Fast Multi-Reference Motion Estimation via Enhanced Downhill Simplex Search

Chen-Kuo Chiang, Hwai-Chung Fei and Shang-Hong Lai

Department of Computer Science, National Tsing Hua University,
Hsinchu 300, Taiwan, R.O.C.

{ckchiang, lai}@cs.nthu.edu.tw

Abstract. Block motion estimation can be regarded as a function minimization problem in a finite-dimensional space. Therefore, fast block motion estimation can be achieved by using an efficient function minimization algorithm instead of a predefined search pattern, such as diamond search. Downhill simplex search is an efficient derivative-free function minimization algorithm. In this paper, we proposed a fast block motion estimation algorithm based on applying the downhill simplex search for function minimization. Several enhanced schemes are proposed to improve the efficiency and accuracy, including a new initialization process, a special rounding scheme, and an early-stop error function evaluation procedure. We also extend the downhill simplex search for the multi-reference frame motion estimation problem. Experimental results show superior performance of the proposed algorithm over some existing fast block matching methods on several benchmarking video sequences

Keywords: motion-estimation, block-matching algorithm, downhill simplex search, multi-reference-frame motion estimation

1 Introduction

Due to the strong demand of storing and transmitting an enormous amount of video data, video compression has been a very important and practical problem in recent years. Motion estimation (ME) is an indispensable part in video compression and has been popularly utilized to reduce the temporal information redundancy. Block matching algorithms (BMA) are required for ME in many video standards, such as MPEG-1 [1], MPEG-2 [2], MPEG-4 [3], H.263 [4], and H.264 [5]. In BMA, frames are divided into non-overlapping macroblocks, and it needs to find a motion vector (MV) in a pre-defined search range for each macroblock. The simplest BMA is the full search (FS) algorithm. This algorithm exhaustively searches over all possible locations in the search range and picks the most suitable block as the MV, so that it finds the optimal solution within the search range. However, FS has a fatal drawback, i.e. the high computational cost. Therefore, it is not practical to use FS in video compression, especially in real-time applications.

To reduce the computational complexity of FS, many fast BMAs, such as three step search [6], new three step search [7], four step search (FSS) [8], and diamond search (DS) [9] are proposed. Fast BMAs strategically check possible candidates in the search range to decrease the number of search points. Most video encoders apply fast BMAs for motion estimation since they can significantly reduce the search time without noticeable video quality degradation. The most important criterion for a fast BMA is to find an accurate MV with as few search points as possible.

In multi-reference frame motion estimation, the motion vector of one block can be predicted from many reference frames. Recently, a number of algorithms have been proposed to reduce the computational complexity. The center-biased frame selection [10] and recent-biased search [11] perform ME with predefined search patterns in 3D space. Su and Sun [12] used composed MVs to predict approximated results in multi-reference frames. In addition, a simplex minimization method [13] [14] is applied in each previous frame to form an initial simplex for searching the minimal solution of the block distortion function.

In this paper, a fast BMA is proposed to use a derivative-free function minimization algorithm, i.e. the downhill simplex search algorithm, to find MVs between adjacent image frames. Furthermore, we extend the downhill simplex search to multiple reference frames and present several improved schemes to boost the efficiency and accuracy of the motion estimation algorithm.

The rest of this paper is organized as follows. In the next section, we describe the proposed fast motion estimation algorithm based on downhill simplex search. The proposed simplex initialization and stopping criterion are also explained in details. Subsequently, we present several improved schemes, including a special rounding technique and an early-stop scheme, in section 3. Experimental results on real videos and comparison of the proposed algorithm with some previous BMAs are given in section 4. Finally, we conclude this paper in section 5.

2 Downhill Simplex Search Motion Estimation Algorithm

Downhill simplex search [15] is a derivative-free multidimensional function minimization method. In the downhill simplex search, a collection of n + 1 points in n-dimensional space is called a simplex and each point in the simplex has a specific function value. During the iterative simplex updating process, the point with the highest function value is iteratively replaced by a new point with a smaller function value until the stopping criterion is satisfied. Therefore, it is an iterative minimization process to search for the optimal function value.

2.1. Enhanced Downhill Simplex Search Algorithm for Motion Estimation

In the motion estimation problem, the goal is to find motion vectors (MVs) with the smallest block distortion measurement (BDM). It is just like the minimization process to search the best MVs in a multi-dimension space. In this point of view, downhill simplex search fits well to the motion estimation problem and the algorithm can be easily implemented.

For the 2-D search space in single-reference-frame ME, three points are required to form a simplex. For the case multiple reference frames, four points are required to form the simplex when it finds MVs in 3-D space, as shown in Fig. 1. Besides, sum of squared errors (SSE) is applied as the function to be minimized in this case. The downhill simplex search can be roughly divided into two parts: the initial simplex selection and the iterative update process. In the first part, the initial simplex can be determined by some heuristic method. It is important to select an appropriate initial simplex since we have better chance to find the correct solution very quickly when the actual solution is near or inside the initial simplex. After the initial simplex is determined, the second part is to update the simplex iteratively until the stopping criterion is satisfied. Finally, the point with the lowest function value in the simplex is the final solution.

