Fast Block Motion Estimation Using Adaptive Simulated Annealing *

Mon–Chau Shie[1], Wen–Hsien Fang[2†], Kuo–Jui Hung[2] and Feipei Lai[1]
[1]Dept. of Electrical Engineering & Dept. of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
[2] Dept. of Electronic Engineering, National Taiwan University of Science and Technology
Taipei, Taiwan
†Tel:886-2-27376412 Fax:886-2-27376424 Email:whf@et.ntust.edu.tw

Abstract

This paper presents a simulated annealing (SA)-based algorithm for fast adaptive block motion estimation. The new algorithm begins with a set of checking points by exploiting the high correlations among the motion vectors of the temporally and spatially adjacent blocks. The SA algorithm is then employed in the decision process to determine the next searching patterns. This new scheme provides a possibility of being able to move out of local minima so that the search process is less susceptible to such a dilemma. Additionally, an alternating search strategy is also addressed to visit more points without increasing computations. Simulation results show that the new algorithm offers superior performance with lower computational complexity compared with previous works.

1. Introduction

Motion estimation underlines the core of motion compensated predictive coding of image sequences. The block matching algorithms (BMA) in particular have received a great deal of attention and been adopted by various standards such as H.261, MPEG 1-2, and etc. [1].

The most straightforward BMA is the full search (FS) algorithm, which searches exhaustively over all allowable displaced points to locate the best match and thus requires enormous amount of computations. To mitigate this, various attempts have been made to reduce the number of search points while retaining acceptable image quality. For example, the popular three-step search algorithm (TSS) iteratively checks the surrounding nine points of the minimum with a diminishing window size [2]. Several variants of the TSS such as the new three-step search (NTSS) algorithm [3] and the four-step search (FSS) algorithm [4] were addressed to determine a more precise motion vector 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]. Consequently, despite their efficiency, the search is easily trapped into local minima, especially for high-activity image sequences.

Some approaches such as [5], which used the motion "flow" concept in determining the motion vector, have been addressed to refrain the search from being easily trapped into local minima. In this paper we propose a new algorithm by using the SA together with the local characteristics of the image. The new algorithm begins with a set of checking points by exploiting the high correlations among the motion vectors of the temporally and spatially adjacent blocks. We then invoke the SA algorithm in deciding the next search patterns. In contrast to previous works which are "greedy" in the search process, the new scheme possesses a mechanism of jumping out of local minima and thus is less susceptible to this dilemma. Furthermore, alternating checking point patterns for adjacent blocks are used so that the search can visit more points without increasing computations. As a consequence, the motion vectors can be accurately determined with lower computational overhead as compared with previous works. The furnished simulations justify this new algorithm.

2. Simulated Annealing-Based Algorithm for Motion Estimation

In this section, we describe an SA-based algorithm for fast adaptive motion estimation. Before address-

*This work was supported by National Science Council of R.O.C. under contracts NSC 87-2213-E-011-014 and 87-2218-E-002-020.

ing the proposed algorithm, we briefly review the SA.

2.1 Simulated Annealing

The SA is a well-known effective technique for solving iterative optimization problems and has been successfully applied to the contexts of circuit design, and vector quantization [6]. In such problems, the traditional iterative search process is easily trapped into local minima on the way to find the global minimum. In contrast, the SA allows the possibility of escaping from local minima by providing a mechanism of uphill move according to a stochastic decision rule. More specifically, the SA is constituted by the following two rules in the transition of "states":

1. A change of state which induces a reduction of the associated energy is always allowed.

2. If a change of state induces an increase of the associated energy, the change is then accepted with the probability

$$P = e^{\frac{-(E_2 - E_1)}{kT}} \tag{1}$$

where E_1 and E_2 are the associated energy of the present and next states, respectively, T is a control parameter (called "temperature"), and k is the Boltzmann constant. As a consequence of (1), we can note that the scheme always takes a downhill step (reduction of energy) in a transition of states, and sometimes an uphill step (increase of energy) with probability P, which decreases as the energy difference increases. It is this uphill climbing technique which enables the search process to be more robust against local minima. Also, the temperature T decreases as iterations proceed. This reflects the fact that it is less likely to be trapped into local minima as the search gets closer to the global minimum. A simple mechanism to adjust T is to reduce it by a ratio in each iteration.

2.2 Proposed Algorithm

The proposed algorithm begins with an appropriate choice of checking points. Due to the fact that most image sequences only involve gentle movements, there exist high correlations among the motion vectors of the temporally and spatially adjacent blocks. As such, we adopt the center of the search window of the present block and the motion vectors of the spatially and temporally adjacent blocks as well as their neighboring points as our starting points to locate the best match. More specifically, these starting candidate points include C_0^t, C_{i-1}^t, C_{j-1}^t, and C_0^{t-1}, where C_0^t denotes the checking point pattern that includes the center located at the present block and its neighboring points, whereas $C_{i-1}^t, and\ C_{j-1}^t$, denote the

