chosen. If two estimators are applied with opposite or at least different convergence directions (bi-direction convergence) the fast step response (in convergence to actual motion vector) as required on the boundary of the moving objects can be achieved. This is because one of the estimators would have converged already at a position where the other has yet to do so. A number of times it may not be possible to define an estimator in the opposite direction because of non availability of results for the blocks after the current block in the scanning order which are yet to be coded. Convergence in such cases can be improved if a previously calculated result (for example from the previous MV field) at an opposite position is used as an additional predictor termed as 'convergence accelerator' in the same estimator. This 'look-ahead' feature can improve convergence behaviour based on the underline assumption that the displacement between the previously evaluated velocity plane (e.g. for a previous frame) and the current frame due to movements in picture is small compared to block size.

Our approach defines a 3-D recursive search in the spatio-temporal space. This involves two spatial estimators and one temporal estimator/predictor for defining the candidate predictor set for motion estimation on a given reference frame. This is illustrated in Fig. 2 in which 'X' is the current block for which motion estimation is being performed. The two spatial accelerators *a* and *b* based on spatial predictions S_a and S_b (Fig. 3) respectively, taken from block locations 7 and 9 respectively are defined. These are defined in two perpendicular directions that are diagonally oriented so that the updated motion vectors of these neighbours are available. The ideal positions of blocks to provide the 'convergence accelerators' for these estimators are 18 or 24 for estimator *a* and 16 or 20 for estimator *b*. Because of the fact that these are temporal predictors derived on previous field results (which may even be temporally separated), their affectivity may get reduced. In case of accelerator *b* a better compromise is offered by selecting an offset position 12 that has been updated for the current motion vector field. Though it is not diagonally opposite but does have a component in the direction opposite to S_b.

This motion estimation process can be supported in case of first reference frame by a third temporal estimator *c* as shown in Fig. 2(b), whose temporal predictor T_c can be taken from one of the closest previously evaluated motion vector field. The functioning of this estimator is the same as a spatial estimator expect that its convergence direction is along the temporal axis and the object boundaries are replaced by the temporal transition of the block from one object to another. Note that a block located at a certain position in a frame may belong to one object or the other with passing time owing to movement of the objects. We assume that such transition takes place at such intervals so as to allow a recursive search process to complete its convergence to new motion after object transition. Defining such an estimator is useful especially where a static background exists in a scene in multiple frames. This temporal estimator may be substantiated by a 'convergence accelerator' drawn from a future frame. However, this is not possible most of the times as the motion estimation of a future frame may not have taken place by this time. In motion estimation on subsequent reference frames it is more attractive to use the latest motion vector result of the current block w.r.t 1st reference frame. This then constitute a temporal predictor T_c for use as one of the candidate.

Thus with reference to Fig. 3 candidate set, *CS*, used for motion estimation is given by:

$$CS = \{ S_a, C_a, S_b, C_b, T_c \}$$

Where C_a and C_b are 'convergence accelerators' of estimators *a* and *b* respectively. Note that while C_a is a temporal predictor, C_b is chosen to be a spatial predictor.

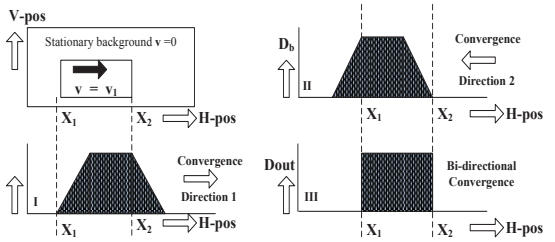

Fig. 1. Bidirectional convergence principle [4]

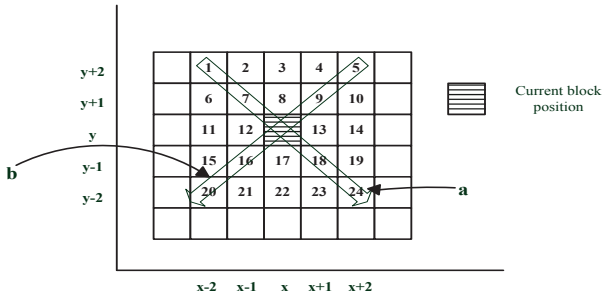

Fig. 2(a). Spatial orientation of the estimator *a* and *b* w.r.t. the current block. Arrows indicate the convergence direction.

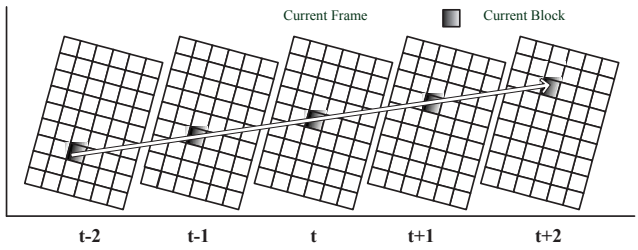

Fig. 2(b). Temporal orientation of the estimator *c*. Arrows indicate the convergence direction.