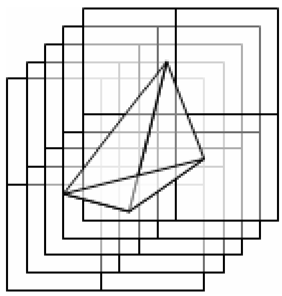

Fig. 1. The downhill simplex search for multi-reference-frame ME uses four points to form a simplex.

Fig. 2 shows the four main steps in downhill simplex search and the geometrical interpretation of these operations. They are reflection, expansion, contraction, and shrinkage. In the reflection step, a reflection point P_r is defined as

$$P_r = P_{ave} + \alpha(P_{ave} - P_h), \quad \alpha > 0 \qquad (1)$$

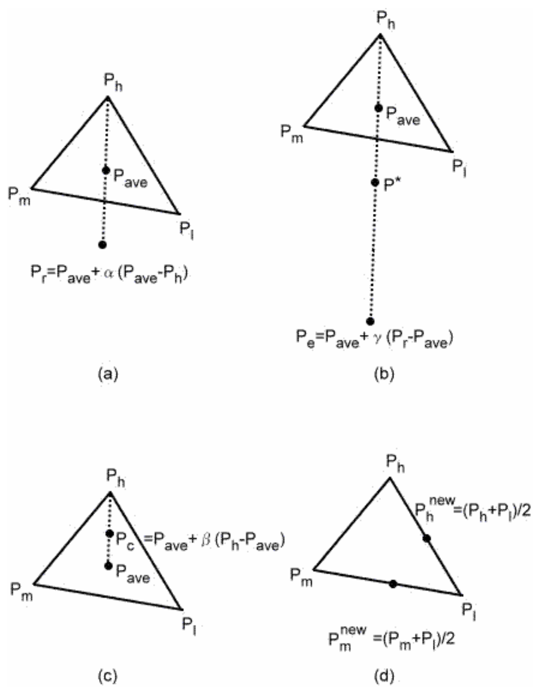

Fig. 2. The four steps in the downhill simplex iteration: (a) reflection, (B) expansion, (c) contraction, and (d) shrinkage.

and

$$P_{ave} = \frac{1}{n+1}\sum_{i=1}^{n+1} P_i \qquad (2)$$

where α is a positive constant and P_{ave} is the average of all points of the simplex. P_h is the point of the simplex with largest function value. After P_r is determined, the SSE of P_r is calculated as its function value Y_r. In the expansion step, an expansion point P_e is defined as

$$P_e = P_{ave} + \gamma(P_r - P_{ave}), \quad \gamma \ge 1 \qquad (3)$$

where γ is a constant greater than or equal to one. Y_e denotes the SSE function value of P_e. In the contraction step, a contraction point P_c is defined as

$$P_c = P_{ave} + \beta(P_h - P_{ave}), \quad 0 < \beta < 1 \qquad (4)$$

where β is a constant between zero and one. Y_c is the function value of P_c.

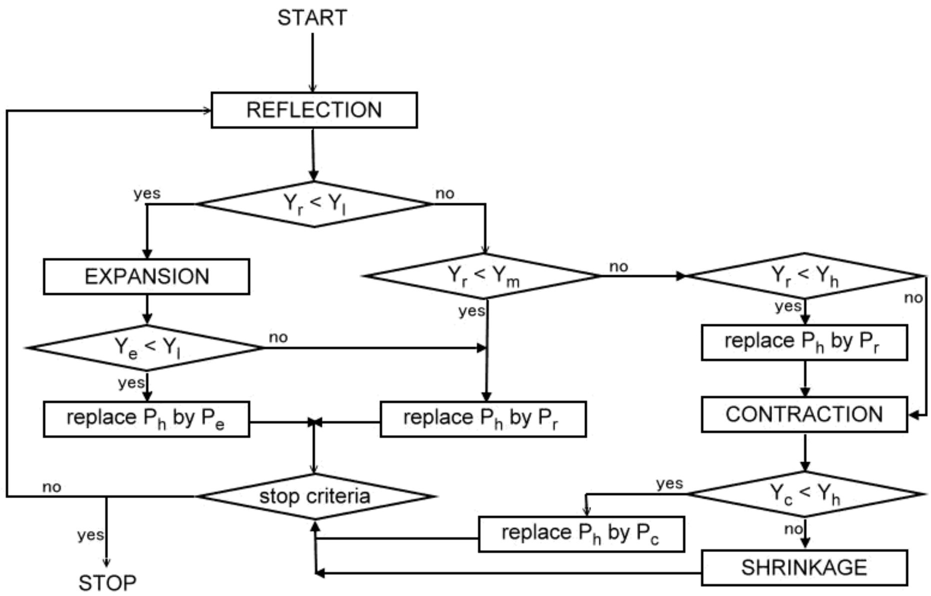

Fig. 3. Flow chart of the downhill simplex search.

