FAST BLOCK-MATCHING MOTION ESTIMATION BY RECENT-BIASED SEARCH FOR MULTIPLE REFERENCE FRAMES

*Chi-Wang Ting, Wing-Hong Lam and **Lai-Man Po

*cwting@ee.cityu.edu.hk, **eelmpo@cityu.edu.hk
Department of Electronic Engineering, City University of Hong Kong, Hong Kong,
83 Tat Chee Avenue, Kowloon, Hong Kong SAR of China

ABSTRACT

Multi-frame motion compensation improves the rate-distortion performance substantially but introduces much higher loading to the system. Without considering temporal correlations, conventional single-frame block-matching algorithms can be used to search multiple frames in a rather inefficient frame-by-frame way. In order to exploit the motion characteristic in long-term memory, a multi-frame extension of the well-known cross-diamond search algorithm is proposed. Unlike those algorithms that evenly search each reference frame, our algorithm adopts a novel recent-biased spiral-cross search pattern to sub-sample the 3-dimensional memory space as a whole. This approach significantly boosts the efficiency of the block-matching process. Two new techniques, *stationary block tracking* and *multiple searching paths*, are employed to further improve the speed and accuracy. As compared to full search, experimental results show that our algorithm can reduce up to 99.5% complexity in terms of searching points while limiting the PSNR loss in 0.04dB. Simulations also prove that our algorithm out-performs the cross-diamond search and diamond search algorithms in speed and accuracy.

Index terms—Motion estimation, multiple reference frames, recent-biased search, 3-dimensional search, H.264.

1. INTRODUCTION

Motion-compensated prediction is a technique to improve compression efficiency by referring pixels from reference frame(s) to a current frame, and hence reducing residual information. The process of finding the reference pixels is called motion estimation. One commonly adopted approach for that is block-matching algorithm in which frames are divided into a number of blocks for matching. A best-matched block is searched exhaustively by full search (FS) from reference frame(s) within a search range or search window (w) based on a block distortion measure (BDM) such as mean absolute error (MAE), sum of absolute differences (SAD) and other matching criterion. The displacement of the best-matched block is then represented by a motion vector.

The benefits of long-term memory motion compensated prediction (LTMCP) [1] have been emphasized in recent years. Consequently, this tool has been adopted by several recent standards like H.263+ and H.264/MPEG-4 AVC [2]. As continuously dropping the costs of semiconductors, notably higher prediction gain can be achieved by estimating more reference frames in the memory buffer. Nevertheless, an obvious drawback is the complexity will increase proportionally. Extra data are also needed to describe the reference indices. These make it becomes not feasible in most cases, such as low bandwidth communication and real-time encoding, particularly for software-based implementations. As a result, various methods were suggested tackling these problems. In this paper, we focus on solving the complexity problem. In general, the fast algorithms of LTMCP can be classified into 3 types: I) *Partial Distortion* - sub-sampling pixels from blocks for faster block distortion measurement; II) *Searching Point Reduction* - sub-sampling blocks from the search range for faster search convergence; III) *Frame Selection* - sub-sampling frames from the memory buffer to eliminate unrelated references. Based on real-world motion properties, type II algorithms are usually superior to others in terms of speed whereas their accuracy is significantly similar. For that reason, they are widely used in pre-existing video coding standards such as MPEG-1/2/4 and H.261/263. Diamond search (DS) [3][4], cross-diamond search (CDS) [5] and adaptive rood pattern search (ARPS) [6] are some more recent well-known algorithms of this type, but they are proposed for single-frame motion estimation only. Directly applying them to multiple frames cannot sufficiently exploit the temporal correlation in LTMCP. Unfortunately, seldom effort is put on extending these algorithms. The idea of generalizing these algorithms into N dimensions is first suggested in [7] that different transformations such as brightness and time can be regarded as an additional dimension. In this paper, we realize this idea by turning the cross-diamond search into 3-dimensional (3D) algorithm for LTMCP. A comprehensive analysis of motion vector distribution is conducted for multiple reference frames to support our novel recent-biased search method. A comparison of those non-modified methods are made to demonstrate the efficiency of our algorithm.

Format	Sequence (100 frames)
MPEG-4 (A) CIF	Akiyo, Hall Monitor, Mother & Daughter
MPEG-4 (B) CIF	Silent
MPEG-4 (C) CIF	Stefan, Table
CIF (352x288)	Sales
SIF (352x240)	Football

Table 1. Image sequences used for analyses and simulations.

