Multi-Direction Search Algorithm For Block-based Motion Estimation

Lai-Man Po, Ka-Ho Ng, Ka-Man Wong, and Kwok-Wai Cheung

Department of Electronic Engineering, City University of Hong Kong
Tat Chi Avenue, Kowloon, Hong Kong, China
Email: eelmpo@cityu.edu.hk

Abstract— **Easily trapped in local minima is one of the well-known problems in search point pattern based fast block motion estimation algorithms. This problem is especially serious in one-at-a-time search (OTS) and block-based gradient descent search (BBGDS). These two algorithms can provide very high speedup ratio but with low robustness in prediction accuracy especially for sequences with complex motions. Multi-path search (MPS) using more than one path have been proposed to improve the robustness of BBGDS, but the computational requirement is much increased. To tackle this problem, a novel multi-directional gradient descent search (MDGDS) is proposed in this paper with use of multiple OTSs in eight directions. Basically, the proposed MDGDS performs eight one-dimensional gradient descent searches on the error surface and therefore can trace to the global minimum more efficiently. Experimental results show that a significant improvement in computation reduction can be achieved as compared with well-known fast block motion estimation algorithms.**

I. INTRODUCTION

Block motion estimation algorithms are widely adopted by video coding standards, mainly due to their simplicity and good distortion performance. Using block motion estimation, a video frame is divided into non-overlapping block of equal size and the best-matched block is determined from reference frames to that block in the current frame within a predefined search window. Normally, this is performed by minimizing a block distortion measure (BDM) between these two blocks such as the sum of absolute difference (SAD). The most straightforward method is referred to as the full search (FS), which exhaustively evaluates all possible candidate blocks within the search window. It has been estimated that the computation of FS could consume up to 70% of the total computation of the video encoding process. To tackle this problem, many fast algorithms have been developed to speed up the motion estimation process. Among them, the most popular class is the block-matching algorithm (BMA) using a fixed set of search point patterns, which is mainly due to their very high speedup ratio in the motion estimation process. Well-known algorithms in this category are the one-at-a-time search (OTS) [1], the three-step search (3SS) [2], the new three-step search (N3SS) [3], the four-step search (4SS) [4], the block-based gradient descent search (BBGDS) [5], the diamond search (DS) [6], the hexagon-based search (HEXBS) [7], the cross-diamond search (CDS) [8] and cross-diamond-hexagonal search (CDHS) [9] etc. These algorithms, however, can be trapped by some local minima as they primarily rely on the unimodal error surface assumption, which means matching error monotonically decreased towards global minimum. In most real-world sequences, local minimum points can spread over the search window especially for the sequences with complex motion contents. Thus, the performance of these fast search algorithms depends very much on the motion content of a video sequence. The BBGDS is suitable for slow motion blocks estimation while the 3SS is relatively more suitable for the estimation of fast motion blocks. The 4SS, DS, and HEXB achieve better prediction accuracy for moderate motion blocks. Unfortunately, the real video sequences usually consist of wide-range motion contents and these fixed search point patterns based fast algorithms cannot provide satisfactory motion estimation results in these sequences. Search pattern switching algorithms [10-12] were proposed to solve this problem by adaptively using different search patterns among the 3SS, 4SS, DS, and BBGDS for achieving higher prediction accuracy. However, the performance of these algorithms highly depends on the accuracy of the motion content estimators and some of these estimators are also quite complex in practical implementation.

Recently, a multi-path search (MPS) algorithm [13] is also proposed with the use of more than one path to avoid using a wrong search path misled by the initial minimum search point. Basically, MPS is a multi-path BBGDS with multiple descending gradient paths. For each of the candidate paths, the compact square-shaped pattern of BBGDS is used. Experimental results show that MPS can provide much robust motion estimation accuracy but its computational requirement is quite significantly increased especially for sequences with complex motions. In this paper, we propose a novel multi-directional gradient descent search (MDGDS) algorithm that utilizes eight directional gradient searches to find at-most eight minimum points for determining a good next search center. This can minimize the chance of following a wrong search path and being trapped in a local minimum. The search point pattern of the proposed MDGDS is not fixed in each step. Instead, it depends on where the eight directional minimum search points are located, which is the major difference between MDGDS and conventional search point pattern based fast motion estimation algorithms.

The rest of this paper is organized as follows. In section II, we will first review the basic concept of 1-dimensional (1-D) and 2-dimensional (2-D) gradient descent searches using OTS and BBGDS, which are the bases of the proposed MDGDS. Moreover, MPS is also reviewed in section II, as it is a multi-path version of BBGDS. The details of the proposed MDGDS are described in section III with experimental results. Finally, the conclusion of this paper is addressed in section IV.

