PAPER *Special Section of Papers Selected from ITC-CSCC '98*

Block Matching Motion Estimation Based on Median Cut Quantization for MPEG Video

Hitoshi KIYA[†], ***Member***, **Jun FURUKAWA**[†], ***Student Member***, ***and*** **Yoshihiro NOGUCHI**[††], ***Nonmember***

SUMMARY We propose a motion estimation algorithm using less gray level images, which are composed of bits pixels lower than 8 bits pixels. Threshold values for generating low bits pixels from 8 bits pixels are simply determined as median values of pixels in a macro block. The proposed algorithm reduces the computational complexity of motion estimation at less expense of video quality. Moreover, median cut quantization can be applied to multilevel images and combined with a lot of fast algorithms to obtain more effective algorithms.

key words: *motion estimation, less gray level images, median cut quantization*

1. Introduction

MPEG (Moving Picture Experts Group), an international standard of video compression, adopts Discrete Cosine Transform (DCT) and Motion Compensation (MC). To compensate motion, it is necessary to estimate a motion vector called Motion Estimation (ME). The block matching motion estimation algorithm is widely used as a motion estimation algorithm because of its simplicity and high efficiency. Although the full search algorithm using 8 bits pixels is optimum in the compression ration, the huge computational cost makes it impractical when a search window is large. To overcome this problem, a lot of fast block motion estimation algorithms have been proposed.

Fast algorithms are classified into two classes. The first reduces the number of search locations [1]–[5] and the second reduces the amount of computation per one location by using images with bits pixels lower than 8 bits pixels [6]–[8]. The computational cost and processing power [9] can be lower if images with lower bits pixels are used because the operation requires less hardware. Moreover, the algorithms may use to generate candidate motion vectors for the full search algorithm with 8 bits pixels [10], [11], or may be combined with algorithms of the first class. The proposed algorithm in this paper belongs to the second class.

In [6], [7], two motion estimation methods using frames with 1 bit pixels were proposed. In these algorithms, frames in video sequences are quantized by using threshold values, before executing motion estimation. The principal difference between [6] and [7] is in how the threshold values are determined. The algorithms in [6] and [7], however, can not be applied to the generation of multilevel images such as 2-bits or 3-bits images.

In [8], the difference between a current block and a reference block is quantized to avoid uniform bits truncation schemes. Although this method is applicable to multilevel images, how to choose threshold values is more complicated than those of [6] and [7], because we have to use the hierarchical pyramid algorithm and generate the difference signal. Moreover, due to the use of the difference signal, this algorithm can not be combined with a lot of fast algorithms such as telescopic search [1] and 2D-Logarithmic search [2].

On the other hand, median cut quantization [12] can be used to generate multilevel images. It can also be applied to intra-frames. This property is important when we want to get better video quality than that using 1-bit images. As a result, the proposed algorithm reduces the computational complexity of motion estimation at less expense of video quality. It can also be combined with other fast algorithms.

2. Block Matching Motion Estimation Using Less Gray Level Images

We review block matching motion estimation using less gray level images.

2.1 Block Matching Motion Estimation

Let us divide a current frame into 16×16 blocks, called macro blocks. The process of block matching is to find a candidate macro block within a search window in a reference frame, which is the most similar to a macro block in the current frame. The accuracy of motion estimation depends on the matching criteria. One of the most popular criteria is MAD (Mean Absolute Difference), given by

$$MAD(k,l) = \frac{1}{16\times 16}\sum_{i=1}^{16}\sum_{j=1}^{16} \mid p(i,j) - f(i+k, j+l) \mid$$

Manuscript received September 30, 1998.
Manuscript revised January 3, 1999.

[†]The authors are with the Department of Electrical Engineering, Graduate School of Tokyo Metropolitan University, Hachioji-shi, 192–0397 Japan.
[††]The author is with the LSI Laboratories, Asahi Chemical Industry, Atsugi-shi, 243–0205 Japan.

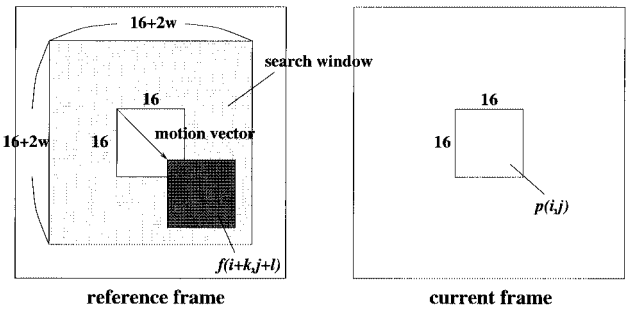


Fig. 1 Block matching motion estimation.

$$-w \leq k, l \leq w \tag{1}$$

where $p(i, j)$ is a pixel of a macro block in a current frame, $f(i, j)$ is a pixel of a search window in a reference frame and w is the size of a search window (See Fig. 1). (k, l) is the location in a search window. In this paper, we will use this criterion for block matching motion estimation.