2. RECENT-BIASED & CENTER-BIASED MOTION VECTOR DISTRIBUTION

The center-biased motion vector distribution has been leading the development of fast block-matching motion estimation algorithms for a long time. By inspecting the motion statistic, we can find out a motion model which can roughly represent the real-world motion behaviors. When motion estimation is shifted

0-7803-8554-3/04/$20.00

to multiple reference frames, it is necessary to create a new model and review the old characteristics. To demonstrate that, an analysis of global minimum motion vector distribution is conducted on the 8 well-known sequences listed in Table 1. They contain various real-world motion contents, such as zooming, panning, translation and slight movement. A FS scheme with a search window |w|=16 is applied to 5 reference frames to locate the "true" motion vectors. Blocks are matched at a fixed 16x16 macroblock size.

In our experiment, 3 different shaped areas, square, diamond, and cross, are analyzed and shown in Fig.1. All of them are located in the central 5x5 grid (radius |r|=2) of the search window. The results are tabulated in Table 2. Three motion vector distribution characteristics are observed and summarized as follow.

I. Motion vectors are biased to the center of the search window and the temporal distance does not affect this property. By comparing the amount of motion vectors in the central 5x5 region with the total amount of full range, we find that around 60% - 80% motion vectors are distributed in the central region (more condensed when closer to the center (0,0)). This behavior also exists in all reference frames individually.

II. The cross pattern is more effective to locate a significant amount of motion vectors as compared to diamond and square patterns. Due to the influence of gravity, most real-world motions are along a horizontal or vertical axis. From Table 2, we find that the amounts of motion vectors found in the 3 portions are very similar, 73.39% for cross, 76.08% for diamond and 79.31% for square. While the difference is relatively small, cross portion can save much more searching points than others (Fig.1). This effect also dominates individual reference frames.

III. Motion vectors are biased to more recent reference frame. It is clear that, for instance, translating objects would probably leave the search range after a certain time interval. The results show that, in general, the amount of motion vectors decreases exponentially along the time. For the full range results, about 50% motion vectors are obtained from the most recent frame F(t-1), 20% from the second recent frame F(t-2), 10% from the third recent frame F(t-3) and so on. That means the correlation between reference frames and the coding frame is decrease by time. This behavior is still obvious within the central 5x5 region.

3. RECENT-BIASED SEARCH

Based on the above observations, we propose a fast block-matching algorithm to fit the motion statistic. In this session, the search patterns adopted by our algorithm are described first, and the details of our search scheme are presented later.

3.1. Search Patterns

As mentioned above, our algorithm is a 3D extension of the cross-diamond search [5]. Four improved search patterns are shown in Fig.2. In this illustration, the *xy*-plane lies along with frames and the extra dimension *z* is the time axis.

To trace the motion in multiple frames, we need some search patterns that are able to move across frames. In Fig. 2(a) and 2(b), the visualization and search-point configuration of these patterns - Small 3-Dimensional Diamond (S3DD) and Large 3-Dimensional Diamond (L3DD) are shown. Their shape is symmetrical from any dimensions. A 3D diamond actually consists of 3 flat diamonds lying on the center of *xy*-, *yz*-, and *zx*-planes. The white spot in the figure indicates the center of the patterns. As they can move across frames to reach the minimum distortion point in the nearby frames directly, they are more effective to sub-sample the 3D space. This gives benefits over the conventional frame-by-frame approach in which a new search is started from the window center again for each reference frame.

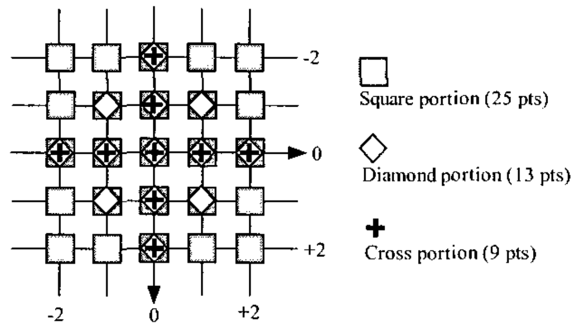

Fig 1. Three different shaped portions within central 5x5 grid.