II. GRADIENT DECSENT SEARCH ALGORITHMS

The basic idea of the proposed MDGDS is to perform 1-D gradient descent search in eight directions. To provide a clearer picture of how this approach works in fast block motion estimation, the conventional 1-D and 2-D gradient descent search algorithms, namely OTS and BBGD, are first reviewed. Then we will review MPS, which is the multi-path version of BBGD.

A. One-at-a-Time Search (OTS)

The first OTS based block motion estimation algorithm was proposed in 1985 by Srinivasan et al. [1] which employs the OTS strategy in horizontal and then vertical direction. An example of OTS search path is shown in Fig. 1(a). First the search window center and

978-1-4244-2342-2/08/$25.00 ©2008 IEEE.

its two horizontally adjacent points, i.e. points at (0, 0), (-1, 0), and (1, 0), are searched. If point (-1, 0), i.e. the left adjacent point of the search window center, has the lowest distortion, OTS will be performed along the left direction of the search window center. If point (1, 0), i.e. the right adjacent point of the search window center, has the lowest distortion, OTS will be performed along the right direction. OTS stops when the minimum distortion point is closeted between two points with higher distortions. This point is regarded as the lowest distortion point in the horizontal direction of the search window center, which is noted as (s, 0). The OTS in vertical direction is performed similarly. First the points (s, 1) and (s, -1) are searched. If point (s, 1) has lower distortion, OTS will be performed along the upper direction of (s, 1). If point (s, -1) has lower distortion, OTS will be performed along the lower direction. When the minimum distortion point is closeted between two points with higher distortions, the motion vector (MV) pointing to that point is returned. In summary, OTS algorithm performs 1-D gradient descent search on the error surface twice. Although it uses fewer search points compared with other fast block motion estimation algorithms, its prediction quality is low. This is because 1-D gradient descent search is insufficient to provide a correct estimation of the global minimum position.

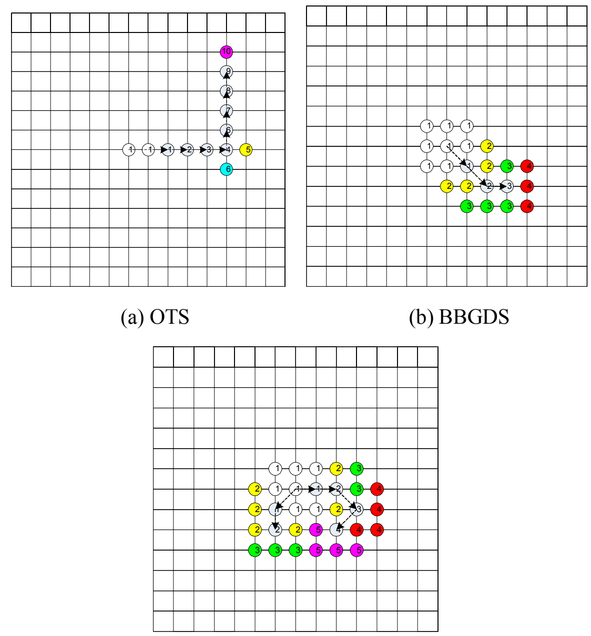
B. Block-based gradient descent search (BBGDS)

BBGDS [5] was proposed in 1996 by Liu et al. It performs 2-D gradient descent search instead of 1-D gradient search as in OTS algorithm. An example of BBGDS search path is shown in Fig. 1(b). First, the search window center point and its eight adjacent points are searched. If a lower distortion point is found among the eight points, that point will be the next search center. Three or five more points around this center will be searched. The procedure is repeated until the search center is enclosed between eight higher distortion points. The motion vector pointing to this point is returned. The eight adjacent points which BBGDS searches correspond to the eight directions, i.e. the upper, lower, left, right, upper-left, upper-right, lower-left, and lower-right directions of the search center. They cover all the possible directions from the search center. In other words, BBGDS performs a small-scale 2-D gradient descent search and then one-at-a-time moves towards the global minimum following a descending gradient path. BBGDS has a much better prediction quality in terms of PSNR than OTS algorithm.

