AN ADJUSTABLE BLOCK MOTION ESTIMATION ALGORITHM BY MULTIPATH SEARCH

Thou-Ho (Chou-Ho) Chen
Department of Electronic Engineering, National Kaohsiung University of Applied Sciences
thouho@cc.kuas.edu.tw

Chou-Yu Chen
Department of Electronic Engineering, National Kaohsiung University of Applied Sciences
1093320122@cc.kuas.edu.tw

ABSTRACT

A speed-quality adjustable block-matching algorithm (BMA) based on multipath flatted-hexagon search (MFHS) is developed for motion estimation. To improve the accuracy of fast BMA near to that of full search (FS), the problem of being trapped at the local minimum block distortion measure (BDM) should be overcome. In the proposed method, an adaptive threshold of BDM is introduced to determine the required search directions in order to escape from being trapped at a minimum BDM, followed by a flatted-hexagon search performed in the direction according to the BDM below a threshold. Then, the motion vector will be refined at each search step unite the searching process is stopped. The BDM threshold will be adaptive for the purpose of adjusting the search speed and matching probability required in the specific application. Experimental results show that the proposed MFHS algorithm can achieve a matching probability upto 99.35% and 13.85 times of search speed of FS in some certain sequence.

Keyword: Multipath search.

1. INTRODUCTION

Motion estimation can make the interframe coding to achieve a very high compression ratio, when compared to the intraframe coding, by exploiting the heavy temporary redundancy between successive frames. Among various motion estimation techniques, the block-matching algorithm (BMA) is the most attractive method for the current international video compression standards including H.261, H.263, H.264, MPEG-1, MPEG-2 and MPEG-4 [3]-[5] [16] [17], because of its effectiveness and simplicity for implementations. However, the matching process of finding the optimal still involves a large amount of calculations, e.x., the most accurate approach, called the full search (FS) method which requires to evaluate all candidate blocks, can consume at least 63% of the computational power for the MPEG-4 encoder. To reduce the intensive computational complexity with a tolerable distortion, many fast block-matching algorithms were developed [1] [2] [7]-[15].

Among the above suboptimal methods, both the search pattern's attributes and initial searching range always directs the developmental processes of these algorithms. By taking advantage of the characteristics of the center-biased motion vector distribution existed in most real-world image sequences, the new three-step search (N3SS) [11], four-step search (4SS) [10] and block-based gradient descent search (BBGDS) [9] perform better than the three-step search (3SS) [15], where these four search patterns are square-shaped. Based on a practical compact-shaped pattern with fewer candidate search points per block, a diamond-search (DS) algorithm [7][12] can not only improve the searching speed but also reduce the chances of being trapped in local optimal, when compared to those four algorithms. The hexagon-based search (HEXBS) algorithm [2] utilized a hexagon-shaped pattern with only 7 checking points in the initial search and 3 checking points in the following searches to achieve substantial speed improvement over the DS algorithm with similar distortion performance for most high-resolution (e.x., 720×480) image sequences. Nevertheless, the matching-probability performance will degenerate with the decreasing resolution of the video format. To obtain a faster searching speed than the DS algorithm while maintaining similar search quality, the cross-diamond search (CDS) algorithm [1] employed a cross search pattern at the initial step to exploit the characteristics of the center-biased motion vector distribution very efficient, followed by the halfway-stop technique, and the large/small diamond search patterns in the subsequent steps. Although, various search patterns and processes at different steps will make the CDS algorithm to be complicated on realization, especially for VLSI implementation due to its favoritism of regularity [14]. Flatted hexagon searching (FHS) [13] method used the simple strategy which is different from CDS and solves the problem which is not enough accuracy of HEXBS. The covering range of a search pattern should be enlarged as horizontal as possible to find the optimal motion vector quickly because the probability of horizontal-biased motions is larger than that of vertical-biased motions in most of the real-world image sequences. However, those previous fast BMAs can't achieve a similar matching probability as that of FS under a moderate search speed, especially for applications in which a higher accuracy of motion estimation is required.