	Motion Vector Distribution (%)				
Ref. Frame	Center	Cross	Diamond	Square	Full Range
t-1	31.66	37.89	39.32	41.52	52.73
t-2	13.49	15.44	16.00	16.32	20.26
t-3	4.32	5.64	6.00	6.29	8.78
t-4	9.02	9.78	9.99	10.20	11.60
t-5	3.87	4.64	4.77	4.99	6.22
Total	62.37	73.39	76.08	79.31	100.00

Table 2. This table shows the average probability of finding the global minimum motion vectors from different portions and reference frames where (t) is the time of coding frame, (t-1) for the previous frame and so on.

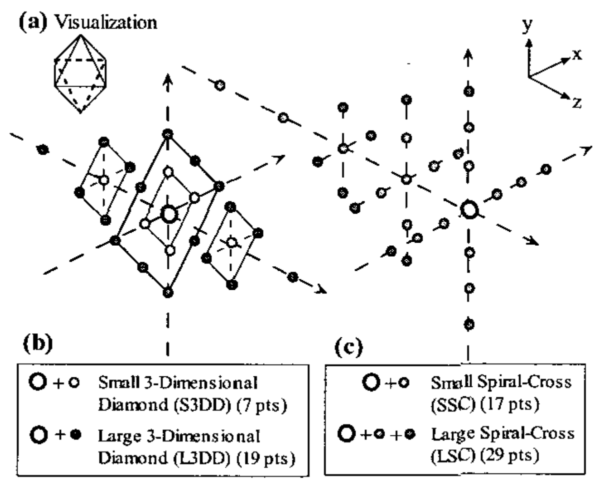

Fig. 2. 3D search patterns employed in our fast algorithm.

To reduce the complexity introduced by 3D search patterns, we need a precise initial guess of possible search directions. Fig. 2(c) shows the advanced cross patterns - Small Spiral-Cross (SSC) and Large Spiral-Cross (LSC). These 2 patterns are formed according to motion properties found in our previous analysis. Different sizes of center-biased crosses (|r|=0,1,2,3) are chained together to form a spiral-like shape. The larger side is on the most recent reference frame, and vice versa. So, they are called recent-biased spiral-cross. Because of the recent-biased property (over 80% motion vectors in most recent 3 ref. frames), spiral-cross is much more efficient than regular cross for locating the search directions.

3.2. Recent-biased Search Scheme

Just like many other block-matching algorithms, our method also assumes the block distortion error decreases monotonically towards the global minimum, but we extend this idea to both spatial and temporal domain. The main difference between the recent-biased search (RBS) and other conventional block-matching algorithms such as diamond search and cross-diamond search is that our scope covers the information obtained from several reference frames instead of just looking at the local statistic. By fitting the motion model we found, the speed and prediction accuracy of our algorithm can be boosted.

From our analysis in Table 2, we know that about 60% blocks are stationary blocks (r=0). One can reduce the complexity significantly by terminating the search in early stage if such blocks are detected. Based on this idea, a novel technique named *stationary block tracking* is proposed. It is incorporated with the SSC pattern to determine the existence of stationary blocks. A stationary block is found if the following equation is hold:

$$D = \sum_{i=1}^{N} \left(|M_i(x)| + |M_i(y)| \right) \le Th \qquad (1)$$

where D is the diffusion factor; N is the number of samples; M_i is the *i-th* minimum block distortion measure (BDM) point of SSC; $M(x)$ is the x-coordinate of M relative to the search window in which center is (0,0); and Th is a threshold value. In our experiments, we select N=5 and Th=0. In this case, 5 samples are taken from the SSC where they are the 1st, 2nd, ... and 5th minimum BDM points. Only if they all lie along the z-axis (i.e. x=0, y=0), then equation (1) is hold and the block is assumed to be stationary. It is very likely that non-stationary blocks would have those samples diffused apart from the center and biased to recent frames, and hence introduce a larger diffusion factor. A negative threshold value simply means to disable this function. Our detection technique makes use of the temporal correlation from multi-frames. Here are the detailed steps of the RBS scheme.

Step 1: (Small Spiral-cross Searching)
The center (white spot depicted in Fig.2) of the small spiral-cross (SSC) pattern is aligned to the center of the search window. The BDM values of totally 17 searching points over 5 reference frames are checked. Stationary block tracking is applied (N=5, Th=0). If stationary block is found, set the motion vector *mv* to the minimum BDM point and stop searching. Otherwise, go to Step 2.

Step 2: (Large Spiral-cross Searching)
Again the large spiral-cross (LSC) pattern is aligned to the center. A minimum BDM point is found from the 29 searching points (i.e. check 12 points more).

