Novel Directional Gradient Descent Searches for Fast Block Motion Estimation

Lai-Man Po, Ka-Ho Ng, Kwok-Wai Cheung, Ka-Man Wong, Yusuf Md. Salah Uddin, and Chi-Wang Ting

***Abstract*—Search point pattern-based fast block motion estimation algorithms provide significant speedup for motion estimation but usually suffer from being easily trapped in local minima. This may lead to low robustness in prediction accuracy particularly for video sequences with complex motions. This problem is especially serious in one-at-a-time search (OTS) and block-based gradient descent search (BBGDS), which provide very high speedup ratio. A multipath search using more than one search path has been proposed to improve the robustness of BBGDS but the computational requirement is much increased. To tackle this drawback, a novel directional gradient descent search (DGDS) algorithm using multiple OTSs and gradient descent searches on the error surface in eight directions is proposed in this letter. The search point patterns in each stage depend on the minima found in these eight directions, and thus the global minimum can be traced more efficiently. In addition, a fast version of the DGDS (FDGDS) algorithm is also described to further improve the speed of DGDS. Experimental results show that DGDS reduces computation load significantly compared with the well-known fast block motion estimation algorithms. Moreover, FDGDS can achieve faster speedup compared with the UMHexagonS algorithm in H.264/AVC implementation while maintaining very similar rate-distortion performance.**

***Index Terms*—Block matching, motion estimation, video coding.**

I. Introduction

BLOCK MATCHING motion estimation (BMME) is widely adopted by video coding standards such as MPEG-2, MPEG-4, and H.264/AVC, mainly due to its simplicity and good distortion performance. Using BMME, a video frame is divided into non-overlapping blocks 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), e.g., the sum of absolute difference (SAD), between this pair of blocks. The most straightforward method is the full search (FS), which exhaustively evaluates all possible candidate blocks within the search window. However, the computational complexity of FS is very high. It has been estimated that FS could consume up to 70% of the total computation of the video encoding process. To tackle this problem, many fast block matching algorithms (BMAs) [1]–[9] have been proposed. These algorithms employ different search point patterns to search for the best matched block. To further speed up the motion estimation process, directional information is used to reduce the number of search points required in a search pattern [10]. However, these algorithms rely primarily on the unimodal error surface assumption, which assumes that matching error monotonically decreases toward the global minimum. In most real-world video sequences, local minimum points can spread over the search window, especially for sequences with complex motion contents. Thus these fast algorithms can be trapped by local minima and cannot provide satisfactory motion estimation results.

Search patterns switching algorithms [11], [12] were proposed to solve the above problem by adaptively using different search patterns to achieve higher prediction accuracy. However, the performance of these algorithms depends highly on the accuracy of the motion content estimators, and some of these estimators are quite complex in practical implementation. Besides the initial search point pattern, it is also found that the initial minimum distortion search point for the next step is very important to lead to a good search path. A hybrid unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) [13] that takes advantage of many search point patterns has been adopted in H.264/AVC reference software. This algorithm has good rate-distortion performance, but its initial search step is still very intensive computationally.

On the other hand, a multipath search (MPS) algorithm [14] has been proposed. It uses more than one search path to avoid following a wrong path misled by the initial minimum distortion point. Basically, MPS is a multipath version of block-based gradient descent search (BBGDS) [5], which searches 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 robust motion estimation accuracy but its computational complexity is high.

Manuscript received March 14, 2008; revised September 5, 2008. First version published April 7, 2009; current version published August 14, 2009. The work described in this paper was substantially supported by a grant from CityU Applied Research and Development Funding (ARD) from City University of Hong Kong (Project No. 9667014). This paper was recommended by Associate Editor M. Comel.

L.-M. Po, K.-H. Ng, K.-M. Wong, Y. Md. Salah Uddin, and C.-W. Ting are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong SAR, China (e-mail: eelmpo@cityu.edu.hk; khng@ee.cityu.edu.hk; kmwong@ee.cityu.edu.hk; ymdsalahu2@student.cityu.edu.hk; cwting@cityu.edu.hk).