C. Multi-Path Search (MPS)

BBGDS provides very high speedup ratio in motion estimation but it is easily trapped in the local minima causing low robustness in prediction accuracy. One of the reasons is that BBGDS only uses one single minimum distortion point found in a search step as the search center of the next step. Therefore, while the steepest descending gradient path is considered, other gradient descending paths will be ignored. Since the steepest descending gradient path may lead to a local minimum point instead of a global one, algorithms which consider all the candidate paths should have better prediction quality. Based on this idea, the MPS algorithm [13] was proposed in 2006 by Goel et al. Basically, MPS is a BBGDS with the use of multiple descending gradient paths. For each of the candidate paths, the compact square-shaped search point pattern of BBGDS is used. The algorithm converges when there is no new descending gradient path found.

Fig. 1(c) shows an example of the MPS algorithm. It can be seen that MPS is BBGDS with multiple descending gradient paths search. However, MPS is not efficient because it uses many points to search all candidate descending gradient paths. Table I lists the average PSNR per frame and the average number of search points used per macro-block for BBGDS and MPS. In this table, FS results are also listed as reference for performance comparison. The test sequences are of size CIF. The macro-blocks are of size 16x16 pixels and the



(a) OTS (b) BBGDS

(c) MPS

Fig. 1. Example for OTS , BBGDS and MPS.

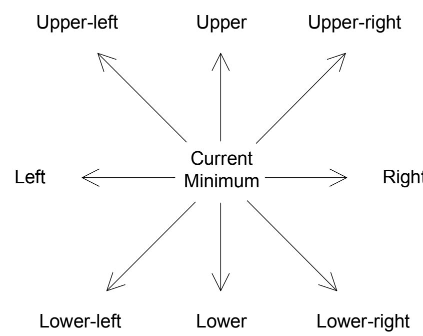


Fig. 2. Eight directional searches of MDGDS.

search window is of size ± 15 pixels. The SAD is used as the BDM and the frame structure is (IPPP…). From Table I, it can be found that MPS improves over BBGDS in terms of PSNR for most video sequences. For large or complex motion sequences the improvement is more obvious. For example, in the sequences of Football and Stefan, PSNRs increase by 0.236 dB and 0.269 dB, respectively. However, MPS is much slower than BBGDS. For Football, Foreman, and Stefan, MPS is 52.31%, 38.99%, and 28.85% slower than BBGDS, respectively. In summary, MPS can improve the robustness of BBGDS, but the computational requirement is significantly increased especially for complex motion sequences.

III. MDGDS

Basically, the strategy of OTS is a 1-D gradient descent search in a particular direction and the conventional OTS motion estimation algorithm performs OTS twice in the search window. A 2-D gradient descent search algorithm, e.g. BBGDS, performs better than a 1-D based search algorithm. MPS is a multiple paths search algorithm for improving the performance of BBGDS, but it is not very efficient in terms of computational complexity. In this section, a novel 2-D gradient descent search algorithm called Multi-Directional Gradient Descent Search (MDGDS) is proposed. It out-performs BBGDS by considering all descending gradient paths while achieving lower computational complexity than MPS by using OTS in eight directions. The proposed MDGDS independently searches eight directions, namely upper, lower, left, right, upper-left, upper-right, lower-left, and lower-right directions, of the point with the current minimum (CURRENT_MIN) distortion. These eight directions are shown in Fig. 2 and each of these eight directional searches uses OTS strategy. In OTS, the point-by-point search along a direction is continued if a newly searched point has lower distortion than the previously searched point. Otherwise the search in that direction stops. The minimum distortion found by each directional search is set as a directional minimum (DIRECTIONAL_MIN). After a search round is completed, the lowest distortion amongst the DIRECTIONAL_MINs is set as the CURRENT_MIN and the next search round starts at the point with the CURRENT_MIN. The proposed MDGDS algorithm can be summarized as:

Step 1: Calculate the BDM of the search window center and set its distortion value as CURRENT_MIN.

Step 2: For each of the eight directions of the point with the CURRENT_MIN, as illustrated in Fig. 2, point-by-point directional OTS is performed. The minimum BDM found by each of the eight directional searches is set as a directional minimum (DIRECTIONAL_MIN). If no point with DIRECTIONAL_MIN is found (The current search center is always the minimum point), go to Step 4; Otherwise go to Step 3.

Step 3: The DIRECTIONAL_MINs are compared. The lowest amongst them is set as the CURRENT_MIN. This is the end of a search round. Go back to Step 2 with the updated CURRENT_MIN and the search point with the CURRENT_MIN.

Step 4: The algorithm is completed. Return with the final motion vector pointing to the search point with the CURRENT_MIN.