Step 3: (Large 3-Dimensional Diamond Searching)
A new large 3-dimensional diamond (L3DD) pattern is formed with the center located in the minimum BDM point found from the previous step. If the new minimum BDM point found from the L3DD is in the center of the pattern (i.e. convergence), then go to Step 4. Otherwise, this step is repeated.

Step 4: (Small 3-Dimensional Diamond Searching)
With the minimum BDM point found from the previous step as the center, a small 3-dimensional diamond (S3DD) pattern is formed. Identify a minimum BDM point from S3DD and it is the final *mv*.

The RBS algorithm has 2 stages. The first stage is to locate the possible direction and frame location. An early termination may occur in Step 1 to make a minimum number of 17 searching points. Step 2 enlarges the spiral-cross to capture more information owing to diffusion of motions. The second stage is to get close to the optimal point. A large sampling grid is used recursively in Step 3 for this purpose. In case all surrounding points have larger BDM than the center, it probably means we reach a sub-optimal point, and so Step 4 converges the search. Note that the center of our search patterns are restricted within the search window, but part of the surrounding points may move outside the window or beyond the frame buffer, then these searching points are ignored. Our algorithm emphasizes the speed performance, at the same time, maintains a reasonable prediction gain for various motion contents.

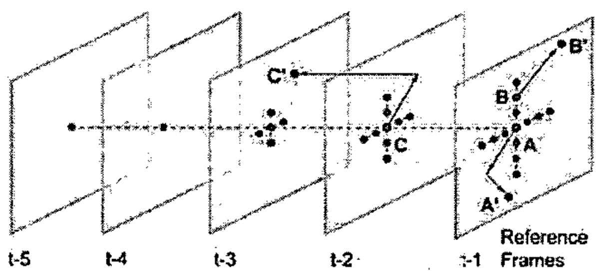

Fig. 3. Example of multiple searching paths with p=3.

An optional amendment to our algorithm named *multiple searching paths* is also proposed. Since the data volume increases with number of reference frames, sub-sampling the search range becomes easier to be trapped by a local optimal. The purpose of this amendment is to increase the prediction gain by initiating multiple searching paths. An example is shown in Fig.3, three minimum BDM points, A, B and C, are selected from LSC (Step 2) instead of one. For each point, a normal 3-dimensional diamond search routine (Step 3 & 4) is performed as usual. As a result, 3 candidate points, A', B' and C', are obtained eventually. Of course, the one with minimum BDM will be the final *mv*. Sometimes, different starting points may lead to the same destination. In this case, the complexity growth will be very limited due to the overlap of searching paths. The number of searching paths p is adjustable to make a tradeoff between complexity and prediction gain.

4. SIMULATION RESULTS

To demonstrate the performance of RBS, simulations using full search (FS), diamond search (DS), cross-diamond search (CDS) and RBS (N=5, Th=0, p=6) are performed on the luminance component of the 8 sequences listed in Table 1. The maximum search range is set to $\pm$ 16 pixels and the mean absolute error (MAE) is used as the BDM function. The block size is fixed at 16x16. While the number of allowed reference frame is set to 5, those single-frame algorithms will search the reference memory frame by frame. Although the simulations are not done within a real system, the results could still show a relatively precise performance comparison and valuable motion analysis. Since the purpose of this paper is to demonstrate the effectiveness of our motion model and the recent-biased search scheme, only PSNR and searching points are considered. Other evaluations such as cost of motion vector and reference indices are out of our scope, but their corresponding schemes and analysis are already well documented in [1][2].

The experimental results are tabulated in Table 3 by three testing criterion - average PSNR per frame (PSNR), average searching points per block (Pts) and speed improvement ratio (SIR). It shows that the search-point complexity of RBS is always much lower than other algorithms. In some particular sequences where most of the motion is gentle and smooth, the SIR can be up to 100 ~ 200, for example, 193 for *Akiyo*, 127 for *Sales*, 124 for *Silent* and 114 for *Hall Monitor*. The amazing speed up ratio is due to successfully terminating the search in early stage by our stationary block tracking method. Besides, RBS has higher PSNR gain as compared to DS and CDS for all sequences except *Akiyo*, in which RBS only has 0.02dB less than that of DS. Undoubtedly, there must be some reduction of PSNR gain when comparing a lossy algorithm to FS. However, this reduction is relatively small in RBS. For small motion sequences, *Akiyo, Hall Monitor, Mother & Daughter and Sales*, the reduction is around 0 ~ 0.2dB, for medium motion sequence, *Football and Silent*, around 0.4dB, and for large motion sequences, *Stefan and Table*, around 1 ~ 1.6dB. Therefore, our RBS algorithm out-performs DS and CDS in terms of accuracy and speed. It is excellent for video-conferencing application, while for large motion video, a satisfactory tradeoff between complexity and prediction gain can also be made.