In the shrinkage step, a contraction point P_i^{new} is defined as

$$p_i^{new} = (p_i + p_l)/2\text{, for } i=1\ldots n+1, i \neq l \quad (5)$$

After these points are determined, the SSE values of these points are calculated to see which point can be used to replace the point with the largest value in the simplex. In 2D-space, Y_h is defined as the SSE value for the point with the largest SSE value. Y_l is defined as the SSE value for the point with the smallest SSE value. Y_m is defined similarly.

Fig. 3 shows the flow chart of the iterative simplex update procedure. Each iteration starts with the reflection step. In this step, it reflects the point of P_h according to α. Conceptually, it finds the point in the opposite direction of P_h and tries to find a smaller SSE value. If a smaller Y_r is obtained, the direction it is trying is possibly right. Thus, it goes to the expansion step. In the expansion step, it goes farther along the direction according to γ. If a large Y_r is obtained, even larger than Y_h after the reflection step, it goes to the contraction step. It means it is hardly to find a smaller SSE value along this direction. Therefore, it goes backward. In the shrinkage step, all points except the point with the lowest function value are moved toward the lowest point to make the triangular bounding area shrunk. After each step, the point with the highest function value is replaced by the new point with a smaller function value, and then the stopping criterion is checked to see if it is satisfied to terminate the iterations.

Another essential element in the downhill simplex process is when to stop the iteration. A stopping criterion must be carefully designed. The stopping criterion here is when any two of the three points in the simplex are the same point.

2.2. Initial Simplex Selection

In this work, an enhanced downhill simplex search is proposed as the BMA to determine the motion vectors for video coding. For the downhill simplex search method, one of the key

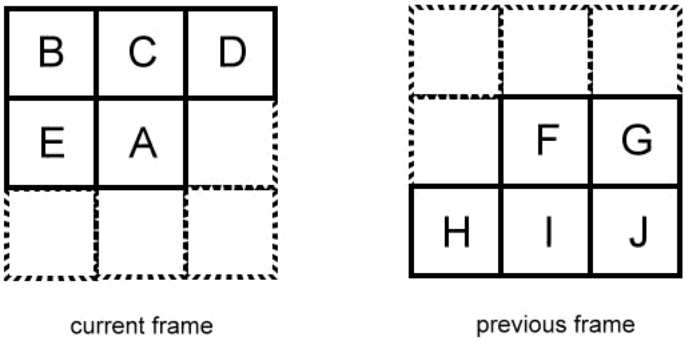

Fig. 4. The blocks used in the motion vector prediction for the initial simplex selection.

factors that determine the search performance is the selection of a good initial simplex. If the correct MVs are near the initial simplex or bounded by the simplex, the algorithm will be more efficient and more accurate. Therefore, the initial simplex plays an important role in this method since it decides the degree of the reduced computational complexity and the decoded video quality. A simple initialization method for downhill simplex search is to find three points around the center of the current block. This method works well for blocks with small motion vectors. However, the performance decreases when the motion vectors are large. In this work, we propose an initialization method to select an appropriate initial simplex from motion prediction results.

As the video coding standards predict motion vectors in the encoding processes, we can predict the current motion vector from the estimated motion vectors available in the neighboring blocks at the current or previous frames. As shown in Fig. 4, the MVs of the neighboring blocks at the current and previous frames are utilized. We average the motion vectors of block B, C, D, and E to obtain a candidate, called MV_{c1}, and average the motion vectors of block G, H, I, and J in the previous frame to obtain another candidate, called MV_{c2}. Besides, the motion vector of block F and the zero vector (0, 0) are chosen as candidates MV_{c3} and MV_{c4}, respectively.

There are two possible ways to determine an appropriate initial simplex from the four candidates. One is to choose the candidate with the smallest SSE as the starting center and then find three nearest neighboring grid points to be an initial simplex. The other is to choose three points from the four candidates directly to form an initial simplex. The former can provide a more localized simplex, and the latter has the advantage of better computational efficiency. Because we focus on the computational efficiency, the latter is adopted in our experiments.

2.3. Initial Simplex Selection for Multi-Reference-Frame ME

For the selection of the initial simplex for the multi-reference-frame motion estimation, we consider both spatial and temporal properties. For the spatial property, most of the motion vectors are small and distributed around zero motion vector (0, 0). Regarding the temporal property, it is most probable that the best MVs occur in the most recent reference frame. However, more reference frames can provide better prediction results especially when there is occlusion in a video sequence or the image frame contains rich textures.

In many multi-reference-frame motion estimation algorithms, the same algorithm used in the single-reference-frame ME is applied to each of the multiple reference frames directly to find motion vectors. Consequentially, the computational complexity increases a lot as the number of reference frames increases. It is also inefficient to apply the downhill simplex search to each previous frame to find motion vectors. Since the downhill simplex search uses n+1 points for the n-dimension search space, a collection of four points are needed to form the simplex for the 3-D search space in multi-reference motion estimation.