To achieve a matching accuracy near to FS with a larger search speed than FS, this paper introduces a multipath flatted hexagon search (MFHS) strategy. It can effectively solve the problem which is trapped at local optimal to improve matching the probability. And, it can adjust the searching speed and matching probability for your requirements.

2. ANALYSIS OF THE SEARCHING STRATEGY

Almost conventional block motion estimation algorithms are explicitly or implicitly based on the assumption: BDM increases monotonically as the checking point moves away from the global minimum. Obviously, this assumption essentially requires that the error surface is unimodal over the search windows. Unfortunately, this is usually not true due to many reasons such as the aperture problem, the textured (periodical) local image content, the inconsistent block segmentation of moving object and background, the luminance change between frames, and etc. As a consequence, the search would easily be trapped at a local minimum.

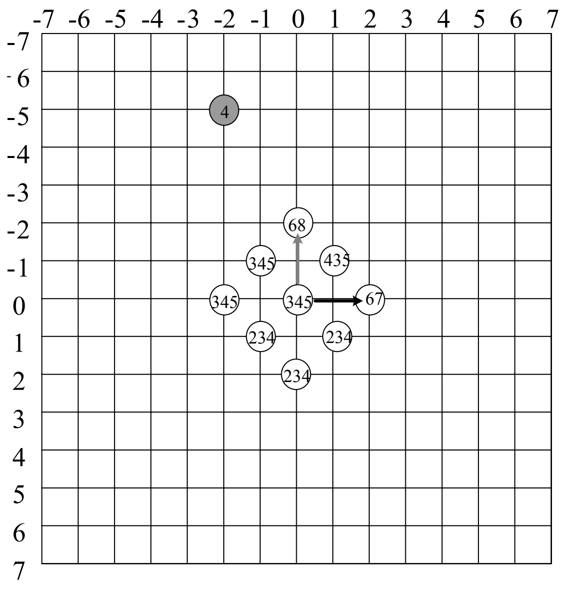


Fig. 1 The example of false searching for direction

The general fast searching methods use search pattern to search the direction of a minimum BDM, and close step by step in minimum BDM. But, they can cause the faulty searching direction due to all above-mentioned reasons while beginning to search BDM, especially they have two or more minimum BDM. When the searching direction is faulty, they usually difficult to go back correct direction and then they are trapped at a local minimum. Fig. 1 is the example which is the faulty searching direction of DS. Among the example, the value of a circle is the BDM value which is calculated. The charcoal gray circle (-2,-5) is the location of a true motion vector. DS calculate first 9 checking points, and the minimum value which is located at (2,0) is 67. Then, DS is based on this point to build a new diamond pattern. So, the search pattern goes forward from the direction of that black arrow. But, the minimum BDM of the example is located at (-2,-5). As a result, the example should be gone forward the direction of the second minimum BDM, namely the direction of the gray arrow is correct. We often ignore the importance of the second minimum value. In fig.1, if we can search the directions of the first and second minimum values at the same time, the faulty matching probability can be to decrease. However, when should we to search the direction of the second minimum value? Whether the direction of the third minimum value also should to search or not? We should compare the BDMs of the points of all directions, and we judge whether the values and minimum BDM are similar. If yes, we will search the points. This guides the basic idea of our adjustable fast motion estimation.

3. METHODOLOGY

3.1 Multi-path search algorithm

We use block matching criteria which is SAD, described as follows:

$$SAD(i,j) = \sum_{m=1}^{M}\sum_{n=1}^{N}\left| f(m,n) - f'(m+i,n+j)\right| \qquad (1)$$