K.-W. Cheung is with the Faulty of Science and Engineering, Chu Hai College of Higher Education, Riviera Gardens, Tsuen Wan, NT, Hong Kong SAR, China (e-mail: kwcheung@iee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2020320

In this letter, we propose a novel directional gradient descent search (DGDS) algorithm which utilizes eight directional gradient searches to minimize the chance of following a wrong search path and being trapped in a local minimum. A fast version of DGDS (FDGDS) is also proposed. The rest of this letter is organized as follows. Section II reviews the basic concepts of 1-D and 2-D gradient descent searches. The details of DGDS and FDGDS are described in Section III. Experimental results and conclusions are presented in Section IV and Section V, respectively.

II. Conventional Gradient Decsent Search Algorithms

To explain the principles of the proposed DGDS, the conventional 1-D and 2-D gradient descent search algorithms (OTS and BBGD) and MPS (multipath version of BBGD) are reviewed in this section.

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

The strategy of OTS is to keep searching along a particular search direction until the minimum point along that direction is found. The first OTS-based BMA [1] employs the OTS strategy in horizontal and then vertical direction. An example of the OTS search path is shown in Fig. 1. If, for example, the current minimum BDM point is at position (0, 1) and the upper-direction OTS is performed, then the point immediately above it, i.e., point (0, 2), will be searched. If point (0, 2) has lower distortion than (0, 1), point (0, 2) will be set as the current minimum distortion point. Point (0, 3), which is above point (0, 2), will then be searched. The search continues until the minimum point is closeted between two higher values, or until the search window boundary is reached. As OTS follows the descending gradient path in a particular direction, it can be considered as a 1-D gradient descent search in that direction. This is an efficient searching strategy because it does not waste effort in probing into unknown terrain of the error surface. Moreover, it is also easy to be implemented in hardware, and data access is efficient because a search point is always adjacent to the previous search point. In summary, the OTS performs 1-D gradient descent search on the error surface twice. Although it uses fewer search points compared with other fast BMAs, its prediction quality is low. This is because a 1-D gradient descent search is insufficient to estimate the global minimum position.

B. Block-Based Gradient Descent Search

BBGDS performs 2-D gradient descent search. An example of BBGDS search path is shown in Fig. 2. The eight adjacent points which BBGDS searches correspond to the eight directions. 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 toward the global minimum following a descending gradient path. BBGDS has a much better prediction quality in terms of PSNR than OTS algorithm.

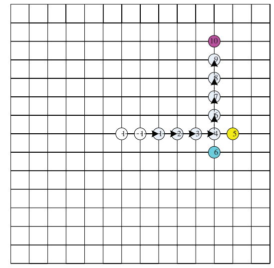

Fig. 1. Example of OTS algorithm.

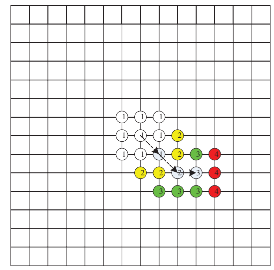

Fig. 2. Example of BBGDS.

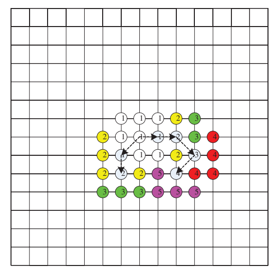

Fig. 3. Example of MPS algorithm.

C. Multipath Search

BBGDS provides very high speed-up ratio in motion estimation but it is easily trapped in the local minima causing low robustness in prediction accuracy. One reason 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 that consider all the candidate paths should have better prediction quality. Based on this idea, the MPS algorithm was proposed. Basically, MPS is a BBGDS using 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. 3 shows an example of MPS. However, MPS is not efficient because it uses many points to search all candidate descending gradient paths. Experiments show that MPS can improve the robustness of BBGDS but with significantly increased computational requirement, especially for complex motion sequences.