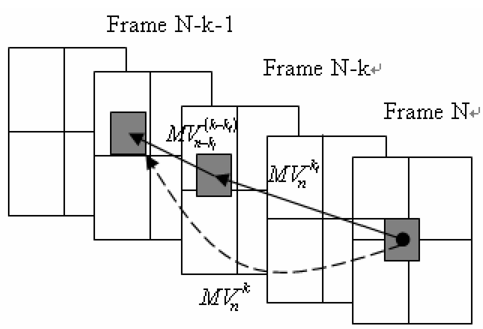

Fig. 5. Tracing motion trajectories.

Generally, the motion field varies slowly and smoothly. The correlation between motion vectors of neighboring blocks in temporal domain can be exploited to find the initial simplex. In [12], the motion vector in the previous frame can be traced along the motion trajectories and composed by:

$$MV_n^{-k} = MV_n^{-k_l} + MV_{n-k_l}^{-(k-k_l)} \qquad (6)$$

where MV_n^{-k} represents the motion vector of frame n referring to the previous k-th frame. Additionally, it can be composed by the motion vector of frame n referring to the previous k_l-th frame and the motion vector of frame n-k_l referring to the previous (k-k_l)-th frame. Fig. 5 shows the relations between motion vector and its trajectory. For example, MV_n^{-5} can be composed of $\mathrm{MV}_n^{-4} + \mathrm{MV}_{n-4}^{-1}$.

In this work, the approximated motion vectors by tracing motion trajectories in the reference frames are adopted to form the initial simplex. The steps are as the following: In each frame, the single-reference downhill simplex search is applied to find the motion vector in the previous frame first. In other words, k_l equals to one in our experiments. Secondly, motion vectors in any other reference frames are composed from the previous results. For example, if five reference frames are used and the current frame number is six, MV_6^{-1}, MV_5^{-1}, ..., and MV_2^{-1} are determined by the single-reference downhill simplex search in the first step. In the second step, MV_6^{-2} can be composed by MV_6^{-1} + MV_5^{-1}. Then, MV_6^{-3} can be composed by MV_6^{-2} + MV_4^{-1} and so on. In the last step, four motion vectors among all candidates with the minimal SSE values are chosen to form the initial simplex.

2.4. Stopping Criterion

In addition, the stopping criterion used in the proposed downhill simplex search algorithm is very intuitive. If two of the three points, P_h, P_m, and P_l, lie at the same point, the simplex has degenerated, then the iteration should stop, i.e., the iteration terminates when

$$P_h = P_m \quad or \quad P_h = P_l \quad or \quad P_m = P_l \qquad (7)$$

3 Improved Schemes

In this work, we present some improved schemes used in the downhill simplex search ME algorithm to achieve better efficiency and compression quality.

3.1. New location rounding scheme

In the downhill simplex search, averaging the points or in the shrinkage step may cause fractional point coordinates. Accurate interpolation techniques for computing the function values at the fractional points require more computational cost. Although the simplest rounding method can be used to round the search location to the nearest integer point, it may degrade the coding accuracy. In this paper, we propose a special rounding scheme to avoid sophisticated interpolation or simple rounding. As shown in Fig. 6, (x,y) is a point with fractional position. The coordinates (i,j), (i+1,j) and (i,j+1) are three neighboring integer points. The function F represents the SSE function of the block. In our special rounding scheme, F(i,j), F(i+1,j) and F(i,j+1) are calculated since the purpose of BMA is to find the most similar block with the smallest SSE value and the downhill simplex search uses a triangle in 2D space to minimize function values. The location with the smallest SSE value among the three nearest integer neighbors can be used as the rounding result for the point (x,y). If (x,y) is located between two points or falls in the center of four points, two or four neighbors are used. In multiple reference frames, three neighboring integral points with the same temporal displacement are compared first. Then, the smallest is chosen from all candidates.

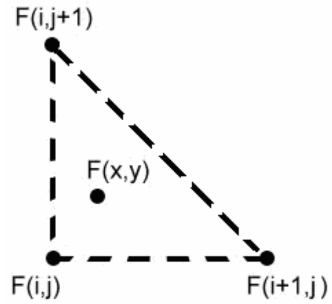

Fig. 6. A new location rounding scheme.

3.2. Early-stop scheme

In the downhill simplex search, the point with the highest SSE, namely P_h, is iteratively replaced with a better point. However, the newly calculated points usually have a higher SSE than that of the current P_h. Therefore, when we compute the SSE of the new point, the SSE computation can be terminated as soon as the accumulative SSE of the new location exceeds the SSE of P_h. The early-stop scheme can be applied in the iteration steps or refinement of the downhill simplex search. It helps reduce the computational load greatly. It is more significant in SSE than in the sum of absolute differences (SAD) error measure. Note that, in our experiments, we use *the total number of effective search locations* to measure the efficiency of BMA. When the SSE accumulation is aborted due to the early-stop scheme, the portion of SSE that has been calculated is counted as a fractional search location depending on when it was terminated in our experiment.

