

Available online at www.sciencedirect.com



J. Vis. Commun. Image R. 17 (2006) 131-142



www.elsevier.com/locate/jvci

Markov model fuzzy-reasoning based algorithm for fast block motion estimation

Po-Hung Chen^{a,*}, Hung-Ming Chen^a, Kuo-Jui Hung^c, Wen-Hsien Fang^c, Mon-Chau Shie^c, Feipei Lai^{a,b}

^a Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan ^b Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan ^c Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan

> Received 19 April 2004; accepted 20 September 2005 Available online 3 November 2005

Abstract

This paper presents a Markov model fuzzy-reasoning based algorithm for fast block motion estimation. To reduce computational complexity, the existing fast search algorithms move iteratively toward the winning point based only on a finite set of search points in every stage. Despite the efficiency of these algorithms, the search process is easily trapped into local minima, especially for high activity video sequences. To overcome this difficulty, we propose a three-states Markov model based algorithm that invokes the fuzzy-reasoning to provide the search an acceptance probability of being able to move out of local minima. Two schemes are employed to further enhance the performance of the algorithm. First, a set of initial search points that exploit high correlations among the motion vectors of the temporally and spatially adjacent blocks as well as their surrounding points are used. Second, an alternate search strategy is addressed to cover more area without increasing computations. Simulation results show that the new algorithm offers superior performance with lower computational complexity and picture quality increase in terms of search points/block and MSE/pel, respectively, compared with the previous works in various scenarios.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Markov chain; Fuzzy reasoning; Block matching; Motion estimation; Video coding

1. Introduction

Motion estimation underlines the core of motion compensated predictive coding of video sequence. The block matching algorithms (BMA), which render efficient implementations, have in particular received a great amount of attention and have been adopted by modern video compression standards such as the H.26x, and MPEG-x, etc. [1].

* Corresponding author.

E-mail address: paully@orchid.ee.ntu.edu.tw (P.-H. Chen).

^{1047-3203/\$ -} see front matter @ 2005 Elsevier Inc. All rights reserved. doi:10.1016/j.jvcir.2005.09.001

The most straightforward BMA is the full search (FS) algorithm, which searches exhaustively over all allowable displaced points in the reference frame to locate the best match. The enormous amount of computations involved, however, has hindered it from practical implementations. To mitigate this, various attempts have been made to reduce the number of search points while, at the same time, without serious degradation of the reconstructed image quality. One of the most widespread BMAs is the simple yet effective three-step search (TSS) algorithm [2], which iteratively checks the winning point as well as the surrounding eight points with a diminishing window size. Several variants of the TSS such as the new three-step search (NTSS) algorithm [3], the four-step (FSS) algorithm [4], and the diamond search (DS) algorithm [5] were also addressed, aiming at locating the motion vector more precisely with reduced computations. All of these fast search algorithms, however, are based on the implicit assumption that the block distortion measure (BDM) monotonically increases around the global minimum [2]. The search thus moves iteratively toward the point that achieves the minimum BDM in every stage. Since this underlying assumption does not hold in many practical situations, the search process is therefore easily trapped into local minima if the point chosen in any stage is not the one that leads to the global minimum. Consequently, despite their efficiency, the search process is vulnerable to local minima, especially for high activity video sequences that contain lots of them.

To overcome this difficulty, several robust BMAs were advocated to refrain the search from being easily trapped into local minima. For example, Jan et al. [6] determined the motion vector based on the concept of "motion flow," which employs multiple candidates in the search process and thus is more robust against local minima. Tang et al. [7] decomposed the template matching into pixel and sub-pixel levels and considered a robust and accurate two-stage approach for motion estimation at the price of substantially more computations.

In this paper, a new algorithm by utilizing the fuzzy-reasoning technique [8] together with the local characteristics of the video sequences is proposed. As discussed above, the main problem of the current fast search algorithms is that they are "greedy" in the search process by only pursuing the winning point in every stage to reduce the computational complexity. These algorithms are inevitably susceptible to the local minima dilemma. In light of this, the proposed Markov model based algorithm invokes the fuzzy-reasoning to provide the search an acceptance probability of being able to jump out of local minima so that the algorithm is more robust against such undesired situations. In addition, two schemes are employed to further enhance the performance of the algorithm. First, a set of initial search points that exploit high correlations among the motion vectors of the temporally and spatially adjacent blocks are used. Second, an alternate search strategy is addressed to cover more area without increasing computations. As a consequence of these techniques, the new algorithm can accurately locate the motion vectors with lower computational overhead as compared with previous works.