(i,j) is a candidate motion vector. $f(m,n)$ is the pixel intensity at (m,n) of the present frame. $f'(m+i,n+j)$ is the pixel intensity at $(m+i,n+j)$ of the reference frame. For solving the problem of searching the faulty direction, we propose a multi-path search algorithm. We define a function as eq. 2. SAD_{current} is the SAD value of every new checkpoint of a search pattern. $SAD_{\min}$ is the minimum SAD value for all checked points, and T is a threshold value. If the correlation of SAD_{current}, $SAD_{\min}$ and T accords with eq. 2, we will build a new search pattern whose center is the location of SAD_{current}. If not, we stop the algorithm.

$$SAD_{current} - SAD_{\min} \le T \qquad (2)$$

3.2 Dynamic threshold

The same value of T can cause some different effective results on various image sequences due to various video features. As a result, we present a scheme which is adaptive adjust threshold for various video contents and features.

$$T = SAD_{\min} \times \beta \qquad (3)$$

After our test and verification, T can make the adjustability multi-path search method speed and accurate rate better through simple processing, and the adaptability is stronger. The β value, which is between 0 and 2, can satisfy various videos through our test. If $\beta = 0$, the performance of MFHS is similar to FHS. In this paper, we use fixed-valued of β to obtain a dynamic threshold value.

3.3 Selection of search pattern

Because the adjustable multi-path search algorithm has not restricted to the search pattern whose kinds, it usually uses much simpler search pattern then much easier realization. The matched search patterns are the search patterns of 3SS, 4SS, DS, HEXBS and FHS which we proposed. But the search pattern of CDS is difficult to apply due to the method of CDS is complexity. The characteristics of original search pattern and applied environment cause what kind of search patter is suitable for the adjustable multi-path search algorithm. In solutions of QCIF, QSIF, CIF and SIF, we suggest the search pattern of FHS to use because the size of the search pattern of FHS is small and suitable for lower solution. We suggest the search pattern of DS to use in solution of CCIR601. In the paper, we use mainly the search pattern for multi-path search algorithm because the number of checkpoints is less and the probability is more accurate than DS.

3.4 The adjustable multi-path flatted hexagon search algorithm

The algorithm of the adjustable multi-path FHS (MFHS) is described as follows:

Step 1. The center of search window is starting point, and we calculate the 7 checkpoints of FHS. We find the minimum BDM out which is defined as $SAD_{\min}$. We judge whether the 7 checkpoints are suitable for eq. (2). If yes, we record the corresponding locations of the suitable checkpoints, and go to Step 2.

Step 2. If one of the suitable checkpoints is located at the center of search window, we build the ending pattern. Otherwise, the every checkpoint is set the center of new FHS, and every FHS adds three new checkpoints. We find the minimum BDM out from these new checkpoints, and judge whether the minimum BDM is less than $SAD_{\min}$. If yes, the minimum BDM is defined as $SAD_{\min}$. We judge whether the new checkpoints are suitable for eq. (2). If yes, we record the corresponding locations if the suitable checkpoints and repeat step 2. If the method is not built a new FHS, we will stop the algorithm. The location of $SAD_{\min}$ is the motion vector.

Fig. 2 is the example of MFHS. We define T as 25. The motion vector is (4,-1) and we cost 24 checkpoints to obtain the motion vector.

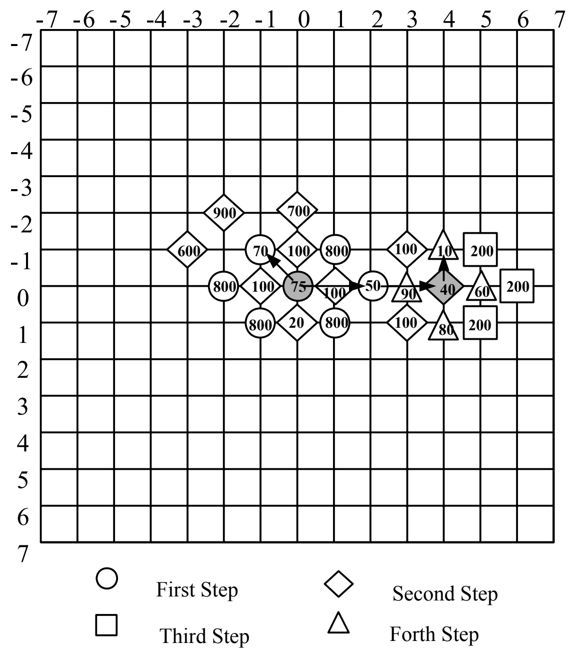


Fig. 2 The searching example of multi-path flatted hexagon search algorithm.

