A 3D SPATIO-TEMPORAL MOTION ESTIMATION ALGORITHM FOR VIDEO CODING

Gwo Giun Lee, Ming-Jiun Wang, He-Yuan Lin, Drew Wei-Chi Su, and Bo-Yun Lin

Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan

{ clee, n2894155, n2894157, n2693238, n2694447}@mail.ncku.edu.tw

ABSTRACT

This paper presents a new spatio-temporal motion estimation algorithm for video coding. The algorithm is based on optimization theory and consists of the strategies including 3D spatio-temporal motion vector prediction, modified one-at-a-time search scheme, and multiple update paths. The simulation results indicate our algorithm is better than other recently proposed ones under the same computational budget and is very close to full search. The low-cost feature and regular demand of computational resource make our algorithm suitable for VLSI implementation. The algorithm also makes single chip solution for high-definition coding feasible.

1. INTRODUCTION

Motion estimation (ME) produces motion vectors (MV's), which indicate the potential scene displacement, and thus removes the temporal redundancy for video compression. The video coding standards, such as MPEG-2 and H.264, adopt block-based motion estimation. The traditional coding standards perform ME on each 16x16 block, whereas recent popular coding standard, H.264, supports variable block size motion estimation (VBSME), which further enhances the coding efficiency.

ME demands much more computation than other modules in video encoders, making it the most crucial module. To reduce the complexity, sub-optimal solutions generally sacrifice the coding efficiency and save the complexity. The quality of ME algorithm directly affects chip area, power consumption, bus bandwidth, and the coding efficiency of encoders. Research of decades focuses on reducing the complexity of ME and maintaining the performance as much as possible. Comprehensive surveys are available in [1][2].

Intuitively, the algorithm that gives the best performance is to evaluate every search position in the predefined search range. This algorithm is full search (FS) [3]. Although this method finds the MV with the minimum distortion, the extremely high complexity raises difficulty in hardware implementation and is almost impossible to code high-definition (HD) video sequences with single chip.

In our previous work, we introduced a 3D recursive motion estimation algorithm and its architecture [4]. In this paper, we propose a ME algorithm in the 3D spatio-temporal sense by exploiting the optimization theory. We also propose a modified one-at-a-time search (OTS), which achieves fast convergence and saves computational resource. Multiple update paths are also the essence of our algorithm, providing good matching performance. We tested our algorithm against others with several critical test sequences and verified the low cost and high performance characteristics of our algorithm. The algorithm design is coherent with the corresponding architecture and is suitable for VLSI implementation.

2. THE PROBLEM OF ME AND ITS SOLUTION

The assumptions in deriving the simplified ME algorithms include: 1) rigid translation of objects; 2) Only one object in a block; 3) unimodal error function. These assumptions are very helpful to reduce the complexity of ME algorithms. Unfortunately, the video sequences in real world contain the following imperfection: 1) occlusion on objects and picture boundaries; 2) image deformation; 3) multi-objects in a block; 4) multi-modal in the error function of ME. FS outperforms other algorithms upon the fourth problem. The complexity-reduced ME algorithms commonly suffer from being trapped into local minimum. Spatio-temporal-styled algorithms overcome this difficulty by searching with good initial points, which are predicted by spatial and temporal neighboring blocks, along with some update scheme. Unfortunately, problems remain under inconsistent MV fields especially at rapidly changed motion. There is no clue to jump to the global minimum far away from the current search center. Several peaks in the error function usually exist on between the search center and the global optimum,

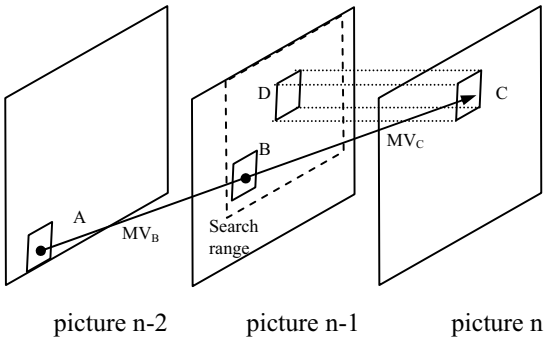

Fig. 1. 3D motion trajectory for motion prediction

and thus block the update scheme from moving to the global optimum location. The occlusion problem is solved more efficiently by bidirectional ME instead of FS. Limited by the currently prevailing ME model of coding convention, the deformation and multi-objects cannot be modeled by more than two parameters, yet the problem can be solved elegantly by VBSME. Therefore, the principle objective of general simplified algorithms should be the robustness under fast motion or scenes with multi-modal error function.