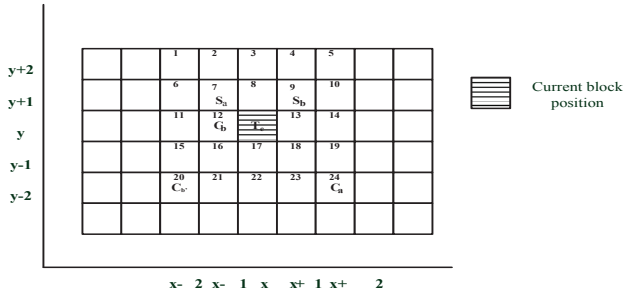

Fig. 3. The relative position of prediction candidates (spatial and temporal) in the current/previous frame based on estimators *a*, *b* and c

The method for generating the required temporal predictors from the available motion vector results (of the current frame or one of the previous frames) is illustrated with an example GOP structure of Fig. 4. Here the distance between anchor frames is 3. With the exception of the first P-frame in the GOP, the motion compensation for both types of frames involves two reference frames. P-frame, however, involves forward motion estimation only while B-frame involves one previous and one future reference frame. Note that in the illustrated GOP structure, a variety of locations of reference frames that are not consecutively placed are involved. In case of consecutive reference frames, without loss of generality, the same process has to be repeated on all frames. Table 1 show against each reference frame the calculated temporal

Table 1. Temporal predictors (C_a, T_c) for IBBP... Case

Coded Frames	Reference Frame for ME		Temporal Predictor values
I_0	NA		NA
P_1	1st Ref	I_0	$C_a = 0;\ T_c = 0$
	2nd Ref	NA	NA
B_1	1st Ref	I_0	$C_a = 0;\ T_c = 0$
	2nd Ref	P_1	$C_a = -2Mv_{24}^{B_1I_0};\ T_c = -2Mv_x^{B_1I_0}$
B_2	1st Ref	B_1	$C_a = Mv_{24}^{B_1I_0};\ T_c = Mv_x^{B_1I_0}$
	2nd Ref	P_1	$C_a = -Mv_{24}^{B_2B_1};\ T_c = -Mv_x^{B_2B_1}$
P_2	1st Ref	P_1	$C_a = Mv_{24}^{P_1I_0};\ T_c = Mv_x^{P_1I_0}$
	2nd Ref	I_0	$C_a = 2Mv_{24}^{P_2P_1};\ T_c = 2Mv_x^{P_2P_1}$
B_3	1st Ref	P_1	$C_a = \frac{1}{3}Mv_{24}^{P_1I_0};\ T_c = \frac{1}{3}Mv_x^{P_1I_0}$
	2nd Ref	P_2	$C_a = -2Mv_{24}^{B_3P_1};\ T_c = -2Mv_x^{B_3P_1}$
B_4	1st Ref	B_3	$C_a = Mv_{24}^{B_3P_1};\ T_c = Mv_x^{B_3P_1}$
	2nd Ref	P_2	$C_a = -Mv_{24}^{B_4B_3};\ T_c = -Mv_x^{B_4B_3}$

prediction value. In the notation Mv_B^{FiFj}, Mv_B represents the motion vector of a block 'B' as numbered in Fig. (3) of frame F_i that has been motion estimated w.r.t. to the reference frame F_j. In case of I reference frame, the temporal predictors are simply taken as zero motion vector to initialize the recursive search. For all other cases appropriate scaling is done of the motion vector taken from the corresponding location but from a motion vector field already evaluated. For forward reference frame, the scaling factor is positive and for backward reference frame, the scaling factor is negative. For example in case of B_1, the temporal distance of B_1 to its 2nd reference is 2 and the temporal distance of P_1 to its 1st reference is 3. Therefore, P_1 motion vector is multiplied by -2/3 as a temporal candidate vectors. Similarly, for B_2 the motion vectors taken from P1 are multiplied by -1/3.

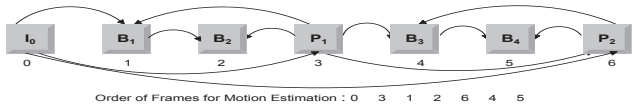

Fig. 4. Proposed Motion Estimation Scheme

3. THE MULTIPLE REFERENCE FRAME MOTION ESTIMATION ALGORITHM

The proposed prediction method has been employed for implementing a Variable Block Size (VBS), Multiple Reference Frame motion estimation algorithm in the DIRAC wavelet based motion compensated codec, which is based on the same paradigm as the standard DCT based codec by replacing DCT with DWT. Thus, the role of the motion estimator essentially is the same in both codecs. Different GOP structures and GOP lengths are selectable according to required time constraints. The motion estimation accuracy goes to ¼-pixel. The mode selection/block size selection employed in DIRAC [1] uses at Splitting level 0 (8x8) 'blocks', Splitting level 1 (16x16) 'sub-superblocks' and at Splitting level 2 (32x32) 'superblocks' with corresponding motion vectors. There are four prediction modes available at each of these levels: Intra coded, only predicted from the first reference, only predicted from the second reference and predicted from both first and second reference frame by taking the average of two blocks.