4. EXPERIMENTAL RESULTS

A theoretical analysis about those fast BMAs has been given in the above section, but the implementation of such fast algorithms with several representative sequences of various motion contents can provide a realistic and interesting evaluation. For the purpose of comparison, six previous fast BMAs including 3SS, 4SS, N3SS, DS, HEXBS and CDS and the proposed MFHS algorithm (with various β) are simulated by using the luminance component of five popular sequences: "Salesman" (CIF, 499 frames), "Coastguard" (CIF, 300 frames), "Garden" (SIF, 115 frames), "Tennis" (SIF, 67 frames) and "Football" (SIF, 125 frames). Evaluation with such five sequences in terms of MAD (mean absolute distortion) used as the BDM, matching probability (i.e., the probability of finding the true motion vector) and number of search points is described in Table 1. For each sequence, the search is performed at a block size of 16×16 within a window of size ±7. In Fig. 3(a), we describe that the various β within MFHS influenced the numbers of the checkpoints on five sequences. In Fig. 3(b), we describe that the various β within MFHS influenced the accurate probability on five sequences. Fig. 4 shows the motion compensated results at various β on tennis sequence.

In Fig. 3(a), we can know that the various β within MFHS influenced the speed on various image sequences. The correlation of β and the number of checkpoints is roughly linear on various image sequences. Fig. 3(b) shows the correlation of β and probability. The general fast motion estimations are difficult to maintain the higher probability in complex motion vector. However, MFHS has a characteristic which can guarantee the probability in sufficient β. From Fig. 3(b) and Table 1, the MFHS has another feature which we add a little β to increase the accurate probability and the checkpoints are not added a lot.

In Table 1, the feature of Salesman sequence is highly centralized motion vector, and Coastguard sequence is the smaller motion vector. MFHS is higher probability and less checkpoints than the other methods on these sequences. When $\beta = 0.5$, the probability and MAD of MFHS are close to FS, and they are 14.17 times and 11.33 times than the probability and MAD of FS, respectively. Fig. 4 shows that the motion compensated results of the 66th frame of the "Tennis" sequence with various β, and we can see the phenomenon of the improvement is slowly increase in accurate rate. When $\beta = 0.36$, the compensated frames of MFHS are similar to FS.

5. CONCLUSIONS

The advantages of MFHS are adjustable search speed and matching probability. And, it can adaptive for various video contents and features. MFHS makes use of the adaptive threshold for keeping higher probability on various image sequences. If you want to obtain the close probability to FS, you can set β which is between 0.36 and 2 and MFHS is faster than FS. If you want to get faster speed, you can set β which is below 0.22. But, we don't suggest that you defined β as 0 because the smaller value of β can increase the accurate probability and reduce the search speed. MFHS is the higher probability method for the general methods. If you want to adjust the speed and probability for various image sequences, MFHS is what we are recommended.

6. REFERENCES

[1] 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.

[2] Ge Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation", IEEE Trans. Circuits Syst. Video Technol., vol. 12, No. 5, May 2002.

[3] Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s–Part 2: Video, ISO/IEC 11 172-2 (MPEG-1 Video), 1993.

[4] Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC 13 818-2 (MPEG-2 Video), 2000.

[5] Information Technology—Coding of Audio Visual Objects - Part 2: Visual, ISO/IEC 14 469-2 (MPEG-4 Visual), 1998.

[6] ITU-T Recommendation H.264/ISO/IEC 11496-10, "Advanced Video Coding", Final Committee Draft, Document JVC-E022, September 2002.