The spatial-temporal predicted ME algorithm is considered to be the most efficient approach in producing accurate MV's with quite simple computation. To remain robust under fast motion scenes, more clues are helpful to start with the best initial point. Hence, we inspected the property of video objects and obtained an important phenomenon. As illustrated in Fig.1, consider a rigid object traveling through block A, block B, and block C in the 3D motion trajectory. Due to the very short interval of time between successive pictures, the movement is assumed to be very regular. In this case, MV prediction from the co-located block D gives no help. On the contrary, the MV of block C is predicted from that of block B. The small fluctuation in velocity can be compensated by the subsequent update scheme. The temporal MV's from the enlarged prediction window give a key to jump to the global minimum of fast motion, which is the hardest problem in ME. This strategy benefits the prediction of inconsistent fast motion, which can be hardly predicted from traditional prediction manners. To design our ME algorithm from this observation, all of the temporal blocks within the search range are possible to be the candidates of initial search center.

3. PROPOSED ALGORITHM

Our proposed algorithm consists of spatio-temporal MV prediction, prioritized candidate list generation, modified OTS strategy, and multiple update paths. Each step is explained in the following.

First, a set of initial candidates are predicted from the spatio-temporal neighbors. These candidates include C_Z, C_S, and C_T, where the suffixes respectively denote zero, spatial, and temporal prediction. C_Z is the zero MV, which is important for stationary background. C_S are the MV's of immediate left and immediate top blocks. Adding the prediction of immediate top-right MV did not help much according to our experiment and it was hence discarded. These two spatially predicted MV's possess the highest correlation over other prediction blocks [2]. The spatially right and bottom MV's are unavailable due to causality. C_T are temporally predicted from all blocks within the search range in the reference picture. From the optimization's perspective, the key idea in the 3D spatio-temporal prediction is that extensive computation is reduced since we start the updating process from the highly correlated neighboring spatial and temporal blocks, which contain the results of optimization on error functions already minimized to a certain extent.

In the second stage, the candidates predicted in the first stage are sorted for further update. The most precious and expensive resource in ME is the SAD calculation and bus access, which need to be controlled and saved very carefully. The first stage produces more candidates than conventional methods to achieve good performance under formidable cases. Under frequently occurring stationary background and smooth motion field, it is not cost effective to compute the SAD of duplicate candidates. Hence, every incoming candidate from the first stage is compared with the candidate arrived earlier and is eliminated in case of duplication. The great amount of temporal candidates should also be filtered such that irrelevant prediction is eliminated and more search steps are saved for other update paths. The farther the temporal prediction block is from the center, the larger motion vector it should have. Otherwise this prediction is not considered. After the candidate filtering, every surviving candidate is then evaluated for its SAD and sorted. The smaller the SAD is, the more likely the candidate could be the final solution. We have to update the candidate earlier with limited computational resource.

In the third stage, the update strategy refines the initial MV candidates as shown in Eq. 1.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)} \tag{1}$$

We propose a modified OTS by exploiting the local gradient of error function. Traditional OTS [5] initially tests the two points next to the search center in opposite directions at the same time, which is not efficient from the local gradient's point of view. Hence, we propose to explore exact one of the following update directions at a time.

$$\{(-1,0),(0,-1),(1,0),(0,1)\} \tag{2}$$

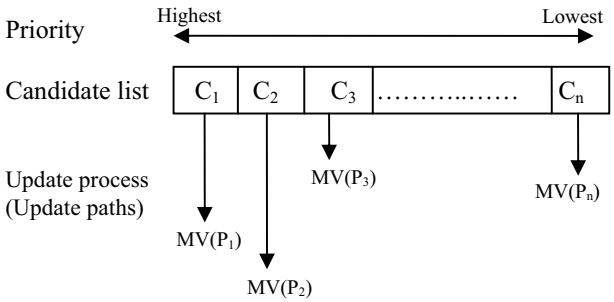

Fig. 2. Illustration of overall algorithm

The update engine randomly picks an update direction as $\mathbf{d}^{(k)}$ and evaluates the SAD of $\mathbf{x}^{(k+1)}$. The random pick avoid update biasing and can be implemented with linear feedback shift register. Smaller SAD manifests the fitness of the selected direction and we accordingly disable the opposite direction, which is supposed to be the ascent direction, from being chosen later on. Otherwise we disable the current direction and the opposite direction is possibly a descending direction. The update process continues until all of the four update directions are disabled or the search point touch the boundaries of search range. Furthermore, a lookup table is responsible for keeping the SAD of MV's in order to avoid SAD recalculation in the update stage.