2. Markov model based block motion estimation

The Markov model of the proposed algorithm is illustrated in Fig. 1. There are three states in this Markov model, including C_{init}, C_d, and C_{end}. Each edge represents a transition to the next state. The transition probability of the next state depends only upon the current state and not upon any previous states. In other words, the probability that the matching process in one state will change to a different state depends only on the two states (and not on the time, earlier states, or other factors). Thus, we have a random sequence in which the dependency extends backwards one unit in time. That is, the way in which the entire past history affects the future of the matching process is completely summarized in the current state of the matching process. Expressed analytically the Markov property of the proposed algorithm may be written as

$$P[X(t_{n+1}) = x_{n+1} | X(t_n) = x_x, X(t_{n-1}) = x_{n-1}, \dots, X(t_1) = x_1] = P[X(t_{n+1}) = x_{n+1} | X(t_n) = x_n],$$

$$P, \text{ shrink the search window, } d \rightarrow d/2$$

$$C_{and} \longrightarrow C_{and} \longrightarrow Motion$$

$$Vector$$

Fig. 1. The proposed three-states Markov model for motion estimation.

Vector

where $t_1 \le t_2 \le \dots \le t_n \le t_{n+1}$ and x_i is included in some discrete state space and $\{X_t\}_{t=0,1,2,\dots}$ is a first order finite state Markov chain.

To improve the drawback of the fast BMA algorithm described in Section 1, we adopt this three-states Markov model herein. First, a rough motion vector is got in the state C_{init} . In the C_d state, mechanism that accepts a suboptimal solution is provided here to avoid getting into local minima. Finally precise fine-tuning is done to achieve the global minimum in the C_{end} state.

The following are the detailed descriptions of each state mapping into the proposed block-matching algorithm.

2.1. State C_{init}

In C_{init} , the starting search points consist of the origin and its surrounding eight points, i.e., the search window of 3×3 . Meanwhile we consider the highly temporal and spatial correlation in image sequences. The motion vector of each block is highly correlated with the motion vectors of its neighboring blocks no matter in temporal or spatial way. By exploiting the spatiotemporal relationship, we can reduce a considerable time in finding a rough initial motion vector via getting enough information from the motion vectors of the neighboring blocks. Hence this algorithm not only uses the aforementioned 3×3 search points centered around the origin, but also utilizes three search points located by the motion vectors of the two spatially and one temporally adjacent blocks. All these search points (12 in maximum if not overlapping) can be regarded as a search points pattern denoted by C_{init} .

2.2. State C_d

In the aforementioned fast algorithms, the candidate point with the minimum error in every iterative stage is chosen as the starting point in the next stage, so that it is easily trapped into local minima. To conquer the local minima problem, the suboptimal solution is accepted here in this state with the acceptance probability between 0 and 1. That means even getting a suboptimal solution, the current state still gets the chance to search outward from the point with this solution. The search points maybe increase slightly though; it is still feasible to get out of the local minimum to enhance the picture quality via a fine scheme. There are many ways to accept a suboptimal solution, e.g., simulated annealing [9,10]. In this paper, we utilized fuzzy-reasoning as a mechanism to provide the acceptance probability of suboptimal solutions, as described in the next section.

2.3. State Cend

When the search reaches near the probably global optimum, we need to give it a smaller search window to let the search be precisely converged. Thus, the 3×3 search points pattern as in Fig. 7 is applied in state C_{end} .

Fuzzy-reasoning is then used to decide whether the matching process shall proceed to the next state based on the transition probability of the state in the Markov chain or not.

Hence, in this paper we propose a fast and robust block motion estimation algorithm that combines the three-states Markov model with a fuzzy-reasoning algorithm [11] to avoid the local optimum dilemma in the aforementioned algorithms.

3. Markov model fuzzy-reasoning based algorithm for block motion estimation

In this section, we describe a fuzzy-reasoning based algorithm for fast and robust motion estimation. Before addressing the proposed algorithm in detail, we briefly review the fuzzy-reasoning.

3.1. Fuzzy-reasoning

Considering about the object function of multi-extreme value distribution, we can solve the local minimum problem by using the concept of fuzzy logic. In this section, we first discuss the characteristic of fuzzy set. Second, we describe how to apply fuzzy-reasoning to block motion estimation.