5. CONCLUSION

In this paper, a novel recent-biased search algorithm is proposed together with an in-depth motion analysis in multiple reference frames. Such an analysis will be very useful in developing multi-frame motion estimation algorithms for various applications. Simulations prove that no matter speed or accuracy our recent-biased approach is better than using DS and CDS to evenly search multiple frames. By exploiting the motion characteristics in spatial and temporal domain simultaneously, up to 99.5% computations can be saved while keeping similar PSNR gain as compared to FS. This ultra-low complexity algorithm is highly suitable for real-time video applications, particularly for software-based implementations e.g. video conferencing. A more realistic experiment on H.264 reference software will be carried out on our next work.

6. ACKNOWLEDGMENTS

The authors would like to thanks Dr. Chun-Ho Cheung for his valuable suggestions and comments. The work described in this paper was substantially supported by a grant from City University of Hong Kong, Hong Kong SAR, China. [Project No.7001385]

Akiyo				Football			
	PSNR	Pts	SIR		PSNR	Pts	SIR
FS	43.39	4924.60	1.00	FS	26.29	4868.52	1.00
DS	43.37	63.07	78.08	DS	25.86	95.76	50.84
CDS	43.34	46.20	106.60	CDS	25.83	88.45	55.04
RBS	**43.35**	**25.53**	**192.90**	**RBS**	**25.92**	**65.37**	**74.48**
Hall Monitor				Mother & Daughter			
	PSNR	Pts	SIR		PSNR	Pts	SIR
FS	34.70	4924.60	1.00	FS	40.70	4924.60	1.00
DS	34.59	66.89	73.62	DS	40.52	74.56	66.05
CDS	34.56	51.25	96.09	CDS	40.49	61.88	79.58
RBS	**34.61**	**43.10**	**114.25**	**RBS**	**40.52**	**58.61**	**84.02**
Sales				Silent			
	PSNR	Pts	SIR		PSNR	Pts	SIR
FS	39.18	4924.60	1.00	FS	36.53	4924.60	1.00
DS	39.14	67.12	73.37	DS	36.09	74.97	65.68
CDS	39.13	50.57	97.39	CDS	36.02	61.12	80.57
RBS	**39.15**	**38.76**	**127.04**	**RBS**	**36.15**	**39.73**	**123.94**
Stefan				Table			
	PSNR	Pts	SIR		PSNR	Pts	SIR
FS	26.61	4924.60	1.00	FS	29.32	4924.60	1.00
DS	24.71	104.51	47.12	DS	28.23	107.64	45.75
CDS	24.63	102.52	48.03	CDS	28.12	107.01	46.02
RBS	**24.98**	**86.00**	**57.26**	**RBS**	**28.38**	**81.53**	**60.40**

Table 3. Comparison of different algorithms on 8 sequences using 100 frames.

7. REFERENCES

[1] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," IEEE Trans. Circuits Syst. Video Technol., vol. 9, no. 1, pp. 70–84, Feb. 1999.

[2] Joint Video Team of ITU-T and ISO/IEC JTC 1, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, Mar. 2003.

[3] S. Zhu and K. K. Ma, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," IEEE Trans. on Image Processing, Vol. 9, No. 2, pp. 287 -290, Feb 2000

[4] 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 Syst. Video Technol., vol. 8, no. 4, pp. 369-377, Aug 1998.

[5] C. H. Cheung, and L. M. Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation," IEEE Trans, Circuits Syst. Video Technol., vol. 12, no. 12, Dec 2002.

[6] Yao Nie; Kai-Kuang Ma, "Adaptive rood pattern search for fast block-matching motion estimation", IEEE trans. Image Processing, vol. 11, pp.1442-1449, Dec 2002.

[7] A.M. Tourapis, H.Y. Cheong, O.C. Au, M.L. Liou, "N-dimensional Zonal Algorithms – The Future of Block-based Motion Estimation," Proc. Of IEEE Int. Conf. On Image Processing, 7-10 Oct 2001.