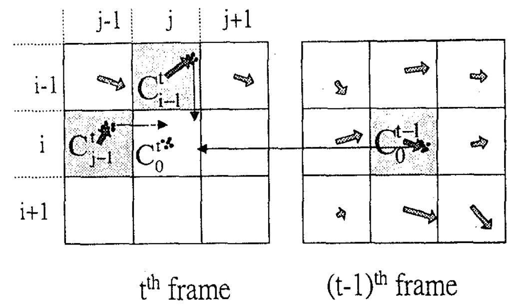

Figure 1: The starting checking point patterns of $C_0^t, C_{i-1}^t, C_{j-1}^t$, and C_0^{t-1}.

checking point patterns based on the motion vectors and their neighboring points of the two spatially adjacent blocks and C_0^{t-1} those of the corresponding block of the previous frame as shown in Fig. 1. Due to the fact that the center-biased characteristic of the image sequences as addressed above, most of these checking points are in common and therefore do not incur too many computations.

In order to avoid the search to be trapped into local minima, we invoke the SA algorithm in the decision of the next search patterns. To apply the SA algorithm in this problem, the state denotes the checking point pattern and the energy associated with the state corresponds to the minimum BDM based on these checking points. Also, we adopt a normalized T as $T = \sqrt{E_1}$ so that the same probability measure (1) can be used for every block.

We start the SA by choosing the energy of the initial state as the minimum BDM, E_1, based on the aforementioned starting checking points and the energy of the next state, E_2, as

$$E_2 = \min\{b_{i-1}^t, b_{j-1}^t, b_0^{t-1}\} \tag{2}$$

where $b_{i-1}^t, b_{j-1}^t, b_0^{t-1}$ denote the minimum BDM of the two spatially and temporally adjacent blocks based on the optimum motion vectors, respectively, and have been previously determined.

If $E_2 < E_1$, we expect that the present block involves fast movement and the next search pattern should be closer to the boundary of the search window. We then use a larger search pattern C_d, where d denotes the corresponding search size, and determines the minimum BDM based on these new checking points as the new E_2. If the new E_2 is still smaller than E_1, we shrink the size of the search pattern as $C_{\frac{d}{2}}$ and repeat the above steps. On the other hand, if $E_2 > E_1$, we use the SA algorithm to determine whether we should proceed the same steps as the above or we should directly go to the final fine search. Such a procedure repeats until the minimum search window is attained. The final fine search is to check the optimum motion vector determined so far along with its nine neighboring points, $\mathcal{C}_{end}$, as shown in Figs. 2 and 3, to locate more accurately the best match.

In addition, in order to reduce computational overhead while still visiting enough checking points, we use alternating search technique between two adjacent blocks, as shown in Figs. 2 and 3, respectively, with $w = 7$. The overall procedures for the proposed algorithm can be summarized as (the mean absolute difference (MAD) is used as the BDM):

1. (Initialization) Let E_1= minimum MAD based on the checking point patterns of $\mathcal{C}_0^t$, $\mathcal{C}_{i-1}^t$, $\mathcal{C}_{j-1}^t$, and $\mathcal{C}_0^{t-1}$, E_2= minimum of $\{b_{i-1}^t, b_{j-1}^t, b_0^{t-1}\}$, the search window size=$(2w+1)\times(2w+1)$, $i = 1$.

2. Do $\mathbf{SA}(E_1, E_2, T,$ACCEPT). If ACCEPT=0, go to Step 3; Otherwise, $d = \log(\frac{w+1}{2^i})$, E_2= minimum based on the checking point patterns $\{C_d\}$, $T = T \times \beta$ (β is a reduction ratio), $i = i + 1$, repeat Step 2.

3. Perform the final checking point pattern $\mathcal{C}_{end}$.

4. Switch the search patterns for the next block and go to Step 1.

where

Procedure $\mathbf{SA}(E_1, E_2, T,$ ACCEPT)
If $E_2 < E1$, ACCEPT=1; Otherwise,
ACCEPT=1 with probability P as given in (1).

3. Simulation Results

In this section, some simulation results are provided to verify the proposed algorithm. The test image sequences include the "Salesman", "Flower", "Table tennis", and "Football" image sequences. Four similar algorithms, including the FS, TSS, NTSS, and FSS, along with the proposed algorithm (with the parameters $\beta = 0.8$ and $k = 0.1$), have been carried out for comparison.

The resulting average mean squares error per pixel (MSE/pixel) and the average search points per block for the first 90 frames of the test image sequences using these algorithms based on H.261 with $M = N = 16$ and $w = 7$ are listed in Table 1. As a vivid illustration, the MSE/pixel vs. the frame number for the test image sequences "Table tennis" and "Football" are also shown in Figs. 4 and 5, respectively (for clarity, only the results from the 21^{st} to the 60^{th} frames are shown). From Table 1, we can note that the proposed algorithm outperforms these four algorithms by providing a smaller MSE/pixel (except the FS) with even lower computational complexity. The small MSE's of the proposed algorithm is due to the full exploitation of the temporal and spatial relationship among the motion fields, and the SA algorithm. The latter also explains that the proposed algorithm works in particular well for high-activity image sequences such as the "Flower" and "Football" sequences, for which the search using the previous fast algorithms is easily trapped into local minima.