III. Directional Gradient Descent Search

The strategy of OTS is a 1-D gradient descent search in a particular direction, and the conventional OTS motion

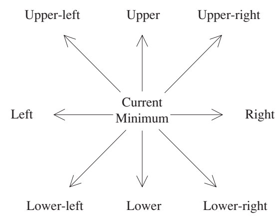

Fig. 4. Eight directional searches of DGDS.

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 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 directional gradient descent search (DGDS) is proposed. It outperforms BBGDS by considering all descending gradient paths while achieving lower computational complexity than MPS by using OTS in eight directions.

A. DGDS Algorithm

DGDS 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. 4 and each of the eight directional searches uses the 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 among the *DIRECTIONAL_MINs* is set as *CURRENT_MIN* and the next search round starts at the point with *CURRENT_MIN*. The proposed DGDS algorithm can be summarized as follows.

DGDS Algorithm

- Step 1: Calculate the BDM of the search window center and set the value as *CURRENT_MIN*.
- Step 2: For each of the eight directions of the point with *CURRENT_MIN* (Fig. 4)
  - (a) perform point-by-point directional OTS;
  - (b) set the minimum BDM found in the current direction as a *DIRECTIONAL_MIN*.
- Step 3: If no point with *DIRECTIONAL_MIN* is found (i.e., the current search center is still the minimum point), go to Step 5; otherwise go to Step 4.
- Step 4: *DIRECTIONAL_MINs* are compared. The lowest one is set as *CURRENT_MIN*. This is the end of a search round. Go to Step 2 with updated *CURRENT_MIN* and its corresponding position.
- Step 5: The algorithm is completed. Return with the final motion vector (MV) pointing to the position with *CURRENT_MIN*.

An example of zero motion vector (ZMV) return from DGDS is shown in Fig. 5. As the search window center is the

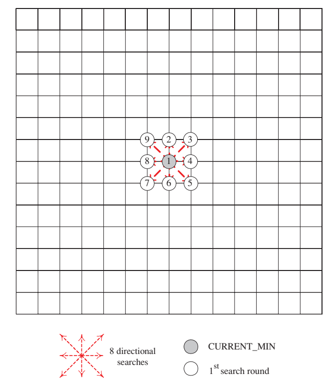

Fig. 5. Example of ZMV return from DGDS.

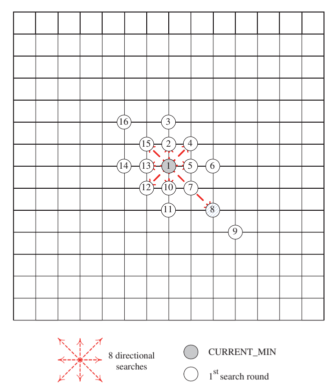

Fig. 6. Example of DGDS in the first search round.

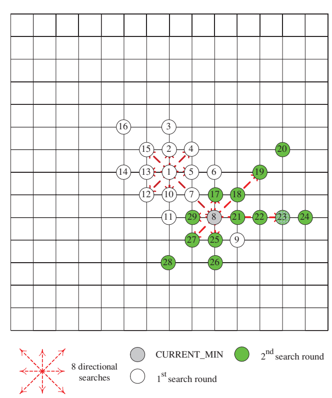

Fig. 7. Example of DGDS in the second search round.

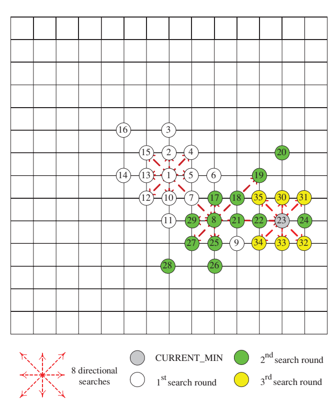

Fig. 8. Example of DGDS converging in the third search round.