3.3. ME Refinement

Sometimes, the search results may converge to suboptimal points. They are usually very close to the global minimum. Therefore, a one-pixel refinement is provided to search the eight nearest neighbors after the convergence of the iterative process. Note that it only slightly increases the computational cost of the motion estimation because most of these neighbors have already been searched.

4 Experimental Results

We compared six block matching algorithms, including full search (FS), four step search (FSS), diamond search (DS), simplex minimization search (SMS), our proposed downhill simplex search (DSS) and multi-reference full search (MR-FS) with our multi-reference frame downhill simplex search (MR-DSS) through experiments on four benchmarking video sequences (foreman, coastguard, news, and container). The foreman sequence is a popular video because it contains different motion directions and large motions in the video. The coastguard sequence contains fast movement through the whole sequence. The news sequence almost remains static in most areas except the small area around the human face. The container sequence contains slow and uniform motions. The test video sequences are selected because they represent different types of video motions. In our experiments, all the video sequences are of QCIF format. For different sequence length, we compute the average number of search locations and the PSNR for each frame. The formats of these sequences are listed in Table 1. Before evaluating the performance of the proposed algorithms, the reflection coefficient α , the contraction coefficient β and the expansion coefficient γ are set to $\alpha = 1, \beta = 0.5, \gamma = 2$ according to the experimental results given in [16], which provides a good compromise between speed of estimation and coding quality. In our experiments, the BDM was defined to be the SSE, the block size was set to 16 by 16 pixels, the maximum allowed motion displacement was ± 16 pixels, and the search was performed to full-pixel accuracy. The experiments focus on the estimation speed and the prediction quality. The term *Number of Effective Search Locations* represents the estimation speed. In the search of motion vectors for one macroblock, the number of search locations is increased by 1 when a candidate block is chosen and the SSE is accumulated for this block. The accumulation can be aborted due to the early-stop scheme. The portion of SSE that has been calculated for one macroblock is counted as a fractional effective search location. At last, the number of search locations for all blocks of all frames in a sequence shows the estimation speed for a specific search algorithm. In addition, PSNR is calculated for compressed video quality assessment. In other words, motion is estimated and compensated using the original, rather than the reconstructed, reference frame for each frame. This provides a particularly fair comparison between the algorithms on a frame-by-frame basis since poor prediction of one frame does not propagate to the next frame.

4.1. Single-Reference ME Experiments

Table 1. The four test video sequences used in our experiments.

Sequence Name	Resolution	Frames
Foreman	176 x 144	320
Coastguard	176 x 144	97
News	176 x 144	200
Container	176 x 144	180

Table 2. Simulation Results : PSNR(dB), Number of Effective Search Locations per Frame (reference frame length = 5).

BMA	Foreman		Coastguard		News		Container	
	PSNR	Locations	PSNR	Locations	PSNR	Locations	PSNR	Locations
FS	32.21	82104	33.25	82104	37.64	82104	42.17	82104
FSS	31.73	1703.99	33.13	1507.57	37.61	1255.12	42.17	1219.58
DS	31.77	1595.29	33.17	1054.94	37.61	965.75	42.16	921.59
SMS	31.31	1106.33	32.53	1478.89	37.54	1045.95	42.13	1043.76
DSS	31.94	645.58	33.23	548.96	37.59	515.96	42.13	499.73
MRFS	33.15	387195	33.61	387195	37.80	387195	42.42	387195
MRDSS	32.47	1721.16	33.41	1380.87	37.65	1080.37	42.21	1004.61

Tables 2 summarizes the PSNR and computational costs for the motion estimation on the four test video sequences by using different ME algorithms. It also shows the prediction quality of the simulated BMAs in terms of average PSNR in decibels. As shown in Table 2, our proposed method significantly outperforms the other methods on the foreman sequence and coastguard sequence because they contain larger MVs. When the MVs are large, most fast BMAs, such as DS and FSS, normally require more computational cost for motion estimation, while the proposed DSS algorithm is quite stable for videos with different types of motion. Fig. 11 and Fig. 12 show the number of effective search locations for all aforementioned BMAs on the foreman and coastguard video sequences, respectively. The proposed DSS algorithm has the lowest numbers of effective search locations in both sequences. It is also evident from the experiments that the numbers of effective search locations required in the proposed method for different types of video sequences are quite stable. For the PSNR comparison shown in Fig. 7 and Fig. 8 for the same sequences, our proposed method apparently has higher PSNR than other fast BMAs for both video sequences. The news sequence and container sequence contain smaller motion, so the PSNR differences between the proposed algorithm and the other fast BMAs are insignificant, as shown in Fig. 9 and Fig. 10. However, our proposed method can still find solutions with much less computational cost than those of other fast BMAs on these two sequences, as shown in Fig. 13 and Fig. 14. In fact, our proposed method is much faster than