The algorithm which evaluates a matching criterion for all values of (k, l) in a search window, is called the full search algorithm. The full search algorithm using 8 bits pixels is optimum in the compression ration. However, the huge computational cost makes it impractical when a search window is large.

2.2 Motion Estimation Using Less Gray Level Images

Here, we explain the block matching motion estimation using less gray level images. Less gray level images are generated from both of a current frame and a reference frame by using threshold values. For example, when we use linear quantization, the threshold values T_n are determined as follows.

$$T_n = \frac{256}{2^k} \times n \qquad (n = 1, \ldots, 2^k - 1) \tag{2}$$

where k is the number of depth bits on each pixel. After that, we reduce the number of bits of each frame from 8 bits pixels. For instance, for choosing $k = 2$, $T_1 = 64$, $T_2 = 128$ and $T_3 = 192$ are obtained, the quantization is executed as

$$p'(i,j) = \begin{cases} 3, & p(i,j) \geq T_3 \\ 2, & T_3 > p(i,j) \geq T_2 \\ 1, & T_2 > p(i,j) \geq T_1 \\ 0, & T_1 > p(i,j) \end{cases} \tag{3}$$

where $p(i, j)$ is the value of a pixel.

In the conventional methods [6], [7], mean and filtering operations were used to determine threshold values. These methods are limited to the generation of 1 bit images and thus are not applicable to multilevel images. Although linear quantization is applicable to multilevel images, motion estimation by using it suffers from the degradation of video quality.

On the other hand, median cut quantization, which will be used in the proposed algorithm, is simple and can be applied to multilevel images, while satisfying less expense of video quality.

3. Median Cut Quantization and Its Application

In this paper, we use median cut quantization [12] to generate fewer bits pixels from 8 bits pixels. Here, we explain median cut quantization and describe its application to block motion estimation.

3.1 Median Cut Quantization

Threshold values for median cut quantization are given as follows. The histogram of each macro block (16×16) in a current frame is computed. Then, the threshold values are chosen as median values of the histogram. For example, if we generate 1 bit pixels from 8 bits pixels, the threshold T_1 will be a value to divide equally the number of pixels in the macro block. Now, we arrange $p(i, j)$ in the macro block $(i, j = 1, \ldots, 16)$ as $p(1) \leq p(2) \leq \ldots \leq p(256)$ according to the values. Then the threshold is given by

$$T_1 = p(128) \tag{4}$$

Similarly, for generating 2 bits pixels, 3 threshold values T_1, T_2 and T_3 are given by

$$T_1 = p(64), \quad T_2 = p(128), \quad T_3 = p(192) \tag{5}$$

The values of pixels in each macro block of a current frame are truncated as Eq. (3). The same threshold values are also used to be truncated the pixels of a search window which corresponds to each macro block.

3.2 Algorithm with Full Search

In general, the full search algorithm is carried out by using images with 8 bits pixels. In this paper, less gray level images generated by median cut quantization are used for the full search algorithm. Thus, the computation of MAD is given by

$$MAD(k,l) = \frac{1}{16 \times 16} \sum_{i=1}^{16} \sum_{j=1}^{16} \mid p'(i,j) - f'(i+k, j+l) \mid$$
$$-w \leq k, l \leq w \tag{6}$$

where $p'(i, j)$ and $f'(i, j)$ are quantized pixels of $p(i, j)$ and $f(i, j)$ respectively. We look for the smallest $MAD(i, j)$ in a search window entirely. (k, l) that gives the minimum $MAD(k, l)$ in the search window is used as the motion vector of the macro block. As a result, it is possible for us to reduce the computation complexity compared to the 8 bits pixels approach.

3.3 Combining with Other Algorithms

The use of less gray level images can be combined with the most of fast algorithms which reduce the number of search locations. These combinations give great reduction of the computational complexity of motion estimation.

In this paper, some simulations will be performed to evaluate the effectiveness of these combinations by using telescopic search [1] and 2D-Logarithmic search [2]. These algorithms are briefly reviewed as follows:

Telescopic search: In telescopic search, frames between a reference frame and a current frame are used as middle reference frames. Motion estimation is executed with tracking the middle reference frames. Each motion vector is used as an initial point for the next motion estimation to reduce the range of search windows.

2D-Logarithmic Search: In 2D-logarithmic search, the search starts with a coarser step size and reduces the step size successfully. Each step consists of searching five locations which contain the center of area and the four points corresponding to the north, south, east and west directions at the distance specified by the step size from the center. The search procedure continues until the search window is reduced to 3×3 pixels. In this step, all nine locations within the window are searched.