TABLE I
PERFORMANCE COMPARISON OF FDGDS WITH OTHER ALGORITHMS

	Akiyo		*News*		*Coastguard*		*Foreman*		*Stefan*	
	PSNR	no. of search points	PSNR	no. of search points	PSNR	no. of search points	PSNR	no. of search points	PSNR	no. of search points
FS	43.044	868.333	36.966	868.333	30.035	868.333	32.842	868.333	25.677	868.333
BBGDS	43.040	8.521	36.660	9.247	29.809	13.718	32.140	16.725	23.575	15.516
MPS	43.040	8.598	36.706	9.882	29.858	15.875	32.270	23.246	23.844	19.992
3SS	42.919	30.616	36.607	30.618	29.397	30.784	31.920	30.776	24.042	30.763
N3SS	43.014	15.912	36.584	16.422	29.372	19.344	32.033	21.973	23.423	23.198
4SS	42.939	23.240	36.699	23.539	29.756	26.015	31.998	26.289	24.619	25.942
HEXBS	42.817	10.342	36.546	10.665	29.785	13.043	31.420	13.709	23.932	13.873
DS	43.025	12.268	36.694	12.800	29.879	16.707	32.096	18.266	24.033	18.016
CDS	43.011	8.700	36.677	9.523	29.879	15.682	32.014	16.686	23.952	16.958
DGDS	43.040	8.550	36.722	9.422	29.868	14.281	32.304	18.246	23.913	15.891
FDGDS	**43.040**	**8.512**	**36.710**	**9.099**	**29.865**	**12.394**	**32.264**	**15.299**	**23.911**	**13.195**
PSNR degradation of FDGDS over DGDS	0.000		−0.012		−0.003		−0.040		−0.002	
Speed improvement of FDGDS over DGDS	*+0.44%*		*+3.43%*		*+13.22%*		*+16.15%*		*+16.96%*	

minimum point, there is no directional minimum point found in the first round of Step 2. The resulting search point pattern is equivalent to the 3 × 3 square search point pattern of BBGDS. Thus, the DGDS can maintain a low computational requirement for stationary blocks like BBGDS.

An example of the first search round of the proposed DGDS is shown in Fig. 6 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 DGDS is performed with this point as center, which is shown in Fig. 7. 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 pattern in each round is not fixed. It depends on where and how directional minimum points are found. The strategy of performing eight directional searches, however, is unchanged. An example of DGDS with the final step occurring in the third search round is shown in Fig. 8, which shows that the terminal round is the same as performing a 3 × 3 square-shaped search pattern at the current minimum point.

Since DGDS searches each candidate's descending gradient path, it performs better than BBGDS. In addition, it uses OTS to search these paths and therefore it requires fewer search points than MPS. From Table I, it can be seen that DGDS has quality improvement in terms of PSNR over BBGDS. For example, the PSNR of DGDS is 0.338 dB higher than that of BBGDS for *Stefan* sequence. For sequence *Foreman*, DGDS has 0.164 dB PSNR improvement over BBGDS. For other test sequences, it also has slight PSNR improvements over BBGDS. The number of search points DGDS uses is close to that of BBGDS. That means the computational complexity of DGDS is similar to that of BBGDS. DGDS also performs better than MPS. For all test sequences, DGDS uses fewer search points than MPS but achieves higher PSNR quality. For sequences *Coastguard*, *Foreman*, and *Stefan*, DGDS is 10.04, 21.51, and 20.517% faster than MPS, respectively. For small motion sequences, the speed improvements of DGDS over MPS are less significant. Based on the results, it is obvious that DGDS is a more efficient multiple path gradient descent search algorithm than MPS.

B. Fast DGDS Using Relative Distortion Ratio

The speed of DGDS can be further increased. If a directional minimum distortion point with a much lower distortion relative to the current minimum distortion is found in one of the eight directional searches, the remaining directional searches can be skipped to speed up the algorithm. A new round of search can start immediately at that directional minimum distortion point. The relative distortion ratio (RDR) between a directional minimum distortion *DIRECTIONAL_MIN* and a current minimum distortion *CURRENT_MIN* is defined as

$$RDR = \frac{DIRECTIONAL\_MIN}{CURRENT\_MIN}. \tag{1}$$

If a directional search has RDR lower than a threshold T, other directional searches could be skipped. The *CURRENT_MIN* point will be set to this *DIRECTIONAL_MIN* point and a new round of search will start from this center.