3.2. Fuzzy set

For a fuzzy set A, membership $\mu_A(x)$ can be defined as

 $\mu_A: X \to [0,1].$

 $\mu_A(x)$ can be represented by real number ranging from zero to unity (including 0 and 1). While element x does not completely belong to A and not completely unbelong to A, the membership $\mu_A(x)$ with respect to this element will be a decimal fraction between 0 and 1. The bigger $\mu_A(x)$ is, the higher degree x belongs to A and vice versa. Hence membership extends the range of crisp set from $\{0,1\}$ to [0,1]. In other words, membership represents the degree of certain element belongs to the fuzzy set. It is subjective and variant depending on human perception; however, we still can describe the egularity of taking value by objective statistical method. One of the often-used memberships, S-shape function, depicted in Fig. 2 is formulated as Eq. (2):

$$S(x, a, b) = \begin{cases} 0 & \text{for } x < a, \\ 2\left(\frac{x-a}{c-a}\right)^2 & \text{for } a \le x < b, \\ 1 - 2\left(\frac{x-c}{c-a}\right)^2 & \text{for } b \le x < c, \\ 1 & \text{for } x \ge c. \end{cases}$$
(2)

Before discussing fuzzy number, we define normalization and convex set of a fuzzy set.

3.3. Normalization

If there exists a fuzzy set A with property of, $\max_{x \in X} \mu_A(x) = 1$ then A is normalized.

3.4. Convex set

For a fuzzy set A consisting of all real numbers, if there exists $\mu_A(y) \ge \mu_x \land \mu_A(Z)$ for any real numbers, $x \le y \le z$ then we call A a convex set. Where \land is to take the minimum from two operands.

The purpose here is to make each value of the membership function in the domain corresponds to one and only one grade in the range. Then we can have one and the only one acceptance(transition) probability value after getting the next state solution (MSE).

Take an example depicted in Fig. 3, the top curve represents a triangular fuzzy number; the middle curve shows a trapezoidal fuzzy number; the bottom curve illustrates a bell-shaped fuzzy number. These three functions, known as membership functions, are all convex (the grade starts at zero, rises to a maximum, and then declines to zero again as the domain increases). However, some fuzzy numbers have concave, irregular, or even chaotic membership functions. There is no restriction on the shape of the membership curve, as long as each value in the domain corresponds to one and only one grade in the range, and the grade is never less than 0 nor more than 1000.

3.5. Fuzzy number

A continuous convex fuzzy set consisting of all real numbers normalized is called fuzzy number.



Fig. 2. Plot of S-shape membership.



Fig. 3. Illustration of convex membership curve for fuzzy set.

That is, a fuzzy number is a quantity whose value is imprecise, rather than exact as is the case with "ordinary" (single-valued) numbers. Any fuzzy number can be thought of as a function whose domain is a specified set (usually the set of real numbers, and whose range is the span of non-negative real numbers between, and including, 0 and 1. Each numerical value in the domain is assigned a specific "grade of membership" where 0 represents the smallest possible grade, and 1 is the largest possible grade.

In many respects, fuzzy numbers depict the physical world more realistically than single-valued numbers. Suppose, for example, that you are driving along a highway where the speed limit is 55 miles an hour (mph). You try to hold your speed at exactly 55 mph, but your car lacks "cruise control," so your speed varies from moment to moment. If you graph your instantaneous speed over a period of several minutes and then plot the result in rectangular coordinates, you will get a function that looks like one of the curves shown in Fig. 3. Note that the grade of membership is not normalized for fuzzy number.

 α -cut is a method that transforms the fuzzy number set into crisp set. According to Eq. (3), we implement the transformation from fuzzy set to crisp set as shown in Fig. 4.

$$A_{\alpha} = \{ x | \mu_A(x) \ge \alpha \}, \quad 0 \le \alpha \le 1.$$
(3)

3.6. The proposed three-states Markov model fuzzy-reasoning based algorithm

The principle of fuzzy-reasoning based block matching algorithm is to apply a membership between 0 and 1 to sub-optimal solutions, representing the degree of acceptability of the current state solution, then



Fig. 4. Transform fuzzy set into crisp set using α -cut with A_{α -cut} = [a_1, a_2].

using α -cut to transform the fuzzy number set into a crisp set. If the current state solution belongs to the worse solution of the crisp set, we keep on searching. There are two parameters α and membership we have to determine. We found that S-shape membership is the most suitable membership for the block-matching algorithm, though Eq. (2) needs some modification as follows:

$$S(x, a, b) = \begin{cases} 1 & \text{for } x < a, \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2 & \text{for } a \le x < \frac{a+b}{2}, \\ 2\left(\frac{x-b}{b-a}\right)^2 & \text{for } \frac{a+b}{2} \le x < b, \\ 0 & \text{for } x \ge b. \end{cases}$$
(4)

In Fig. 5, the value of S(x,a,b) is corresponding to x = (a + b)/2, *a* is the block distortion measure (BDM) of the current state solution (the optimal solution till now), *x* represents the BDM of the next state solution (the optimal solution found in the neighborhood of the current state solution), and *b* stands for a factor multiplied by $a(b = a \times \text{factor})$. The reason why BDM be taken to define the fuzzy number set is that BDM is the criteria for judging the performance of the search result, from which, fuzzy-reasoning can deduce if the search shall keep on from the suboptimal point or not.