The motion estimation comprises three stages: In the first stage, motion vectors are calculated for every 'block' of each frame to one pixel accuracy. In the second stage, these vectors are refined to sub-pixel accuracy. In the third stage, mode decision is performed: This begins at the lowest 'block' level where the full-pixel best motion vector is determined in two steps: A coarse search and a fine search/refinement process. The coarse search stage involves 3D recursive search as discussed in the previous section 3. The motion vector is refined in the second step by conducting a one pixel local diamond search. An early stopping criterion has been incorporated to skip the refinement process when the corresponding SAD (Sum of Absolute Difference) is already less than a threshold value.

The sub-pixel refinement and mode decision process use an approach similar to what is employed by the original DIRAC 1.0.2 ME algorithm [1]. This is not a necessary choice but has been done for the sake of fair experimental comparison of our approach with that of previously proposed schemes in DIRAC coder. The sub-pixel refinement process uses the full pixel motion vector as the initial guide for finding ½-pixel accuracy motion vectors which in turn are used as a initial guide for ¼-pixel accuracy. At each stage small diamond search is used to find the best sub-pixel motion vector. In the mode decision, these 'block' level motion vectors are used as candidate motion vectors to evaluate best motion vector at 'sub-superblocks' which in turn are used at the 'superblock' level. For finalizing the prediction mode, the costs for intra prediction and other prediction modes at each level is made available out of which the one yielding the best SAD is selected.

Furthermore, note that early termination has been used in our algorithm (and the others tested) to reduce computations. This means that the SAD calculation is terminated prematurely as soon as the intermediate SAD value exceeds the current minimum SAD.

4. EXPERIMENTS AND RESULTS

DIRAC (version 1.0.2) has been employed for our current experiments and testing as it also opens avenues for our future research on new scalable video approaches. For exact and fairer assessment of the capability of our motion estimator, we have implemented full search and a few other motion estimator in same codec. The results of our approach are shown in the Tables 2 and 3 for different videos and compared with original DIRAC 1.0.2 ME, Fast ME and full search approaches. Different GOP sizes and structures but using 2 reference fames as in Fig. 4 are used. The result shows that the proposed approach has reduced the ME time and total encoding time by 40% -50% and 30% -40% respectively compared to DIRAC 1.0.2 ME and 93% - 94% and 85% - 86% respectively compared to full search while maintaining the PSNR-Y close to Full search. The window size in case of full search is ±8

Table 2. The Comparison of Motion Estimation (ME) for CIF with GOP Size = 12, B Frames = 6 (Distance Bet. Anchor Frames = 2); ¼ pixel, Variable Block Size Case

SEQUENCE	ALGORITHM	FILE SIZE (KB)	PSNR-Y (DB)	AVG(SAD)/ BLOCK	%ME TIME W.R.T DIRAC1.0.2 ME	%TOTAL TIME W.R.T DIRAC1.0.2 ME
FOREMAN	DIRAC 1.0.2 ME	1475	37.7730	70	---------	---------
	PROPOSED 3DRS	1436	37.6557	24	51.83%	63.65%
CONTAINER	DIRAC 1.0.2 ME	1015	37.7635	44	---------	---------
	PROPOSED 3DRS	962	37.8283	13	48.79%	60.78%
HIGHWAY	DIRAC 1.0.2 ME	1119	39.5129	61	---------	---------
	PROPOSED 3DRS	1043	39.3916	24	60.69%	70.50%

Table 3. The Comparison of Motion Estimation (ME) for QCIF with GOP Size = 18, B Frames = 12 (Distance Bet. Anchor Frame = 3) (With Full-Pixel and Variable Block Size Case)