TABLE II
PERFORMANCE OF FDGDS WITH DIFFERENT THRESHOLDS (T)

	Akiyo		*News*		*Coastguard*		*Foreman*		*Stefan*	
T	PSNR	no. of search points	PSNR	no. of search points	PSNR	no. of search points	PSNR	no. of search points	PSNR	no. of search points
0.1	43.040	8.550	36.722	9.424	29.868	14.236	32.307	17.975	23.913	15.681
0.2	43.040	8.548	36.722	9.403	29.868	13.922	32.306	17.450	23.913	14.993
0.3	43.040	8.541	36.721	9.318	29.868	13.439	32.300	16.813	23.912	14.200
0.4	43.040	8.528	36.716	9.195	29.867	12.938	32.285	16.046	23.912	13.613
0.5	**43.040**	**8.512**	**36.710**	**9.099**	**29.865**	**12.394**	**32.264**	**15.299**	**23.911**	**13.196**
0.6	43.040	8.498	36.704	9.025	29.853	11.735	32.234	14.613	23.904	12.894
0.7	43.039	8.485	36.697	8.965	29.833	11.228	32.191	14.024	23.886	12.598
0.8	43.038	8.475	36.682	8.905	29.807	10.960	32.136	13.471	23.844	12.250
0.9	43.038	8.464	36.661	8.836	29.762	10.823	32.075	12.898	23.735	11.808

This DGDS with fast convergence is denoted as Fast DGDS (FDGDS). The procedures of FDGDS are as follows.

FDGDS Algorithm

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

Step 2: For each of the eight directions of the point with *CURRENT_MIN*:
- (a) perform point-by-point directional OTS;
- (b) set the minimum BDM found in the current direction as a *DIRECTIONAL_MIN*;
- (c) if *DIRECTIONAL_MIN* is found, calculate the relative distortion ratio (RDR) for the current direction using (1). If RDR $< T$, update *CURRENT_MIN* with this *DIRECTIONAL_MIN* and repeat Step 2 (i.e., skip the remaining directional searches).

Step 3: If no point with *DIRECTIONAL_MIN* is found, go to Step 5; Otherwise go to Step 4.

Step 4: *DIRECTIONAL_MINs* are compared. The lowest one is set as *CURRENT_MIN*. This is the end of a search round. Go to Step 2 with updated *CURRENT_MIN* and its corresponding position.

Step 5: The algorithm is completed. Return the final MV pointing to the position with the *CURRENT_MIN*.

The threshold T used in FDGDS controls the speed of convergence of the algorithm. It is in the range of 0 to 1. For example, if it is set at 0.5, that means that whenever the RDR is lower than 0.5, other directional searches will be skipped and a new round of search will be started. This implies that the value of the directional minimum distortion is less than 50% of the value of the current minimum distortion. Although setting a higher threshold T will speed up the convergence of the algorithm, it will also degrade the prediction quality because it will lower the chance of finding the global minimum point. The impacts of using different thresholds can be found by experiment and is studied in the following section.

IV. EXPERIMENTAL RESULTS

Simulations are conducted with the luminance components of a number of popular test video sequences to evaluate the proposed algorithms. SAD is used as the block distortion measure and the block size is 16×16 pixels. The frame structure is (IPPP…) and the search range is ± 15 pixels. Simulation results are expressed as the average number of search points used per block and average PSNR per frame.

A. Evaluation of the Speedup by FDGDS Over DGDS

Experiment is conducted to select a threshold for FDGDS in order to get performance similar to that of DGDS. Table II tabulates the average PSNR per frame and the average number of search points used per block of FDGDS using different thresholds for fast convergence. The results show that the number of search points together with the PSNR quality decreases with increasing threshold T value. The higher the threshold T, the faster the convergence and therefore the lower the prediction quality as the algorithm uses fewer number of search points. From Table II, it can be seen that $T = 0.5$ is a good balance between speed and quality. Therefore $T = 0.5$ is chosen as the threshold value in FDGDS. The last two rows of Table I show the comparison between DGDS and FDGDS. It can be found that FDGDS has a maximum of 0.04 dB PSNR degradation over DGDS in *Foreman*. For other sequences the quality degradation is very slight or there is no degradation at all. However, FDGDS has significant speedup over DGDS. For sequences *Coastguard*, *Foreman*, and *Stefan*, FDGDS is 13.22, 16.15, and 16.96% faster than DGDS, respectively. Therefore FDGDS performs even better than DGDS.