The algorithm will keep on searching when the next state solution is less than the current state solution. On the contrary, as long as we give an appropriate factor, if the BDM of the next state solution is larger than that of the current state solution, we will apply these solutions to S-shape membership, and use α -cut to decide whether we shall keep on searching or not. Since the fast search block motion estimation algorithm is an iterative two-dimensional searching process, to follow this approach we consider a new algorithm that incorporates the aforementioned fuzzy-reasoning technique.

To apply the fuzzy-reasoning in this problem, the state will denote the search points pattern and the solution associated with the state will correspond to the minimum BDM based on these search points [9,10]. In every search stage, we consider two search points patterns, the current one and the next one, and invoke the fuzzy-reasoning for deciding the appropriate one. More specifically, suppose that M_1 and M_2 stand for the minimum BDM based on the current and the next search points patterns, respectively. If $M_2 < M_1$, then the search will proceed based on the next search points pattern. Otherwise, the search will either jump to the final fine search or stay in the current one to further search decided by Eqs. (3) and (4).



Fig. 5. Plot of the modified S-shape membership with $A_{\alpha-\text{cut}} = [0, (a+b)/2]$.

137

Furthermore, two strategies are employed to further enhance the performance of the proposed algorithm. In view of the fact that most video sequences only involve gentle movements, there exist high correlations among the motion vectors of the temporally and spatially adjacent blocks. The first scheme is then to appropriately choose the initial search points by exploiting these intimate relationships. Similar ideas have also been reported in previous literature [12–14]. Here, we extend their ideas by choosing not only the adjacent neighboring points as the initial search points but also the points located by the motion vectors of the temporally and the spatially adjacent blocks. More specifically, we adopt the starting search points which include C'_0, C'_{i-1}, C'_{j-1} , and C'_0^{-1} as shown in Fig. 6, where C'_0 denotes the search point pattern that includes the center of the search window of the present block and its eight neighboring points, C'_{i-1} and C'_{j-1} denote the search points based on the motion vectors of the two spatially adjacent blocks, respectively, and C_0^{t-1} represents that of the collocated block of the previous frame. In light of the center-biased characteristic of the video sequences as addressed above, most of these search points are in common and therefore do not incur too many computations. In addition, to reduce the computational complexity while still covering enough search area, the second scheme uses the alternate search points patterns as shown in Fig. 7 for successive blocks.

The proposed algorithm begins with the determination of the solution of the initial state, M_1 , as the minimum BDM based on the aforementioned C_{init} search points pattern. Also, we consider the solution of the next state, M_2 , as

$$M_2 = \text{median}\{b_{i-1}^t, b_{j-1}^t, b_0^{t-1}\},\tag{5}$$

where median{} stands for the median value of the variables inside the bracket, and b_{i-1}^t, b_{j-1}^t , and b_0^{t-1} denote the minimum BDM of the two spatially and one temporally adjacent blocks based on their respective optimum motion vectors, which have already been determined.

If $M_2 < M_1$, we expect that the present block involves fast movement and the next search pattern should be closer to the boundary of the search window. As such, we use a larger search pattern C_d , where d denotes the corresponding search size, and determines the new M_2 based on the minimum BDM of these new search points pattern. If the new M_2 is still smaller than M_1 , we set this M_2 as the new M_1 and begin a new search with the center at the winning point with a search points pattern of shrinking size, $C_{d/2}$, and repeat the above steps. On the other hand, if $M_2 > M_1$, we use the fuzzy-reasoning to determine whether we shall proceed with the same steps as the above or we shall go directly to the final fine search. The above procedures repeat until



Fig. 6. The starting search points patterns $C_0^t, C_{i-1}^t, C_{j-1}^t$, and C_0^{t-1} , where the short arrows denote the corresponding motion vectors.



Fig. 7. $\{C_d\}$ and C_{end} for alternate search points patterns (A) and (B) with w = 7.

the minimum search window is attained. The final fine search pattern, C_{end} , is to check the optimum motion vector determined so far along with its eight neighboring points to more accurately locate the best match as shown in Fig. 7. Based on the discussion above, the overall procedure for the proposed algorithm can be summarized as follows.