The overall ME process is depicted in Fig. 2. The sorted MV candidates are placed in the priority queue and updated independently. The update steps that follow a certain candidate make up an update path. Whenever an update path is terminated, the MV and SAD are kept for final decision and the optimization process moves to the next candidate. Various lengths of update paths correspond to different numbers of update steps. Since hardware design looks for regularity, we setup a maximum number of search points (MNSP) for a block. ME for a block completes as soon as all of the update paths are evaluated or the MNSP is run out. The final MV is chosen among

$$\{MV(P_i)|i=1,2, \ldots, n\} \quad (3)$$

with the smallest SAD.

The overhead of our algorithm lies in the irregular MV matching, which seems to place extra burden on bus bandwidth. In practice, the bus bandwidth requirement is greatly reduced as a result of significant reduction of search steps. The update scheme also results in the spatial locality of memory access. Hence the data reusability is quite high. Sorting operation can be easily solved by conditioned shift registers and will not cause problems in hardware design.

4. SIMULATION RESULTS

We analyzed the cost and performance of the algorithms considered in this paper from the hardware's viewpoint. We first determined the MNSP per block for our algorithm. The MNSP per block with good compromise between cost and performance is 20 for CIF sequences and 35 for HD sequences. This implies that the complexity of our algorithm in terms of search step is 0.107% to 0.977% of that in FS. To validate the performance of our algorithm, it is compared to FS, and two recently proposed algorithms, namely, adaptive irregular pattern search (AIPS) [6], and enhanced hexagonal search (EHS) [7]. The search steps of AIPS and EHS vary from blocks to blocks and are not suitable for hardware implementation. Hence we slightly modified AIPS and EHS so as to make their MNSP the same as ours and left FS unchanged. The MNSP of all algorithms are shown in Table I. We excluded the video coding engine and calculated the PSNR alone so that the performance of various algorithms is directly observed by qualitative analysis.

Table I
Number of Search Steps

Algorithm	Resolution	Search Steps
FS	CIF	2048
	HD	32768
Ours, AIPS, EHS	CIF	20
	HD	35

Table II
Simulation Setup

Items	Setting
block size	16x16
horizontal search range (CIF)	[-32,32)
vertical search range (CIF)	[-16,16)
horizontal search range (HD)	[-128,128)
vertical search range (HD)	[-64,64)
PSNR calculation	without residue
MV over picture boundaries	valid

Table II shows the setup of our simulation. The algorithms were simulated on several sequences and the results of those highly vivid sequences, including Foreman, Stefan, Vectra color, Silent, Highway, Pedestrian area, and Traffic, are favored to differentiate the performance of the algorithms. To eliminate the interference of occlusion on the picture boundaries, MV's over picture boundaries are permitted in our simulation. This feature is supported by modern coding standard H.264.

Table III shows the simulation results in terms of PSNR. Our computational budget is the same as that of AIPS and EHS. However, our matching performance is obviously better than that of AIPS and EHS. If we further inspect the PSNR difference frame by frame as shown in Fig 3, the PSNR difference of ours and EHS can go up to 5dB in the sequence "Stefan". Although FS achieves the best matching result, a great deal of computation is further interpreted as bus bandwidth and power consumption in hardware, making FS in HD application almost infeasible.