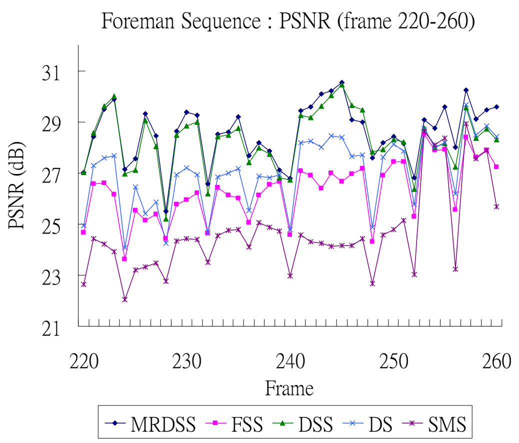

Fig. 7. PSNR values for different BMAs on Foreman sequence from frame 220 frame 260.

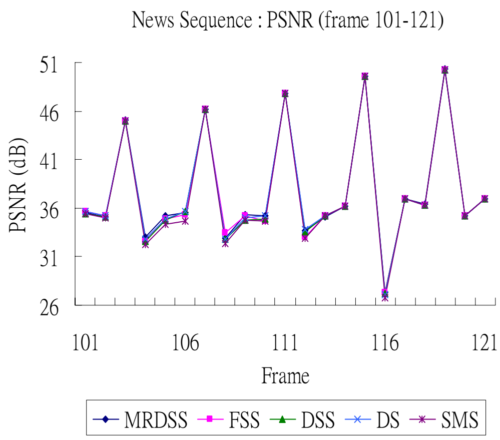

Fig. 9. PSNR values for different BMAs on news sequence from frame 101 to frame 121.

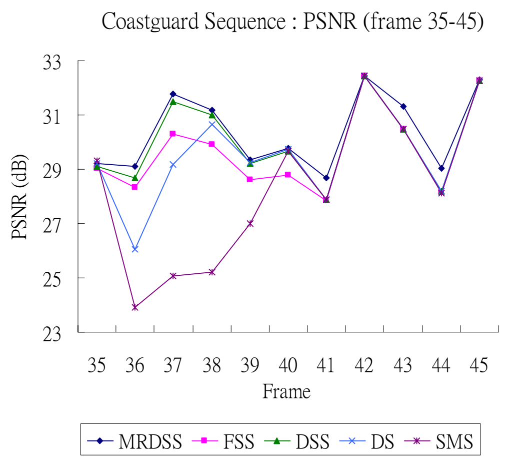

Fig. 8. PSNR values for different BMAs on coastguard sequence from frame 35 to frame 45.

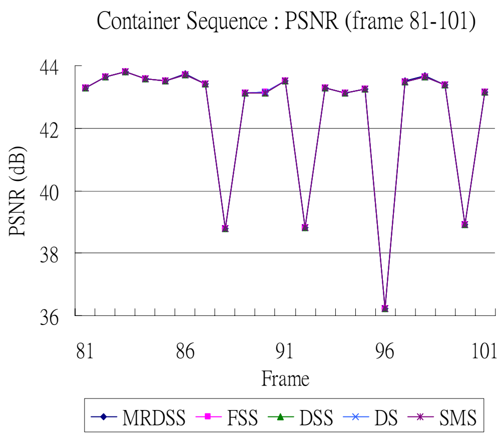

Fig. 10. PSNR values for different BMAs on container sequence from frame 81 to frame 101.

other BMAs for all the test videos in our experiments. This is mainly due to the reasons that the initial simplex is carefully selected, the downhill simplex search is efficient and the early-stop scheme further enhances the search speed. On the other hand, the PSNR of our proposed method is higher than other fast BMAs in video sequences with large motion (foreman and coastguard sequences) and is similar to other fast BMAs for video sequences with small motion (news and container sequences). Figures 11-14 show the number of effective search locations in these motion estimation methods for some periods of frames in these test sequences. The number of effective search locations in the proposed method is consistently the least among all these BMAs and the computation required in our algorithm does not frustrate significantly even when the MVs during some short periods of a video sequence are quite large.

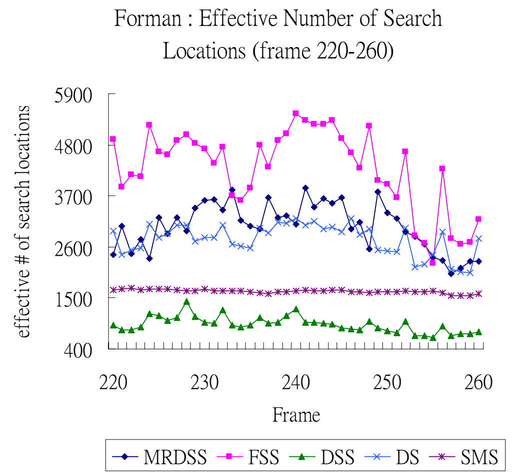

Fig. 11. The number of effective search locations for foreman sequence during frames 220-260.