SEQUENCE	ALGORITHM	FILE SIZE (KB)	PSNR-Y (DB)	AVG(SAD)/ BLOCK	%ME TIME W.R.T DIRAC1.0.2 ME	%TOTAL TIME W.R.T DIRAC1.0.2 ME
FOREMAN	DIRAC 1.0.2 ME	411	35.8203	59	---------	---------
	FAST ME	440	35.3764	36	64.53%	72.44%
	PROPOSED 3DRS	410	35.8062	13	37.95%	56.60%
	FULL SEARCH	415	35.9126	1842	564.73%	390.85%
CONTAINER	DIRAC 1.0.2 ME	226	37.3489	43	---------	---------
	FAST ME	220	37.2969	12	53.56%	71.68%
	PROPOSED 3DRS	216	37.6411	6	36.57%	58.44%
	FULL SEARCH	228	37.3572	1842	675.45%	430.73%
HIGHWAY	DIRAC 1.0.2 ME	281	38.3707	49	---------	---------
	FAST ME	280	38.1008	32	69.18%	76.67%
	PROPOSED 3DRS	267	38.1825	12	40.55%	61.07%
	FULL SEARCH	283	38.4963	1842	740.90%	482.13%

pixel when the reference frame is located at one temporal distance. Fig. 5 shows the bitrate vs PSNR graphs for Foreman and Container Sequence of CIF format at 30 frames per second. The graphs clearly demonstrate that the PSNR values of proposed 3DRS at different bitrates are nearly equal to as compared to other approaches.

Though implemented in different codecs, relative performance of our algorithm with UMHexagonS[7] and Predictive 3D Search Algorithm[3] can be made in relation to Full Search approach. All the three approaches show a PSNR performance close to full search. For [3] taking the case of five reference frames and full pixel accuracy (and fixed block size), the number of search locations is 75 at best (i.e. 15 search locations per frame). Total search number rise to 106 if all types of searches in the search scheme are utilized at least once and yield 274 searches in case of high motion under the same assumption. In our case a maximum of 9 search points per frame are evaluated (if all predictors are different). This difference shall increase further if mode decision and mode level prediction candidates are also included in the above picture. In [7] for full pixel and fixed block size search in window of size ±16 pixels, the reported speedup of UMHexgonS algorithm over full search for different video sequences are 26 to 47 times. In our case under the same conditions this speed up factor comes out to be 121.

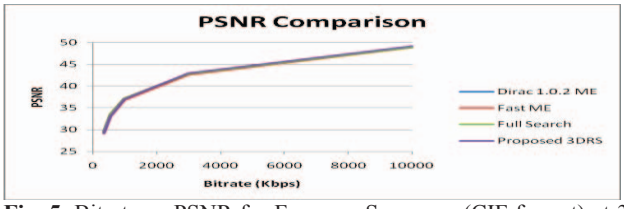

Fig. 5. Bitrate vs PSNR for Foreman Sequence (CIF format) at 30 frames per second

5. CONCLUSION

In this paper, a multi-frame/bidirectional algorithm based on the concept of 3D Spatio-Tempoal Recursive Search has been proposed. This approach is able to maintain a PSNR performance very close to full search yet yields a very low complexity motion estimator compared to existing approaches like [3]. Both high and low motion can be dealt with the same set of predictors and there is no requirement of reference frame being consecutive. The result clearly demonstrates the efficiency of our proposed scheme in a motion compensated codec framework. Our proposed algorithm is generic to support all GOP structures.

6. REFERENCES

[1] M. Tun, K. K. Loo, and J. Cosmas, "Semi-Hierarchical Based Motion Estimation Algorithm for the DIRAC Video Encoder," *WSEAS Transactions on Signal Processing*, Issue 5, vol. 4, pp. 261-270, May 2008.

[2] Xingyu Wen and Guiju Li, "Optimization on Motion Estimation Algorithm based on H.264", *In Proc.* 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, pages V5-590 - V5-593, 2010.

[3] Hong Yin Lim, Ashraf Ali Kassim, and Peter H.N. de With, "Predictive 3D Search Algorithm for Multi-Frame Motion Estimation", *IEEE* Transactions on Consumer Electronics, December 2008, Volume 54, pages 1938-1946.

[4] G. De Haan, P.W.A.C. Biezen, and O.A. Huijgen, "True Motion Estimation with 3-D Recursive Search Block Matching", *IEEE* Transactions on Circuits and Systems for Video Technology, October 1993, Volume 3, pages 368-379

[5] A. Ali, S.F. Ali, N.A. Khan, and S. Masud, "Performance Improvement in Motion Estimation of DIRAC Wavelet based Video Codec", *In Proc. ISCIT* , Icheon, South Korea, pp 764-769, 2009.

[6] W. Barreh, F. Tlili, A. Benazza-Benyahia, "Fast coding of bidirectional frame for H264 standard," *IEEE International Conference on Electronics, Circuits and Systems ICECS 2005,* Gammarth, Tunisia, December 2005.

[7] Qiu Xiao-bin, Huang Chun-qing, "An improved Algorithm of Fast Motion Estimation Based on H.264", *In Proc.* 5th International Conference on Computer Science and Education (ICCSE), Hefei, China, Pages 1717 – 1721, 2010.