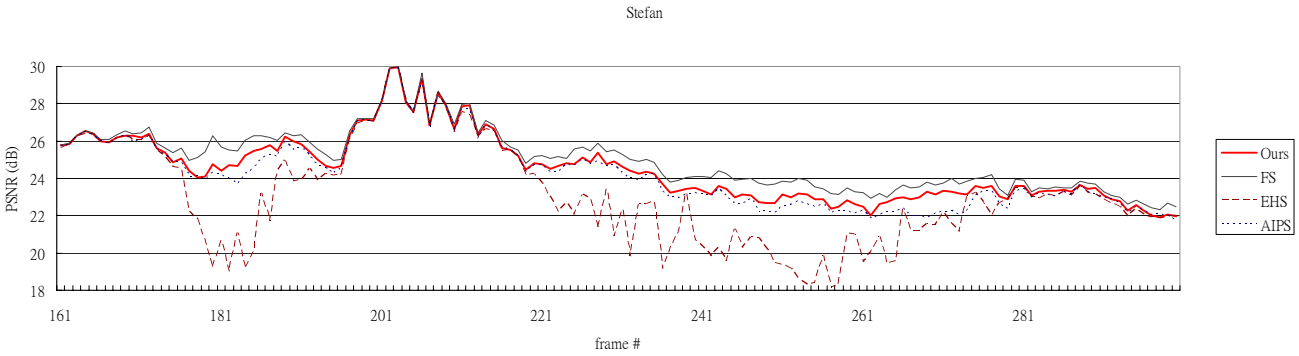

Fig 3. Frame-by-frame PSNR comparison of Stefan

Table III
Average PSNR of the Algorithms

Sequences	Algorithms	Average PSNR(dB)	PSNR Drop(dB)
Foreman 352x288@30fps 300 frames	**Ours**	**32.11**	**-0.41**
	FS	32.52	0
	AIPS	31.89	-0.63
	EHS	31.87	-0.65
Stefan 352x288@30fps 300 frames	**Ours**	**25.42**	**-0.31**
	FS	25.73	0
	AIPS	25.26	-0.47
	EHS	24.67	-1.06
Vectra color 352x288@30fps 142 frames	**Ours**	**26.73**	**-0.44**
	FS	27.17	0
	AIPS	26.39	-0.78
	EHS	26.05	-1.12
Silent 352x288@30fps 300 frames	**Ours**	**35.61**	**-0.52**
	FS	36.13	0
	AIPS	35.22	-0.91
	EHS	35.43	-0.7
Highway 352x288@30fps 300 frames	**Ours**	**35.28**	**-0.7**
	FS	35.98	0
	AIPS	34.58	-1.4
	EHS	34.82	-1.16
Pedestrian area 1920x1080@30fps 375 frames	**Ours**	**35.05**	**-0.69**
	FS	35.74	0
	AIPS	34.44	-1.30
	EHS	34.26	-1.48
Traffic 1920x1080@30fps 480 frames	**Ours**	**32.79**	**-0.16**
	FS	32.95	0
	AIPS	32.45	-0.50
	EHS	32.37	-0.58

5. CONCLUSION

In this study, we present a new spatio-temporal ME algorithm based on optimization theory. The uniqueness of our algorithm includes: 1) low cost implementation against FS; 2) special spatio-temporal MV prediction with large temporal reference window; 3) maximum likelihood selection for predicted candidates; 4) modified one-at-a-time search for update; 5) multiple update paths; 6) scalable complexity for various application; 7) suitable for hardware implementation; 8) better matching performance in terms of PSNR than other recently proposed algorithms under the same computational budget.

REFERENCES

[1] A. Chimienti, C. Ferraris, and D. Pau, "A complexity-bounded motion estimation algorithm," *IEEE Trans. Image Processing*, vol. 11, pp. 387-392, Apr. 2002.

[2] B. Montrucchio and D. Quaglia, "New sorting-based lossless motion estimation algorithms and a partial distortion elimination performance analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp.210-220, Feb. 2005.

[3] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe coding," *IEEE Trans. Commun.*, vol. 29, pp. 1799-1808, Dec. 1991

[4] G. G. Lee, K. A. Vissers, and B. Liu, "On a 3D recursive motion estimation algorithm and architecture for digital video SoC," Midwest Symposium on Circuits and Systems, vol. 2, pp. II449-II451, 2004.

[5] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Circuits Commun,* vol. COM-33, pp. 888-896, Aug. 1985.

[6] Y. Nie and K. Ma, "Adaptive irregular pattern search with matching prejudgment for fast block-matching motion estimation," *IEEE. Trans. Circuits Syst. Video Technol.*, vol.15, pp. 789-794, June 2005.

[7] C. Zhu, X. Lin, L. Chau, and L. Po, "Enhanced hexagonal search for fast block motion estimation," *IEEE. Trans.Circuits Syst. Video Technol.*, vol 14, pp. 1210-1214, Oct. 2004.