MOTION ESTIMATION METHOD BY EMPLOYING A STOCHASTIC SAMPLING TECHNIQUE

Jinwuk Seok, Pyeong-Soo Mah and Yongki Son

Real-Time Multimedia Research Team Embedded S/W Technology Center
Computer and Software Technology Laboratory Electronics and Telecommunications Research Institute
161 Gajeong-Dong Yuseong-Gu, Daejon, 305-350, South Korea
(Tel: +82-42-860-6365; Fax: +82-42-860-6671; Email:jnwseok@etri.re.kr)

Abstract: In a motion estimation method for use in encoding a moving picture, a full-pixel motion vector is estimated by stochastically sampling a pixel to be processed in a predetermined-sized block of a previous frame or a next frame as a reference frame for each of a plurality of equal-sized blocks in a current frame. Then, a half-pixel motion vector is estimated based on the full-pixel motion vector. Accordingly, both the calculation amount and the calculation time required for the motion estimation are effectively reduced. Further, it can be prevented that the hardware becomes complicated..

Keywords: Motion Estimation, Integer PEL search, Stochastic Sampling

1. Introduction

In various electrical/electronic applications such as a video conference, a high definition television, a receiver for a video on demand (VOD), a personal computer compatible with an MPEG (moving picture experts group) image, a game machine, a receiver for a ground wave digital broadcast, a receiver for a digital satellite broadcast and a cable television, an image signal is transmitted in a digitized form. However, in the process of changing an analog signal into a digitized form, the amount of the data is greatly increased. For the reason, it is inevitable to compress the digitized image data for a successful transmission thereof. To compress the digitized image data, three methods are conventionally used: one is a method for reducing a temporal redundancy of the data; another is a method for reducing a spatial redundancy of the data; and the third is a method for decreasing a statistical redundancy by using characteristic of generated encoded data. Particularly, to reduce the temporal redundancy of the image data, a motion estimation and compensation technique is widely used in the majority of motion picture compression standards, e.g., MPEG and H. 263. In the motion estimation and compensation technique, a portion that is the most similar to a certain part in a current frame (hereinafter, a best matching portion) is searched out from a reference frame, e.g., a previous frame or a next frame. Then, only the difference value between the certain part in the current frame and the best matching portion in the previous or the next reference frame is transmitted. Herein, it is understood that a precise estimation of a motion vector will reduce the difference value that should be transmitted, thereby effectively diminishing the transmission data. However, a considerable amount of calculations and time are required in order to find the best matching portion in the previous or the next frame. In fact, the motion estimation is the most time-consuming stage of all the stages performed for the encoding of the motion picture. Accordingly, there has been continuously made an effort to reduce the processing time for the motion estimation. In the meanwhile, the motion estimation technique adopts two approaches in general; one is a pixel-by-pixel approach and the other is a block-by-block approach. Among these, the block-by-block estimation algorithm has gained a prominent popularity in the art. According to the block-by-block estimation algorithm, a current frame is divided into a plurality of equal-sized search blocks. For a search block in the current frame, a best matching block is searched out from a search region within a previous frame. A motion vector, by definition, represents a displacement between the search block and the best matching block. The motion vector is encoded and then processed. In estimating a degree of matching between two blocks, various matching functions are employed. Among these, a method for obtaining a sum of absolute difference (SAD) between pixels in the two blocks is widely used. A full-region search method is the simplest block-by-block estimation technique, where a SAD value is calculated at all possible search positions within a search region of a certain search block in the current frame. Then, a search position having the smallest SAD value is selected to thereby obtain a corresponding motion vector. Though this full-region search method has a merit in that a search mechanism is sim-

ple and an optimum motion vector can be effectively obtained, a real-time application of this method is very difficult because the amount of calculations are excessively large. As a resolution to the drawbacks of the full-region search method described above, the full-pixel searching operation is the most time-consuming process, occupying about 60% of the whole processing time. Further, the full-pixel searching operation has been included in almost all the moving picture related standards adopted so far. Thus, a more efficient full-pixel search algorithm has been a key to a realization of a real-time encoding of the moving pictures.

Therefore, in this paper, we propose a fast motion estimation algorithm based on stochastic sampling. In the integer PEL search process, the proposed algorithm make the calculation burden of block matching to be decreased about the 1/4 comparing to the burden of calculation in a conventional block matching method.