 C_{init} state: Let $M_1 = \text{minimum BDM}$ based on the search points pattern C'_0, C'_{l-1}, C'_{l-1} , and $C'^{l-1}, M_2 = \text{median}\{b'_{l-1}, b'_{l-1}, b^{l-1}_0\}$, search window size $= (2w + 1) \times (2w + 1), l = 1$. C_d state: Do FR $(M_1, M_2, \text{ACCEPT})$. If ACCEPT = 0, go to state C_{end} ; otherwise, $M_1 \leftarrow \min\{M_1, M_2\}$, $d = \frac{w+1}{2^l}, M_2 = \min$ minimum BDM based on the search points pattern C_d with the center being the winning point in the previous iteration. If d > 1, then l = l + 1 and repeat state C_d . C_{end} state: Perform the final fine search points pattern C_{end} .

PROCEDURE FR(M1, M2, ACCEPT)

{If $M_2 < M_1$, ACCEPT = 1; else if $M_2 > M_1$ with the acceptance probability $\mu_A(M_2) > \alpha$ given in Eqs. (3) and (4), ACCEPT = 1; else ACCEPT = 0.}

The search points pattern will alternate between patterns (A) and (B), as shown in Fig. 7 with w = 7, which stands for search window size of 15×15 .

An example to illustrate the above algorithm is shown in Fig. 8. In Fig. 8, numbers mean the orders of the search steps. Search points denoted by No.1 represent that C_{init} search point pattern including $\{C_0^t, C_{i-1}^t, C_{0-1}^t, C_0^{t-1}\}$ is applied in step one. Nos. 2, 3, 4 stands for applying C_4 , C_2 , and C_1 in step 2, 3, and 4, respectively. No. 5 means that the final fine search point pattern C_{end} is applied. Hence, each arrow is on behalf of the motion vector estimated in each step. The fuzzy-reasoning mechanism will be activated between each step if M_2 is larger than M_1 .

4. Experimental results

Simulations are conducted in this section to evaluate the proposed algorithm. Two sets of video sequences are considered. The first set, including the "Claire," "Miss America," and "Salesman" video sequences, main-



Fig. 8. An example with motion vector (7, 6) to demonstrate the proposed algorithm.

ly consists of low activity blocks; whereas the second set, including the "Flower," "Football," and "Table tennis" video sequences, consists of lots of high activity blocks. The computationally efficient mean absolute difference (MAD) given by

$$MAD_{d}(x) = \frac{1}{|\beta|} \sum_{i \in \beta} |f_{k}(x+i) - f_{k-1}(x+i+d)|$$
(6)

is used as the BDM, where $f_k(i)$ denotes the density of pixel at $i = (i_1, i_2)$ of the kth frame, β is an $M \times N$ block, and $|\beta|$ denotes the number of points in β . For comparison, six existing algorithms, including the FS, TSS, NTSS, FSS, and DS along with the proposed one (with the parameters factor = 1.3, $\alpha = 0.5$), are carried out. Two standard criteria: the average mean squares error per pixel (MSE/pixel) and the average search points per block are utilized to evaluate the effectiveness of these algorithms. The former criterion is to measure the reconstructed image quality, whereas the latter the computational complexity.

The resulting average MSE/pixel and the average search points per block for the first 90 frames of the test sequences using these algorithms based on video standards with M = N = 16 and w = 15 are listed in Tables 1 and 2, respectively. Table 3 lists the average number of carrying out Eq. (4) per block for the test sequences.

From Table 1, it is seen that the proposed algorithm outperforms the others by providing a smaller MSE/ pixel (except for the FS), whereas we can observe from Table 2 that the new algorithm also calls for the lowest computational complexity. Note that we can regard FS as the "gold standard" which guarantees the global optimum in block matching under the same conditions, ex. the same block size and search range. In addition, the simulation of the deviation of the motion vectors from the ones found by FS is ignored due to its strong correlation with the criterion MSE/pel.