B. Performance Comparison of FDGDS With Conventional Algorithms

Table I also compares the performance of FDGDS with three-step search (3SS) [2], new three-step search (N3SS) [3], four-step search (4SS) [4], block-based gradient descent search

TABLE III
PERFORMANCE COMPARISON IN H.264 REFERENCE SOFTWARE OF FDGDS WITH 3SS, BBGDS, AND UMHEXAGONS (QP = 28)

	Akiyo			*News*			*Coastguard*		
	PSNR (dB)	Bitrate (kbits/s)	Run time (s)	PSNR (dB)	Bitrate (kbits/s)	Run time(s)	PSNR (dB)	Bitrate (kbits/s)	Run time (s)
FS	40.00	404.33	371.86	38.35	698.41	373.35	34.45	1896.58	373.88
3SS	39.97	404.84	21.88	38.34	702.16	21.80	34.44	1899.68	22.12
BBGDS	39.97	404.45	11.65	38.34	699.30	12.02	35.45	1893.92	12.27
UMHexagonS	39.99	404.38	10.27	38.35	697.99	11.16	34.44	1892.34	17.43
FDGDS	**39.97**	**404.47**	**8.79**	**38.34**	**698.9**	**8.98**	**34.44**	**1892.17**	**10.25**
FDGDS speedup over UMHexagonS	**14.40%**			**19.49%**			**41.17%**		

	Foreman			*Stefan*			*Container*		
	PSNR (dB)	Bitrate (kbits/s)	Run time (s)	PSNR (dB)	Bitrate (kbits/s)	Run time (s)	PSNR (dB)	Bitrate (kbits/s)	Run time (s)
FS	36.69	877.49	370.63	35.22	2550.76	490.68	36.50	849.14	372.36
3SS	36.64	911.64	21.92	35.21	2572.02	22.31	36.49	849.10	21.81
BBGDS	36.65	892.93	12.03	35.20	2574.13	12.24	36.49	849.02	11.64
UMHexagonS	36.68	878.75	14.07	35.21	2553.82	15.92	36.49	851.35	11.54
FDGDS	**36.64**	**892.80**	**9.96**	**35.21**	**2663.62**	**10.16**	**36.49**	**849.03**	**9.12**
FDGDS speedup over UMHexagonS	**29.20%**			**36.19%**			**20.95%**		

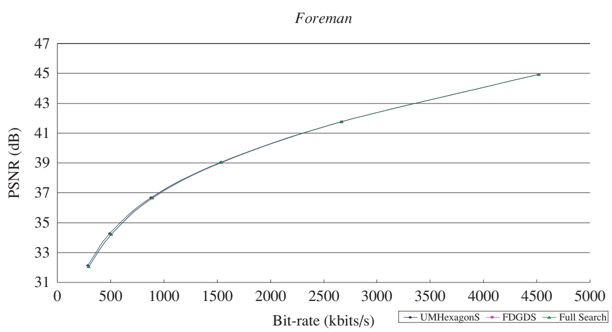

Fig. 9. Rate distortion comparison of FS, UMHexagonS, and FDGDS for sequence *Foreman*.

(BBGDS), multipath search (MPS), hexagon-based search (HEXBS) [7], diamond search (DS) [6], and cross-diamond search (CDS) [8]. The performance of FS is also included as reference. For sequences *Akiyo*, *News*, *Coastguard*, and *Stefan*, FDGDS is the fastest algorithm among the algorithms. For sequence *Foreman*, FDGDS is the second fastest algorithm. For sequences with complex motion contents, FDGDS has a substantial speedup over the other algorithms. For example for sequence *Stefan*, FDGDS has a speedup of 49.14 and 22.19% over 4SS and CDS, respectively. Comparing the matching quality, FDGDS achieves the highest PSNR for sequences *Akiyo* and *News*. It has the second highest PSNR for sequences *Coastguard* and *Foreman*. From the results, it can be seen that FDGDS has high prediction quality but with much lower computational complexity compared with other algorithms.