This paper organized as follows. Section 2 discusses the basic theory of the paper. In section 3, we provide a detailed methodology of the proposed fast ME algorithm. Finally, section 4 and 5 represents the computer simulation of the proposed algorithm and conclusions respectively.

2. Block Based Motion Estimation

Block based motion estimation is among the most popular approaches. Block based motion estimation and compensation has been adopted in the international standards for digital video compression, such as H.261, H.263 and MPEG-1, 2, 4. In the block based motion estimation method, the block-matching has been considered as the most popular method for practical motion estimation due to its lesser hardware complexity [1].

The basic idea of block matching is as follows : Let a pixel (n_1, n_2) in frame k (the present frame) be determined by considering $N_1 \times N_2$ block about left-upper location and searching frame $k-1$ for the location of the best matched block of the same size. The matching of the blocks can be quantified according to various criteria including the maximum cross-correlation, the minimum mean square error (MSE), the minimum mean absolute difference (MAD), and maximum matching pel count. In the these criteria, the sum of the absolute difference (SAD) has been widely used owing to the convenience of a hardware implementation and not using a divide operation.

In the minimum SAD criterion, we evaluate the SAD, defined as follows :

$$SAD(d_1, d_2) = \sum_{(n_1,n_2)\in\mathcal{B}} |s^{n_1}_{n_2,k} - s^{n_1-d_1}_{n_2-d_2,k-1}| \quad (1)$$

where $\mathcal{B}$ denotes an $N_1 \times N_2$ block, for a set of candidate motion vectors (d_1, d_2). The estimate of the motion vector is taken to be the value of (d_1, d_2) which minimizes the SAD. That is,

$$(d_1, d_2)^T = \arg \min_{(d_1,d_2)} SAD(d_1, d_2) \quad (2)$$

Using the block matching criterion, we have to find the best matching block requiringoptimizing the matching criterion over all possible candidate displacement vectors at each pixek (n_1, n_2). This can be accomplished by the so-called "full search", which evaluates the matching criterion for all values of (d_1, d_2) at each pixel, and is extremely time-consuming.

Therefore, the research which decrease the computational burden has been conducted. The first method is "search window method". The search window method is that finding the best matching block process is only conducted in a limited region instead of full frame such that

$$-M_1 \le d_1 \le M_1 \; and \; -M_2 \le d_2 \le M_2 \quad (3)$$

The most widely used searching algorithm is "Hierarchical Motion Estimation Algorithm". The basic idea of hierarchical block matching algorithm is to perform motion estimation at each level successively, starting with the lowest resolution.

The three step search method is the most prominent ME algorithm of the hierarchical method. In the three step method, the best matching block can be found only with 3 steps. At each step, around the standard block denoting (n_1, n_2), there are 9 candidate blocks which is located with same distance and we can find the first level candidate block in the 9 candidate blocks. In the Second step, around the first step best candidate, let the 9 candidate blocks with smaller distance than that of the first step, and the second level best candidate block is found with the second level candidates. Finally, the best matching block is found in the third level candidates with only 1 pixel distance around the second level best candidate.

In attempt to generalize this procedure to other search window parameters, the distance between the candidates and the standard may be decreased in the viewpoint of a logarithm. Consequently, the hierarchical method is so called as "logarithmic search method".

3. The Stochastic Sampling Motion Estimation Method

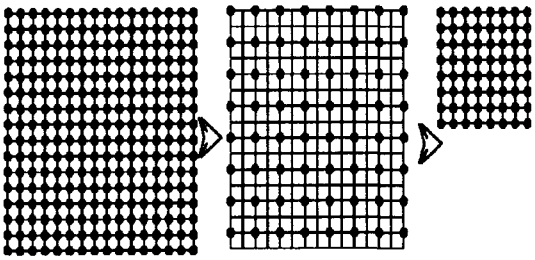

Fig. 1. Overview of Stochastic Sampling in Macro Block

The main idea of the proposed algorithm is that the pixels in the certain block are sampled by using a probability of 1/2 for both X-axis and Y-axis and, then, the sampled pixels are imputed to an estimation unit. Thus, the estimation amount is reduced by 1/4, so that a high-speed integer pel search is realized. If the certain block has a size of 16 × 16 pixels, 8 × 8 pixels are stochastically sampled so that the estimations for the total of 256 pixels are reduced by 1/4 to the estimations for only 64 pixels. As illustrated in Figure 1, estimation points are sampled in a block having a size of 8 × 8 as shown in Fig. 3A with the probability of 1/2 for both the X-axis and the Y-axis. Thus, the number of calculation points for the block of 8 × 8 pixels becomes identical with that for a block having a size of 4 × 4. In order to select or sample the pixels with the probability of 1/2, a stochastic variable which has the probability of 1/2 as a value of 1 is required. To obtain such a stochastic variable, a various method can be adopted. For instance, a pseudo random number generator or a method for storing a random number table in a memory may be utilized. Alternatively, a least significant bit (LSB) of pixels to be processed can be employed, using the characteristic of a binary data.