The improvements of the proposed algorithm are more obvious in fast-motion sequences such as "Flower," "Football," and "Table Tennis" in terms of MSE/pel and search points/block. It is because the proposed algorithm not only using the fuzzy concept to jump out of the local optimum dilemma but also utilized the

Table 1 Comparison of MSE/pixel for various algorithms on the first 90 frames of the test sequences

	Claire	Miss America	Salesman	Flower	Football	Table Tennis
FS	9.14	10.11	27.60	277.00	384.88	184.86
TSS	9.35	10.57	28.29	320.33	416.43	240.07
NTSS	9.31	10.24	27.92	285.01	412.54	217.46
FSS	9.31	10.50	28.16	299.69	428.89	213.28
DS	9.29	10.26	28.13	287.03	433.32	205.94
(SA) Simulated annealing-based	9.26	10.22	27.77	279.80	411.00	196.35
(FR) Fuzzy reasoning-based	9.29	10.20	27.75	279.71	411.95	196.78

	-	-	-			
	Claire	Miss America	Salesman	Flower	Foot ball	Table Tennis
FS	204.28	204.28	204.28	202.05	202.05	202.05
TSS	23.28	23.44	23.23	23.25	23.09	23.32
NTSS	20.28	21.78	16.85	21.58	20.56	21.57
FSS	17.59	18.83	16.24	18.90	18.04	19.03
DS	14.99	16.60	12.92	17.02	16.06	16.87
(SA) Simulated annealing-based	14.20	16.53	11.70	15.81	14.53	15.97
(FR) Fuzzy reasoning-based	14.95	16.59	11.49	15.42	14.33	15.74

 Table 2

 Comparison of average number of search points per block for various algorithms on the first 90 frames of the test sequences

Table 3

Comparison of the average number of performing Eq. (4) per block on the first 90 frames of the test sequences

	Claire	Miss America	Salesman	Flower	Football	Table Tennis
(SA) Simulated annealing-based	0.99	1.11	1.09	1.12	1.10	1.11
(FR) Fuzzy reasoning-based	1.12	1.34	1.03	1.06	1.08	1.15

spatial and temporal correlations. The insignificant improvements in quasi-still or slow motion sequences such as "Claire," "Miss America," and "Salesman" is because it is not necessary to utilize spatial and temporal correlations in these kinds of sequences. The center-biased characteristic of the Diamond Search and Four-Step Search can handle them well enough. As for fast-motion sequences, DS and FSS do not utilize the spatial and temporal correlation to correctly allocate the initial search point and speed up the search. Besides, DS and FSS do not handle the local optimum case, from which fast-motion sequences usually suffer. Hence, if the algorithm performs well in these sequences, we could say the proposed technique can effectively avoid the local minima.

The small MSE and low complexity of the proposed algorithm are due to the fuzzy-reasoning technique as well as the full exploitation of the temporal and spatial relationships among the motion vectors field. The incorporation of the fuzzy-reasoning scheme also explains that the proposed algorithm works in particular well for high activity video sequences, in which the previous fast algorithms are easily trapped into local minima in the search process.

We can find from Table 3 that values for all of the video sequences are about 1, which is negligible comparing with 256 subtractions and 255 additions needed for a search point. Thus, the overhead induced by performing Eq. (4) is negligible.

Three states are adopted in the proposed Markov model based fast block-matching algorithm. State C_{init} equals $\{C_0^t, C_{i-1}^t, C_{j-1}^t, C_0^{t-1}\}$, which theoretically consists of 12 points, whereas the experimental results in Table 4 shows that state C_{init} actually only accounts for slightly more than 9 points. Because the most blocks belong to still or slow motion type, most of the search points in $\{C_{i-1}^t, C_{j-1}^t, C_0^{t-1}\}$ overlap that in C_0^t . There are only a few blocks belong to fast motion type. At the same time, the search points of C_{init} increase in proportion to the picture complexity. State C_d composed of $\{C_4, C_2, C_1\}$ is allocated to search outward for the probable blocks with local minima. Its execution count depends on the characteristic of the block. In state C_d , the search points decrease with the search window shrinks, which means $C_4 > C_2 > C_1$ as

Table 4 The average number of search points of different search point patterns

•						
	$C_{ m init}$	C_4	C_2	C_1	$C_{\rm end}$	Points
Claire	9.06	2.07	0.85	0.72	2.26	14.95
Miss America	9.50	1.98	0.99	0.94	3.17	16.59
Salesman	8.60	1.90	0.10	0.09	0.78	11.49
Flower	9.23	1.79	0.09	0.08	4.16	15.42
Football	9.22	1.88	0.27	0.26	2.61	14.33
Table tennis	9.53	1.78	0.39	0.38	3.65	15.74

shown in Table 4. It conforms to that there is only a small number of the blocks with high motion quantity, for which search will not stop early as in slow motion ones. To cover more area under limited search points, alternate search is adopted in state C_{d} . Because the incomplete search, fune-tune will be done in state C_{end} to converge on global minimum in expectation. The increase points in state C_{end} are also tolerable. Combining the above three states, we can correctly classify each block to give it appropriate search points to find the best match.