[7] 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, pp. 369-377, Aug. 1998.

[8] K. R. Rao and J. J. Hwang, Techniques and Standards for Image, Video, and Audio Coding. New Jersey: Prentice Hall, Inc., 1996.

[9] 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, pp. 419–423, Aug. 1996.

[10] 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, pp. 313-317, June. 1996.

[11] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438-443, Aug. 1994.

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

[13] T. H. Chen and Yi-Fan Li "A Novel Flatted Hexagon Search Pattern for Fast Block Motion Estimation", IEEE 2004 International Conference on Image Processing (ICIP), Singapore, Oct. 2004.(pp. 1477-1480)

[14] T. H. Chen," A Cost-Effective 3-Step Hierarchical Search Block-Matching Chip for Motion Estimation", IEEE Journal of Solid-State Circuits, vol. 33., No. 8, pp. 1253-1258, Aug. 1998.

[15] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc.

National Telecommunications Conf., New Orleans, LA, Nov. 1981, pp. G5.3.1-G5.3.5.

[16] Video Codec for Audiovisual Services at p × 64 kbits/s, ITU-T Recommendation H.261, Mar. 1993.

[17] Video Coding for Low Bit Rate Communication, ITU-T Recommendation H.263, Feb. 1998.

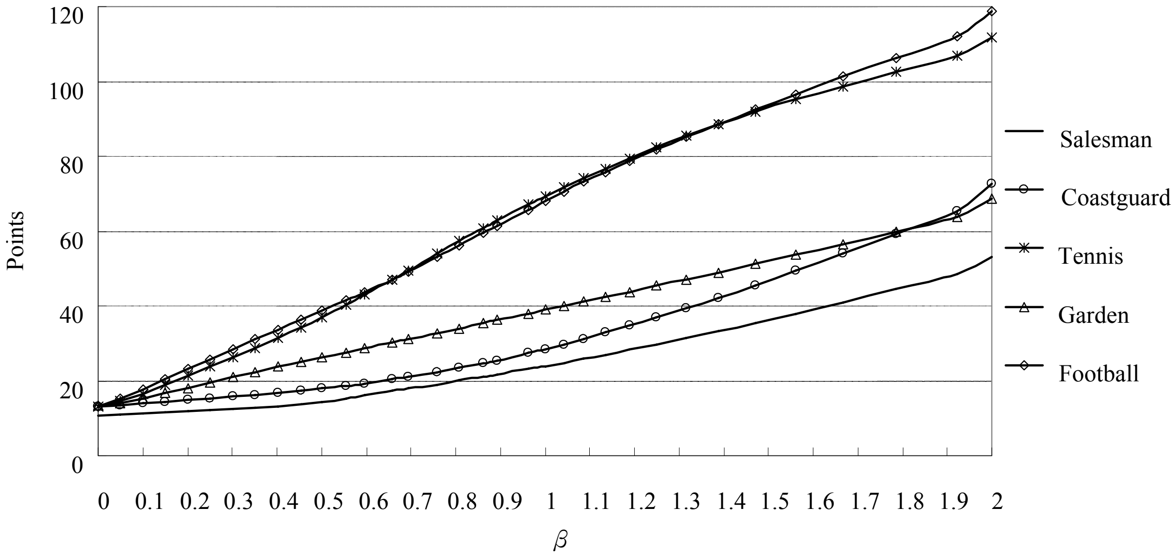
Table 1 Simulation results of MFHS and other BMAs

Salesman			
BMAS	**MAD**	**Pro.**	**Points**
FS	2.775	1.000	204.283
3SS	2.825	0.946	23.212
N3SS	2.781	0.979	16.776
4SS	2.817	0.953	16.144
DS	2.814	0.952	12.892
HEXBS	2.826	0.943	10.565
CDS	2.783	0.976	9.413
MFHS(β)	**MAD**	**Pro.**	**Points**
1	2.775	0.996	23.914
0.5	2.776	0.991	14.413
0.36	2.778	0.987	12.789
0.22	2.779	0.984	11.912
0.1	2.780	0.980	11.417
0.05	2.782	0.977	11.078
FHS(0)	2.785	0.966	10.640