$$\begin{aligned} R(t+1) &= R(t)+r+c+1 \\ C(t+1) &= C(t)+r+c+1 \end{aligned} \tag{4}$$

In the above equation (4), r represents a probability variable for use in selecting or sampling a pixel on the X-axis, wherein the probabilities of r being "1" and "0" are respectively 1/2; c represents a probability variable for use in sampling a pixel on the Y-axis, wherein the probabilities of c being "1" and "0" are respectively 1/2. $R(t)$ and $C(t)$ represents a rotation coefficient of a currently sampled pixel on the X-axis and on the Y-axis, respectively. Each of $R(t+1)$ and $C(t+1)$ represents a location coefficient of a pixel to be sampled next.

By using the equation (4), it is found that the probability of sampling a pixel on the X-axis which is increased by 1 from $R(t)$ is 1/4; the probability of sampling a pixel on the X-axis which is increased by 2 from $R(t)$ is 1/2; and the probability of sampling a pixel on the X-axis which is increased by 3 from $R(t)$ is 1/4.

$$1x+2y+3z=16 \tag{5}$$

$$(x+y+z)\cdot\binom{x+y+z}{y}\binom{y+z}{z}\left(\frac{1}{4}\right)^x\left(\frac{1}{2}\right)^y\left(\frac{1}{4}\right)^z \tag{6}$$

In the above equation (5) and the formula (6), x represents a number of cases where the sum of the probability variables r and c is 0; y represents a number of cases where the sum of r and c is 1; and z represents a number of cases where the sum of r and c is 2. Under the restriction of the equation (5), the average value (AV) is obtained by using the formula (6). Accordingly, if the present invention is applied to a block having a size of 16 × 16 pixel, a block with a size of 8 × 8 pixel is obtained, thereby reducing the amount of calculations by 1/4.

Further, the present invention has a hill climbing effect which is a representative algorithm for avoiding a local minimum point.As described above, since an average of 8×8 block operations are performed for the 16×16 block, if the present invention is applied to a pixel $s^x_{y,t-1}$ in a previous frame corresponding to a pixel $s^x_{y,t}$ of a current coordinate (x,y), the SAD value is obtained by using the following equation :

$$\sum_x^N\sum_y^N\left|\left(s^x_{y,t}-s^x_{y,t-1}\right)\cdot I(x,y)\right| = \frac{1}{4}\sum_x^N\sum_y^N\left|s^x_{y,t}-s^x_{y,t-1}\right|+\varepsilon \tag{7}$$

where, N represents a number of pixels on the X-axis or the Y-axis in a block and $I(x,y)$ represents a characterizing function for designating whether the pixel in the current coordinate (x,y) is sampled or not, the $I(x,y)$ having a value of 1 when the (x,y) is sampled while having a value of 0 when the (x,y) is not sampled. Thus, if the search process is performed through the comparisons of original SAD values, the hill climbing effect is generated as illustrated in the following equation (8), such that it can be prevented that the search result is found to be the local minimum point, in the viewpoint of which the $\beta(t)$ is not always less than 0.

$$\begin{aligned} &\tfrac{1}{4}\ \textstyle\sum_x^N\sum_y^N\left|s^x_{y,t}-s^x_{y,t-1}\right|+\varepsilon_t \\ &\quad -\tfrac{1}{4}\textstyle\sum_x^N\sum_y^N\left|s^x_{y,t-1}-s^x_{y,t-2}\right|-\varepsilon_{t-1}<0 \\ &\quad \Rightarrow \tfrac{1}{4}\textstyle\sum_x^N\sum_y^N\left|s^x_{y,t}-s^x_{y,t-1}\right| \\ &\quad -\tfrac{1}{4}\textstyle\sum_x^N\sum_y^N\left|s^x_{y,t-1}-s^x_{y,t-2}\right|<\varepsilon_t-\varepsilon_{t-1} \\ &\quad \Rightarrow \tfrac{1}{4}\textstyle\sum_x^N\sum_y^N\left|s^x_{y,t}-s^x_{y,t-1}\right| \\ &\quad -\tfrac{1}{4}\textstyle\sum_x^N\sum_y^N\left|s^x_{y,t-1}-s^x_{y,t-2}\right|<\beta(t) \end{aligned} \tag{8}$$