The experimental results of three-state Markov model incorporated with simulated annealing-based (SA) approach [9,10] instead of fuzzy reasoning-based (FR) one are also listed in Tables 1–3 for comparison. MSE/pel of FR is higher than SA in sequences "Claire," "Football," "Table Tennis" as shown in Table 1, while Average search points is lower in sequences "Miss America," "Salesman," "Flower," "Football," and "Table Tennis" as shown in Table 2. The acceptance probability of SA is as follows:

$$P = e^{\frac{-(E_2 - E_1)}{kT}},\tag{7}$$

where E_1 and E_2 are the associated energy (BDM) of the current and next states, respectively, T is a control parameter (called "temperature"), and k is the Boltzmann constant. The average number of performing Eq. (4) for FR is less than Eq. (7) for SA in sequence "Salesman," "Flower," and "Football." The overhead induced by performing Eq. (7) is also negligible.

5. Conclusions

In this paper, we describe a Markov model based fast block motion estimation algorithm that is also robust against local minima. The spirit of this new algorithm is adopting a three-states Markov model, which gets the initial rough motion vector in state C_{init} via utilizing high correlations among the motion vectors of the temporally and spatially adjacent blocks as well as the surrounding points centered at origin point, applying fuzzy-reasoning to provide an acceptance probability for suboptimal solution in state C_{d} , and precisely fine-tuning to converge toward global optimum in state C_{end} . The alternate search technique for successive blocks is also applied to maintain the same search window size without incurring extra overhead.

The major difference between the proposed algorithm and conventional multi-step ones is the employment of the fuzzy-reasoning concept to keep on search from the acceptable suboptimal search point. The conventional multi-step methods do not accept any suboptimal solution in each step, so they won't regard the suboptimal search point as the winning point from which the search proceeds.

The characteristic of this proposed technique is its simplicity yet effective compared with the existing popular algorithms. The results show the reduction of the computation complexity and the increase in the picture quality in terms of search points/block and MSE/pel, respectively, as shown in Tables 1 and 2.

Note that although it is necessary to calculate the acceptance probability $\mu_A(M_2)$ to decide if the suboptimal solution is acceptable, the overhead of applying fuzzy reasoning is about one time to calculate the Eq. (4), which is negligible and represents the feasibility of the fuzzy-based algorithm. Because this algorithm attains superior performance with lower computations compared with previous works, it offers an appealing alternative to block motion estimation, which is compatible with those adopted in all H.26x and MPEG-x series.

References

- Yun Q. Shi, Huifang Sun, Image and Video Compression for Multimedia Engineering Fundamentals, Algorithms, and Standards, CRC Press, New York, NY, 1999.
- [2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in: Proceedings of the NTC 91, New Orleans, LA, Nov./Dec. 1981, pp. C9.6.1-9.6.5.
- [3] R. Li, B. Zeng, M.L. Liou, A new three-step search algorithm for block motion estimation, IEEE Trans. Circuits Syst. Video Technol. 4 (1994) 438–442.
- [4] L.-M. Po, W.-C. Ma, A novel four-step search algorithm for fast block motion estimation, IEEE Trans. Circuits Syst. Video Technol. 6 (1996) 313–317.

- [5] J.-Y. Tham, S. Ranganath, M. Ranganath, A. Kassim, A novel unrestricted center-biased diamond search algorithm for block motion estimation, IEEE Trans. Circuits Syst. Video Technol. 8 (1998) 369–377.
- [6] J.-S. Jan, W.-H. Fang, M.-Y. Yu, An adaptive flow-based dynamic search algorithm for block motion estimation, in: Proceedings of the IEEE International Symposium Circuits and Systems, Hong Kong, 1997, pp. 2092–2095.
- [7] C.-Y. Tang, Y.-P. Hung, Z. Chen, Robust two-stage approach for image motion estimation, Electron. Lett. 34 (1998) 1091–1093.
- [8] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338-353.
- [9] M.-C. Shie, W.-H. Fang, K.-J. Hung, F. Lai, Fast block motion estimation using adaptive simulated annealing, in: Proceedings of the IEEE Asia-Pacific Symposium Circuits and Systems, Thailand, 1998, pp. 607–610.
- [10] M.-C. Shie, W.-H. Fang, K.-J. Hung, F. Lai, Fast block motion estimation using adaptive simulated annealing, IEICE Trans. Fundam. Electronics Commun. Computer Sci. E83-A (2000) 121–127.
- [11] P.-H. Chen, H.-M. Chen, K.-J Hung, M.-C. Shie, F. Lai, Robust fuzzy-reasoning based algorithm for fast block motion estimation, in: Proceedings of the IEEE International Symposium Intelligent Signal Processing and Communication Systems, 2002, pp. 305–309.
- [12] J. Chalidabhongse, C.-C. Jay Kuo, Fast motion vector estimation using multiresolution-spatio-temporal correlations, IEEE Trans. Circuits Syst. Video Technol. 7 (3) (1997) 477–488.
- [13] A.M. Tourapis, O.C. Au, L. Liou, Predictive motion vector field adaptive search technique (PMVFAST)—enhancing block based motion estimation, Proc. Visual Commun. Image Process. (2001).
- [14] A.M. Tourapis, O.C. Au, Ming L. Liou, Highly efficient predictive zonal algorithms for fast block-matching motion estimation, IEEE Trans. Circuits Syst. Video Technol. 12 (10) (2002) 934–947.