An example of zero motion vector (ZMV) return from MDGDS is shown in Fig. 3. As the search window center is always the minimum point, there is no directional minimum point found in the first round of the step 2. The resulting search point pattern is equivalent to the 3x3 square search point pattern of BBGDS. Thus, the MDGDS can maintain low computational requirement for stationary blocks as BBGDS. An example of the first search round of the proposed MDGDS is shown in Fig. 4 with six directional minimum points in the upper, lower, left, right, upper-left and lower-right directions. The lowest directional minimum point is found in the lower-right direction (point number 8), thus the second round of MDGDS is performed with this point as center, which is shown in Fig. 5. In the second search round, there are also six directional minimum points and the lowest distortion among them is found in the right direction (point number 23). It can be seen that the search point patterns in each round is not fixed, which depends on where and how directional minimum points are found. The strategy of performing eight directional searches, however, is unchanged. The final round of the MDGDS occurred in the third search round is shown in Fig. 6, which shows that the terminal round is the same as performing a 3x3 square-shape search pattern at the current minimum point.

As MDGDS searches each candidate descending gradient path, it should perform better than BBGDS. In addition, it uses OTS to search these paths and so it requires fewer search points than MPS. The performance comparison of MDGDS with FS, BBGDS, MPS, 3SS, 4SS and DS is shown in Table I. From Table I, it can be found that MDGDS has quality improvement in terms of PSNR over BBGDS. For example, the PSNR of MDGDS is 0.338dB higher than that of BBGDS for Stefan sequence. For the sequences of Football and Foreman, MDGDS has 0.311dB and 0.164dB PSNR improvement over BBGDS, respectively. For other test sequences, it

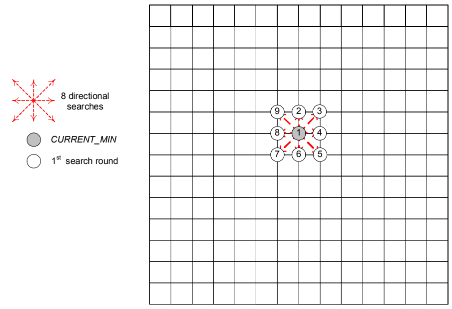


Fig. 3. An example of ZMV return from MDGDS.

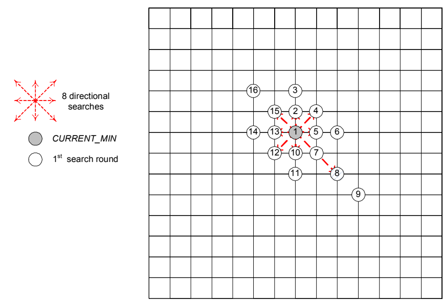


Fig. 4. An example of MDGDS in 1st search round.

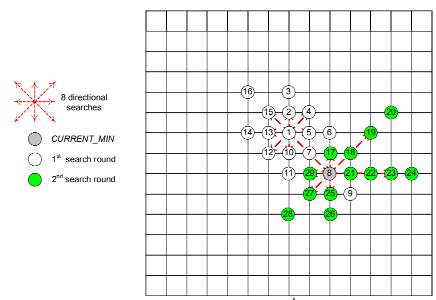


Fig. 5. An example of DGDS in 2nd search round.

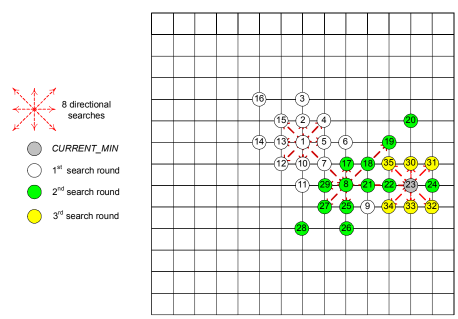


Fig. 6. An example of MDGDS converges in 3 search rounds.

also has slight PSNR improvements over BBGDS. The number of search points MDGDS uses is close to that of BBGDS. That means the computational complexity of MDGDS is similar to that of BBGDS.