4. Conclusions

In this paper, we describe a new BMA for fast motion estimation. The algorithm uses high correlations among the motion fields of the spatially and temporally adjacent blocks, the SA in the transition of search patterns, and the alternating search technique for adjacent blocks, to attain superior performance with lower computations compared to previous approaches. Consequently, it offers an appealing alternative for the BMA for motion estimation.

References

[1] A. M. Tekalp, *Digital Video Processing,* Prentice-Hall, Englewood Cliffs, NJ, 1995.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Proc. NTC 81*, pp. C9.6.1-9.6.5, New Orleans, LA, Nov./Dec. 1981.

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

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

[5] Jai–Shin Jan, Wen–Hsien Fang and Ming–Yi Yu "An adaptive flow-based dynamic search algorithm for block motion estimation," *in Proc. Int'l Symp. Circuits and Sys.*, pp. 2092-2095, Hong Kong, June 1997.

[6] P.J.M. Van Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications,* Kluwer Academic Publishers, Dordrecht, Holland, 1987.

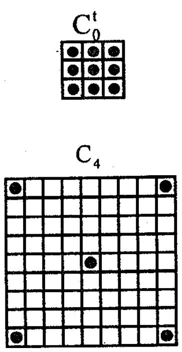

Figure 2: The checking point patterns of $C_0^t, C_{i-1}^t, C_{j-1}^t, C_0^{t-1}, C_d$, and C_{end} used in this paper with $w = 7$

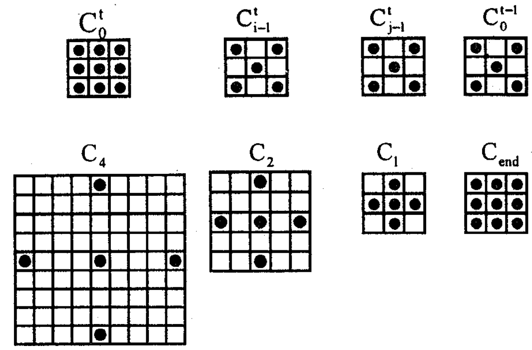

Figure 3: The alternating checking point patterns of Fig 2.

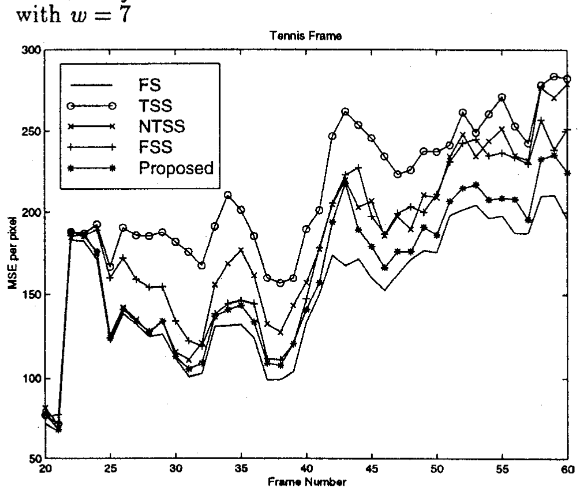

Figure 4: Comparison of MSE/pixel vs. frame number for various algorithms based on the Table Tennis sequence.

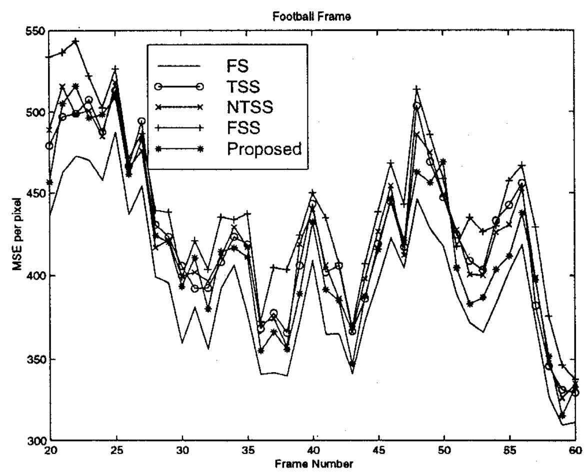

Figure 5: Comparison of MSE/pixel vs. frame number for various algorithms based on the Football sequence.

	Sales.	Flower	T. Tennis	Football
FS	27.29	269.15	140.25	384.88
TSS	28.15	315.49	174.92	416.43
NTSS	27.40	277.64	161.06	412.53
FSS	27.93	292.23	159.69	428.89
Proposed	27.36	272.57	150.35	405.93

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

	Sales.	Flower	T. Tennis	Football
FS	204.28	202.05	202.05	202.05
TSS	23.24	23.25	23.14	23.09
NTSS	16.75	21.49	19.72	20.57
FSS	16.21	18.89	17.94	18.04
proposed	13.19	17.83	16.60	17.62

Table 2: Comparison of average number of search points per block for various algorithms based on the first 90 frames of the test image sequences.