Po-Hung Chen was born in 1974. He received the B.S. and M.S. degrees in electrical engineering from Tatung University, Taiwan, in 1996 and 1998, respectively. He currently is a Ph.D. candidate and member of the Archi Group in electrical engineering of National Taiwan University, under the supervision of Prof. Feipei Lai. His research interests are in video technology and low power design.

Hung-Ming Chen was born in 1971. He received the B.E. degree in computer science and information engineering from Tatung University, Taiwan, in 1993 and the M.E. degree in electrical engineering from National Taiwan University in 1995. He is currently a Ph.D. candidate and member of the Archi Group in electrical engineering of National Taiwan University, under the supervision of Prof. Feipei Lai. In June 1997, he joined Tatung Company as a design engineer specializing in embedded system. He has been promoted to be a Senior Manager of Multimedia Communications Division and Center for Research of Advanced Information Technologies (CRAIT) of Tatung Company. His research interests are in low power, multimedia, software architecture, and human-computer interaction designs for embedded systems.

Kuo-Jui Hung was born in Taiwan, Republic of China, in 1971. He received the B.S. and M.S. degrees in electrical engineering from National Taiwan University of Science and Technology, respectively. He is now with the ASUS Comp., designing the PC motherboards and notebooks.

Wen-Hsien Fang was born in Taipei, Taiwan, ROC, in 1961. He received his B.S. degree from the National Taiwan University in 1983, and his M.S.E. degree and Ph.D. degree from the University of Michigan, Ann Arbor, in 1988 and 1991, respectively, in electrical engineering and computer science. From 1988 to 1991, he was a Reserch Assistant at the University of Michigan. In fall 1991, he joined the faculty of National Taiwan University of Science and Technology, where he currently holds a position as a Professor in the Department of Electronic Engineering. His research interests include fast algorithms for signal processing and their VLSI hardware implementations, statistical signal processing, wireless communication, and video coding.

Mon-Chau Shie He received the B.E. degree in electrical engineering from National Taiwan University, the M.E. degree in electrical engineering from National Taiwan University, and the PH.D. degree in electrical engineering from National Taiwan University. He is currently a assistant professor in the Department of Electronic Engineering of National Taiwan University of Science and Technology. His research interests are in computer video and image processing, operation system and microprocessor system design.

Feipei Lai received the B.E. degree from National Taiwan University in 1980, and M.E. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 1984 and 1987, respectively. He is a professor in the Department of Electrical Engineering and in the Department of Computer Science and Information Engineering at National Taiwan University. He was a visiting professor in the Department of Computer Science and Engineering at the University of Minnesota, Minneapolis, USA. He was also a guest Professor at University of Dortmund, German and a visiting senior computer system engineer in the Center for Supercomputing Research and Development at the University of Illinois at Urbana-Champaign. Dr. Lai holds four Taiwan Patents and two USA patents currently. He served as a consultant at ERSO, ITRI during 1988 and at Faraday Technology Corp. from 8/94 to 7/95. His current research interests are high performance microprocessor chip design, computer architecture, optimizing compiler, VLSI design. Prof. Lai is one of the founders of the Institute of Information & Computing Machinery. He is also a member of Phi Kappa Phi, Phi Tau Phi, ACM, The Chinese Institute of Engineers, and The Institute of Electronics, Information and Communication Engineering. He received Acer awards five times in 1989, 1991, 1992, 1993 and 1995 and The Taiwan Fuji Xerox Research award in 1991. Dr. Lai is a Senior member of IEEE and included in "Who's Who in Science and Engineering" and "Who's Who in the World."