MDGDS also performs better than MPS. For all test sequences except Sean, MDGDS uses fewer search points than MPS but achieves higher PSNR quality. The PSNR of MDGDS is only 0.002dB lower than that of MPS for sequence Sean. For other sequences MDGDS achieves slightly higher PSNR quality than MPS. However, MDGDS has huge speedup over MPS. For sequences

	Football		Stefan		Foreman		Silent	
	PSNR	*# of SPs*	*PSNR*	*# of SPs*	*PSNR*	*# of SPs*	*PSNR*	*# of SPs*
FS	25.844	868.333	25.677	868.333	32.842	868.333	36.236	868.333
BBGDS	24.709	20.760	23.575	15.516	32.140	16.725	35.560	9.976
MPS	24.945	31.619	23.844	19.992	32.270	23.246	35.631	11.150
MDGDS	**25.020**	**22.025**	**23.913**	**15.891**	**32.304**	**18.246**	**35.677**	**10.198**
3SS	24.905	30.676	24.042	30.763	31.920	30.776	35.845	30.619
4SS	25.055	27.478	23.619	25.942	31.998	26.289	35.882	23.755
DS	24.881	22.179	24.033	18.016	32.096	18.266	35.663	13.341

	Sean		Mobile		Coastguard		Akiyo	
	PSNR	*# of SPs*	*PSNR*	*# of SPs*	*PSNR*	*# of SPs*	*PSNR*	*# of SPs*
FS	39.461	868.333	24.336	868.333	30.035	868.333	43.044	868.333
BBGDS	39.447	8.924	24.276	10.673	29.809	13.718	43.040	8.521
MPS	39.450	9.281	24.278	12.037	29.858	15.875	43.040	8.598
MDGDS	**39.448**	**9.068**	**24.278**	**11.177**	**29.868**	**14.281**	**43.040**	**8.550**
3SS	39.314	30.627	23.948	30.636	29.397	30.784	42.919	30.616
4SS	39.332	23.444	24.028	23.698	29.756	26.015	42.939	23.240
DS	39.423	12.573	24.244	13.465	29.879	16.707	43.025	12.268

Table I: Performance Comparison of MDGDS with FS, BBGDS, MPS, 3SS, 4SS, and DS.

Football, Foreman, and Stefan, MDGDS is 30.34%, 21.51%, and 20.517% faster than MPS respectively. For small to medium motion sequences, the speed improvements of MDGDS over MPS are less significant. For example in Coastguard, Silent, and Mobile, MDGDS is 10.04%, 8.54%, and 7.14% faster than MPS respectively. Based on the above results, it is obvious that MDGDS is a more efficient multiple paths gradient descent search algorithm than MPS. On the other hand, Table I also listed the performance of some conventional well-known algorithms such as 3SS, 4SS and DS, MDGDS. It is also found that MDGDS can achieve much high speedup ratio with better or similar prediction accuracy in terms of PSNR.

IV. CONCLUSION

A new multi-direction search algorithm with use of eight 1-D gradient descent searches in each search stage is proposed in this paper. It outperforms traditional fast block motion estimation algorithms by considering all descending gradient paths while maintaining lower computational complexity by using OTS on eight directions. Compared with other well-known fast block motion estimation algorithms, MDGDS has higher prediction quality and faster speed. It is also very robust as it works well in videos with different motion contents.

ACKNOWLEDGMENT

The work described in this paper was substantially supported by a grant from CityU Applied R & D Funding (ARD) from City University of Hong Kong [Project No. 9667014].

REFERENCES

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

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun, Conf., NO, L.A., Nov-Dec 1981*, pp. G5.3.1-G.5.3.5.

[3] R. Li, B. Zeng, and M. L. Lio, "A new three-step search algorithm for block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 4, pp. 438-443, Aug. 1994.

[4] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 6, pp. 313-317, June 1996.

[5] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 6, pp. 419-422, Aug 1996.

[6] 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. on Circuits and Systems for Video Technology,* vol. 8, pp. 369-377, Aug. 1998.

[7] C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 12, pp. 349-355, May 2002.

[8] C. H. Cheung and L. M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 12, pp. 1168-177, Dec. 2002.

[9] C. H. Cheung and L. M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEE Transactions on Multimedia,* vol. 7, pp. 16-22, Feb 2005.

[10] J. H. Lim and H. W. Choi, "Adaptive motion estimation algorithm using spatial and temporal correlation," in *Proc. IEEE Pacific Rim Conference on Communications, Computers and signal Processing 2001*, pp. 473-476.

[11] I. Ahmad, W. Zheng, J. Luo, and M. Liou, "A fast adaptive motion estimation algorithm," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 16, pp. 420-438 Mar 2006.

[12] K. H. Ng, L. M. Po, and K. M. Wong, "Search Patterns Switching for Motion Estimation using Rate of Error Descent," in *Proceedings of International Conference on Multimedia & Expo*, Beijing, China, 2007, pp. 1583-1586.

[13] S. Goel and M. A. Bayoumi, "Multi-path search algorithm for block-based motion estimation," in *Proceedings of IEEE International Conference on Image Processing*, Atlanta, USA, 2006, pp. 2373-2376.