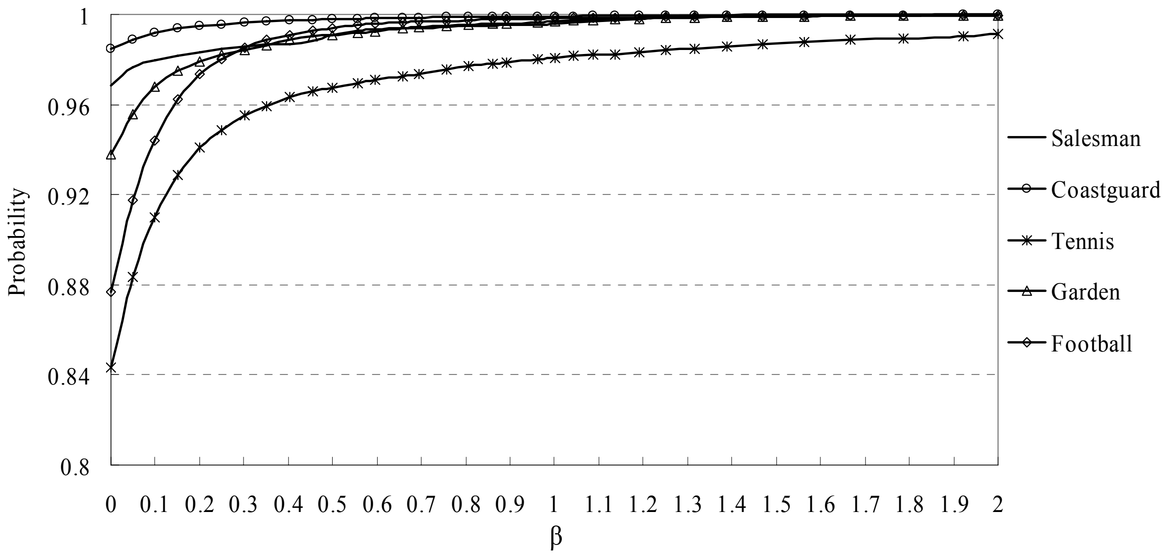
Tennis			
BMAS	**MAD**	**Pro.**	**Points**
FS	5.249	1.000	202.048
3SS	6.211	0.729	23.111
N3SS	5.656	0.840	20.293
4SS	5.659	0.850	18.455
DS	5.453	0.900	15.943
HEXBS	5.842	0.746	12.651
CDS	5.487	0.886	15.065
MFHS(β)	**MAD**	**Pr.**	**Points**
1	5.267	0.981	69.430
0.5	5.290	0.967	36.904
0.36	5.319	0.960	29.262
0.22	5.385	0.944	22.340
0.1	5.502	0.910	16.610
0.05	5.585	0.883	14.660
FHS(0)	5.718	0.843	13.093

Coastguard			
BMAS	**MAD**	**Pro.**	**Points**
FS	4.657	1.000	204.283
3SS	4.753	0.974	23.375
N3SS	4.698	0.978	19.786
4SS	4.726	0.984	18.500
DS	4.711	0.989	16.509
HEXBS	4.757	0.967	12.909
CDS	4.713	0.988	15.917
MFHS(β)	**MAD**	**Pro.**	**Points**
1	4.658	0.996	23.914
0.5	4.662	0.991	14.413
0.36	4.666	0.987	12.789
0.22	4.674	0.984	11.912
0.1	4.694	0.980	11.417
0.05	4.713	0.977	11.078
FHS(0)	4.738	0.966	10.640