4. Simulation Results

The proposed ME algorithm is verified with well known conventional ME methodology. In the computer simulation, a considered ME methodology is the full search method and another is 3-step search. In attempt to verify the proposed algorithm, we modify the conventional full search and 3-step search algorithms which are worked well on the full sampled macro block, to contain the proposed algorithm. Thus, it is convenient to compare the performance between the conventional ME algorithms and ME algorithms with the proposed algorithm.

When we adopt the proposed algorithm to the conventional algorithm, there exists a little number of different motion vectors comparing caused by the proposed algorithm comparing to the conventional algorithm. However, in spite of the existence of those difference of motion vectors, the average value of SAD for each macro block is very similar to each other. Consequently, if the proposed algorithm would be adopted to the conventional ME algorithm, a significant deterioration of picture quality is not occurred shown in table(). Since we conduct computer simulation for 10 frames of foreman moving picture, all simulation results represent average value for each ME algorithm.

Algorithm	SNR(dB)	TpF(%)
Full search method (FSM)	22.58	100
FSM with Stochastic Sampling	22.36	26.2
3-Step search method (TSM)	22.13	0.12
TSM with Stochastic Sampling	22.20	0.031

Table 1. Simulation results for each algorithm through early 10 frames of the foreman data

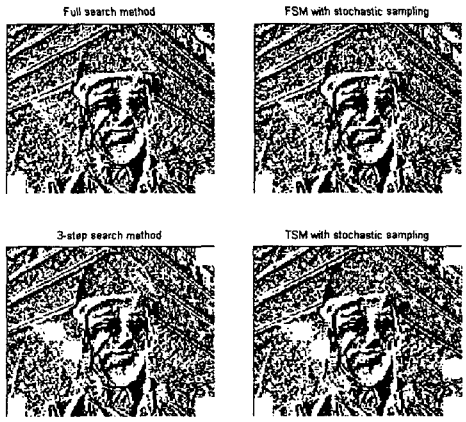

Fig. 2. Difference image with original frame at t+1 and reconstruction image by each ME algorithm at t

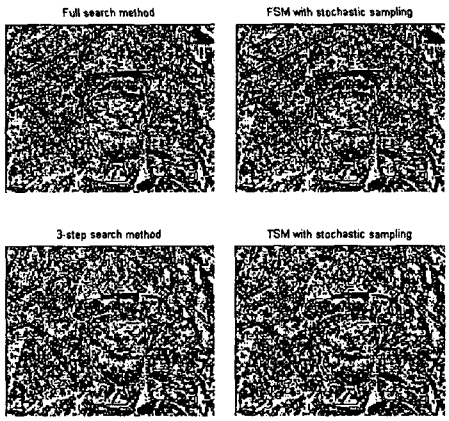

Fig. 3. Enhanced error image with original frame at t+1 and reconstruction images at t+1

5. Conclusions

The proposed stochastic sampling ME algorithm gives the fast operation and sufficient picture quality verified by the computer simulation. However, the proposed algorithm require full random access to system memory. In the viewpoint of an ME implementation on the system, it is significant problem to use random access to memory, since a lot of real system equipped with synchronized DRAM or a similar types of memory. In attempt to avoid this defection, we have to transport a search range of a previous frame to cache memory before ME operation. This preprocessing caused to consume a surplus time. Since most cases in a real implementation of ME needs this preprocessing, we claim that the proposed algorithm is sufficiently practical one.

References

[1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding" *IEEE Trans. Commun.*, , Vol. 29, pp. 1799-1808, 1981.

[2] H. G. Musmann, P. Pirsh and H. -J. Grallert, "Advances in picture coding" *Proc. IEEE*, Vol. 73, pp. 523-548, April, 1985.

[3] A. N. Netravali, J. D. Robbins, "Motion-compensated television coding : part I", *Bell Syst. Tech. J.*, Vol. 58, pp. 631-670, March, 1979.

[4] T. Koga et al, "Motion-compensated interframe coding for video conferencing", in *Proc. Nat. Telecommun. Conf.*, Nov./Dec., 1981, pp.G 5.3.1-G5.3.5.