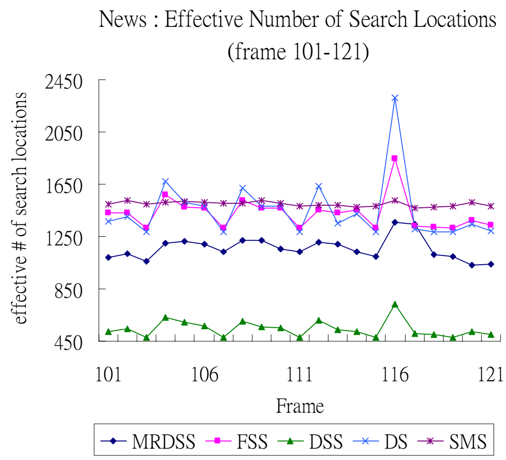

Fig.13. The number of effective search locations for news sequence during frames 101-121.

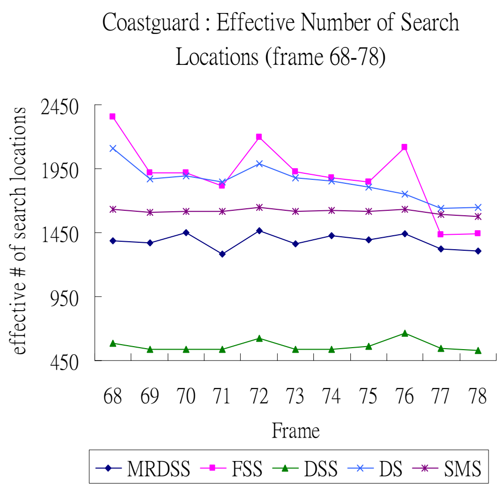

Fig.12. The number of effective search locations for coastguard sequence during frames 68-78.

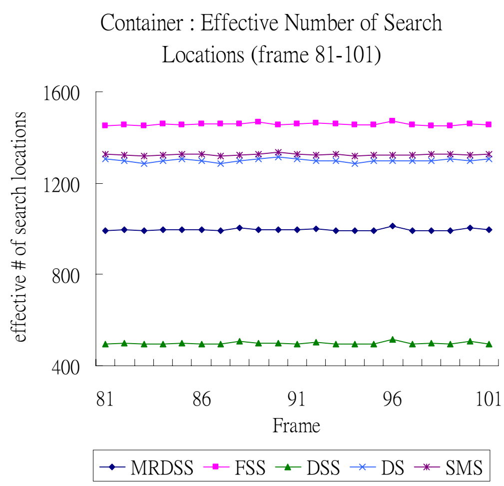

Fig.14. The number of effective search locations for the container sequence during frames 81-101.

4.2. Multi-Reference Downhill Simplex Search

In the experiment of multi-reference downhill simplex search, the total number of reference frames is set to 5, which complies with H.264 coding standard. The proposed MR-DSS performs a two-level downhill search. One level is in the initial step. Each frame uses the single-reference DSS to search the motion vectors in the previous one frame. These motion vectors are used to compose MVs in any other reference frames, which become the candidates of the initial simplex. Four points with minimum SSE values are chosen to form an initial simplex. On the second level, a 3D-version DSS is applied to obtain more accurate motion vectors. As shown in Table 3, our proposed MR-DSS provides significant reduction in computational cost. Compared with MR-FS, a speed-up ratio ranges from 225 to 385 in our experiments. Table 3 represents the speed-up ratio of the proposed MR-DSS algorithm with reference to the FS algorithm. The speed-up ratio R is defined as the ratio between the total number of search locations in MRFS and the total number of search locations in MRDSS. Table 4 shows the execution time required for different motion estimation methods on some test sequences. DSS has the shortest execution time in windows XP environment on P4 3.0 platform. In multi-reference downhill simplex search, our proposed method may compete with many single-reference block matching algorithms from the aspect of speed and it can achieve higher PSNR than the other fast BMA in all the cases.

Table 3. Computational speed-up ratio R of the proposed algorithm on the four video sequences.

Sequence	Computation Reduction R
Foreman	224.96
Coastguard	280.40
News	358.39
Container	385.42

Table 4. Time complexity for different motion estimation methods (ms / frame).

	Foreman	Coastguard	News
FS	485.41	480.52	573.3
FSS	20.75	18.25	18.75
DS	20.88	16.91	17.20
SMS	19.63	14.95	17.80
DSS	13.13	11.65	13.45
MRFS	2371.63	2277.53	2852.09
MR-DSS	20.31	17.94	18.30

5 Conclusion

In this paper, a fast motion estimation algorithm is proposed by using an enhanced downhill simplex search for function minimization in a finite-dimensional search space. The proposed downhill simplex search ME algorithm contains a new initial simplex selection method based on motion prediction, a special location rounding scheme, and an early-stop function evaluation to reduce the computational complexity. A special feature of the proposed motion estimation method is the very stable computational cost for videos of different motion types, which is evident from the experimental results.