Garden			
BMAS	**MAD**	**Pro.**	**Points**
FS	8.856	1.000	202.048
3SS	9.853	0.834	23.204
N3SS	9.005	0.930	21.139
4SS	9.463	0.868	18.696
DS	9.108	0.929	16.638
HEXBS	9.745	0.815	13.048
CDS	9.046	0.938	14.998
MFHS(β)	**MAD**	**Pro.**	**Points**
1	8.859	0.997	38.988
0.5	8.870	0.991	26.175
0.36	8.882	0.987	22.583
0.22	8.912	0.980	18.651
0.1	8.945	0.968	15.253
0.05	8.978	0.956	14.115
FHS(0)	9.040	0.937	13.281



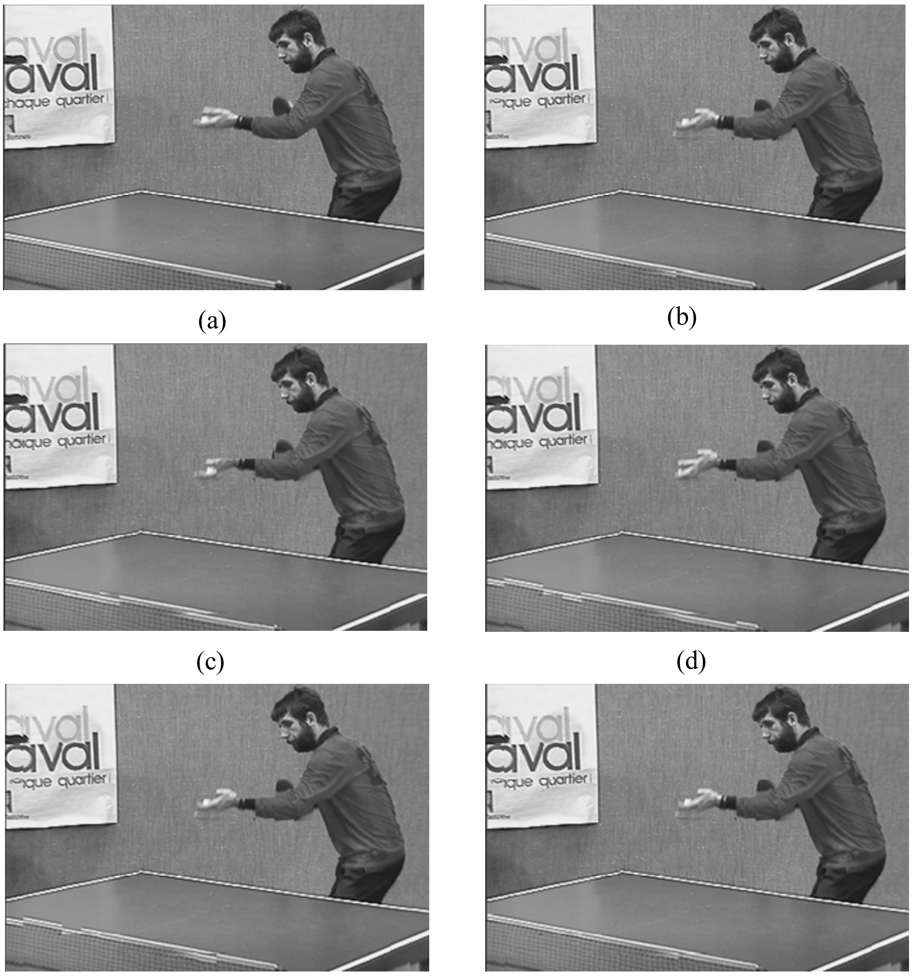
(a)



(b)

Fig 3 (a) The correlation of the β and checkpoints.

(b) The correlation of the β and probability.



(a) (b) (c) (d) (e) (f)

Fig4 A visual comparison using the SIF sequence " Tennis" motion-compensated for: (a) the 66th frame of the original sequence (non-compensated); (b)FS, MSE =155.7; (c)MFHS, β=0, MSE=233.2; (d) MFHS, β=0.1, MSE=198.1; (e) MFHS, β=0.22, MSE=178.8; (f) MFHS, β=0.36, MSE=163.2;