C. Experimental Results in H.264 Reference Software

In H.264 reference software "JVT H.264/AVC Reference Software Joint Model" (JM), some fast BMAs, e.g., the enhanced predictive zonal search (EPZS) [15], are implemented. In our experiments, JM9.6 was used, and UMHexagonS [13] is adopted as default fast inter-pixel motion estimation algorithm. The UMHexagonS combines many techniques from different motion estimation research fields. For example, it utilizes techniques in motion vector prediction and early termination researches. The proposed FDGDS algorithm is implemented in JM9.6 to compare with UMHexagonS. The configurations of the experiments are IPPPIPPP…frame structure, CAVLC, Hadamard transform, and one reference frame. RD optimization is turned off. Video sequences *Akiyo*, *News*, *Coastguard*, *Foreman*, *Stefan*, and *Container* are tested and 100 frames are used for each sequence. The sub-pixel motion search is disabled so that the experimental results can clearly reflect the performances of the algorithms. The reference software is run on Linux kernel 2.6 on an Intel Pentium 4 machine. Table III compares FDGDS with 3SS, BBGDS, and UMHexagonS, using the Quantization Parameter value 28. The average PSNR (dB) and bit rate (kbits/s) are used for video quality evaluation. The computational complexity is measured in total motion estimation time of JM9.6. It can be seen that FDGDS is much faster than UMHexagonS. FDGDS also achieves nearly the same PSNR and bit rate performance of UMHexagonS. Fig. 9. shows the rate distortion comparison of FS, UMHexagonS, and FDGDS for sequence *Foreman*. Quantization parameter (QP) values 16, 20, 24, 28, 32, and 36 are used to plot those curves. From the graph, we can see that the rate distortion performance of FDGDS is very close to that of FS and UMHexagonS.

V. Conclusion

Novel directional gradient descent search algorithms have been proposed in this letter. The proposed DGDS is a 2-D gradient descent search algorithm. It outperforms other fast BMAs by considering all descending gradient paths while maintaining lower computational complexity by using OTS on eight directions. In addition, a fast DGDS (FDGDS) with even better speedup is also proposed. Compared with other fast BMAs, FDGDS provides higher prediction quality and higher speed. It is also very robust as it works well in videos with different motion contents.

References

[1] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. 33, no. 8, 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.*, New Orleans, LA, Dec. 1981, pp. G5.3.1-G5.3.5.

[3] R. Li, B. Zeng, and M. L. Lio, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 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. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, Jun. 1996.

[5] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, 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. Circuits Syst. Video Technol.*, vol. 8, no. 4, 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. Circuits Syst. Video Technol.*, vol. 12, no. 5, 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. Circuits Syst. Video Technol.*, vol. 12, no. 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," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 16–22, Feb. 2005.

[10] B. Kim, S. Song, and P. Mah, "Enhanced block motion estimation based on distortion-directional search patterns," *Pattern Recognition Lett.*, vol. 27, no. 12, pp. 1325–1335, Sep. 2006.

[11] I. Ahmad, W. Zheng, J. Luo, and M. Liou, "A fast adaptive motion estimation algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, 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 *Proc. Int. Conf. Multimedia Expo*, Beijing, China, 2007, pp. 1583–1586.

[13] Z. B. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," in *Proc. 6th Meeting: JVT–F017*, Awaji, Island, Japan, 2002.

[14] S. Goel and M. A. Bayoumi, "Multi-path search algorithm for block-based motion estimation," in *Proc. IEEE Int. Conf. Image Process.*, Atlanta, GA, 2006, pp. 2373–2376.

[15] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Proc. SPIE Visual Commun. Image Process.*, San Jose, CA, 2002, pp. 1069–1079.