We also extend the single-reference frame downhill simplex search algorithm to multi-reference frame problem. To apply the downhill simplex search for multi-reference motion estimation, we select tracing motion trajectories to form a good initial simplex so that it makes the downhill simplex search very efficient. In addition, a special location rounding scheme and an early-stop strategy also help further improve the search speed and accuracy in motion estimation. Experimental results on several video sequences of different types show the superior performance of the proposed enhanced downhill simplex search algorithm over some well-known fast motion estimation methods.

Acknowledgement

This work was supported in part by MOEA project under grant 95-EC-17-A-01-S1-034 and National Science Council, under grant

95-2220-E-007-028.

References

[1]. ISO/IEC 11172, Information technology - coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s, 1993.

[2]. ISO/IEC 13818, Information technology - Generic coding of moving pictures and associated audio information, Part 2: Video, 1995.

[3]. ISO/IEC 14496-2. Information technology - coding of audio-visual objects - Part 2: Visual, 1998.

[4]. ITU-T Recommendation H.263 Video coding for low bit rate communication, Version 2, 1998.

[5]. ISO/IEC 14496-10 and ITU-T Recommendation H.264. Advanced Video Coding, 2003.

[6]. Koga, T., Iinuma, K., Hirano, A., Iijima, Y., and Ishiguro, T., "Motion compensated interframe coding for video conferencing." Proc. Nat. Telecommun. Conf., New Orleans, LA, Nov. 29-Dec. 3, (1981), G5.3.1-G5.3.5.

[7]. Li, R., Zeng, B., Liou, M.L.,"A new three-step search algorithm for block motion estimation." IEEE Trans. Circuits Syst. Video Technol., Vol. 4. Aug, (1994), pp.438-442.

[8]. Po, L.M., Ma, W.C., "A novel four-step search algorithm for fast block motion estimation." IEEE Trans. Circuits Syst. Video Tech., Vol. 6, June, (1996), pp. 313-317.

[9]. Zhu, S., Ma, K.-K., "A new diamond search algorithm for fast block-matching motion estimation." IEEE Trans. Image Processing, Vol. 9, Feb, (2000), pp. 287-290.

[10]. C.-W. Ting, L.-M. Po and C.-H. Cheung, "Center-biased frame selection algorithms for fast multi-frame motion estimation in H.264," IEEE Int. Conf. Neural Networks & Signal Processing, Nanjing. China, December 14-17, 2003.

[11]. C. W. Ting, W. H. Lam, and L. M. Po, "Fast block-matching motion estimation by recent-biased search for multiple reference frames," Proc. IEEE International Conference on Image Processing, Singapore , Oct. 2004, pp. 1445-1448.

[12]. Y. Su and M.T. Sun, "Fast multiple reference frame motion estimation for H.264," IEEE International Conference on Multimedia and Expo (ICME), 2004.

[13]. M.E. Al-Mualla, C.N. Canagarajah, and D.R. Bull, "Simplex minimization for single- and multiple-reference motion estimation," IEEE Trans. Circuits Syst. Video Techn, VOL. 11, NO. 12, Dec. 2001, pp. 1209-1220.

[14]. M.E. Al-Mualla, C.N. Canagarajah and D. R. Bull, "Simplex minimization for multiple-reference motion estimation," Proc. IEEE International Symposium on Circuits and Systems, Geneva, Switzerland, May 2000.

[15]. Nelder, J.A., Mead, R. "A simplex method for function minimization." The Comput. J., Vol. 7, 1965, pp. 308-313.

[16]. M. E. Al-Mualla, "Video Coding for Mobile Communications: A Motion-Based Approach," Ph.D. dissertation, Univ. of Bristol, Faculty of Engineering, Dept. of Elect. and Electron. Eng., Bristol, U.K., 2000.

Chen-Kuo Chiang received his B.S. degree in computer information science from National Chiao Tung University, Hsinchu, Taiwan, in 1998, and M.S. degree in computer science information engineering from National Taiwan University, Taipei, Taiwan, in 2000. He was an engineer from 2001 to 2005 at Institute of Information Industry. He is currently pursuing his Ph.D degree in computer science at National Tsing Hua University, Hsinchu, Taiwan.

Shang-Hong Lai (M'95–) received the B.S. and M.S. degrees in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, and the Ph.D. degree in electrical and computer engineering from University of Florida, Gainesville, in 1986, 1988 and 1995, respectively. He joined Siemens Corporate Research in Princeton, New Jersey, as a member of technical staff in 1995. Since 1999, he returned to Taiwan as a faculty member in the Department of Computer Science, National Tsing Hua University. He is currently an associate professor in the same department. In 2004, he was a visiting scholar with the Department of Electrical Engineering, Princeton University. Dr. Lai's research interests include computer vision, visual computing, pattern recognition, medical imaging, and multimedia signal processing. He has authored more than 90 papers published in the related international journals and conferences. Dr. Lai holds ten US patents for inventions related to computer vision and medical image analysis. He has been a member of program committee of several international conferences, including CVPR, ICCV, ECCV, ACCV, ICPR and ICME.