4. Discussion on Quality

To compare the performance of motion estimation, some simulations are performed. Here, we use three gray video sequences with 10 frames. They are mobile & calendar (MC), flower garden (FG) and football (FB) of size 352×240. These sequences are coded by MPEG-1 algorithm within the MPEG2 software encoder [13] and coded on 1 Mbps and 128 kbps as target bit rate. The bit rate control is based on deviancy from estimated buffer fullness and normalized spatial activity. In those simulations, we compare the peak signal-to-noise ratio ($PSNR$) of decoded images, defined as

$$PSNR = 10 \log_{10} \frac{ImageSize * 255^2}{\sum_{i=1}^{16} \sum_{j=1}^{16} \{p(i,j) - f(i+k, j+l)\}^2} \quad (7)$$

4.1 Full Search

We encode these video signals using motion vectors which are estimated with low bits images. Figure 2 is the PSNR of encoded images on 1 Mbps. Table 1 shows the PSNR average of the encoded images.

From these results, we see that

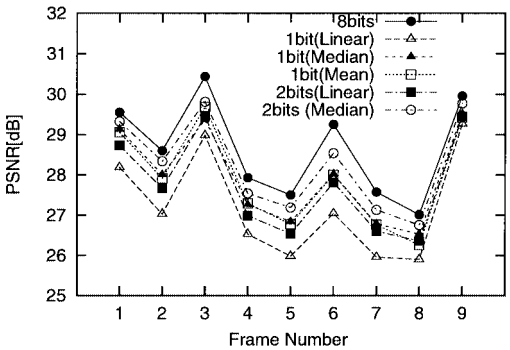


Fig. 2 PSNR of encoded images on 1 Mbps (FB).

Table 1 The comparison of PSNR average of encoded images.

(a) 1 Mbps

	FG	MC	FB
1 bit(Linear)	27.20	27.19	27.21
1 bit(Median)	27.91	27.43	27.95
1 bit(Mean)	27.14	27.30	27.91
2 bits(Linear)	27.89	27.74	27.74
2 bits(Median)	28.21	27.77	28.27
8 bits	28.35	28.07	28.65

(b) 128 kbps

	FG	MC	FB
1 bit(Linear)	24.77	24.56	25.84
1 bit(Median)	24.91	24.62	25.99
1 bit(Mean)	24.75	24.60	26.07
2 bits(Linear)	24.96	24.74	26.02
2 bits(Median)	25.00	24.75	26.12
8 bits	25.07	24.85	26.39

FG:flower-garden MC:mobile&calendar
FB:football

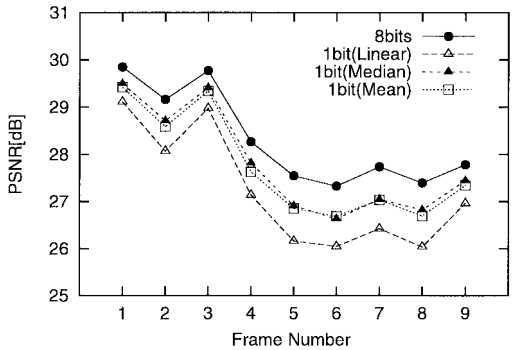
- the PSNR degradation of the proposed method using median cut quantization is less than 0.7 dB, even with 1 bit images compared to the use of 8 bits images
- linear quantization is inferior to other methods and PSNR by median cut quantization for 1 bit truncation is almost the same as that by linear quantization for 2 bits
  This is why linear quantization is independent of the property of images.

Please note that mean operation [6] can not be applied to multibit images. When we use 3 bits images, PSNR of encoded images is nearly equal to that of 8 bits images. Therefore, the PSNR of encoded images is independent of the methods to determine threshold values for more than 2 bits images.

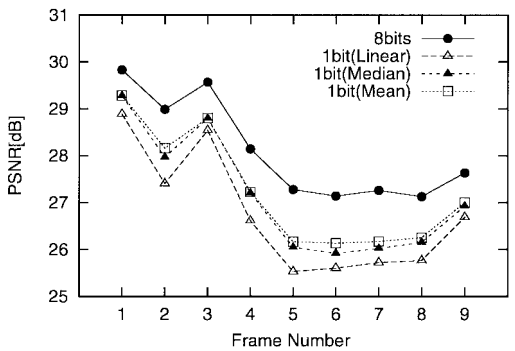
From the above discussion, we can see that median cut quantization not only can be applied to multilevel images but also can be executed at less expense of video quality.

4.2 Fast Algorithms with Low Bits Images

Here, we execute some fast algorithms with low bits



(a) telescopic search



(b) 2D Logarithmic search

Fig. 3 Combining of algorithms on 1 Mbps (FG).

images. We use telescopic search [1] and 2D-logarithmic search [2] as fast algorithms. From Fig. 3 and Table 2, we see that

- the PSNR degradation of telescopic search with 1 bit images is less than 0.7 dB and that of 2D-logarithmic search with 1 bit images is less than 0.97 dB compared to the use of 8 bits images
- linear quantization is inferior to other methods

5. Discussion on Computational Complexity

We compare the computational complexity based on bit-serial operation since our proposed method uses lower bits pixels. We show how many clock cycles needed for all operations of motion estimation. It is assumed that the computational complexity of each operation is linear to the number of bits. For instance, the complexity of 8 bits operation is 8 times as large as that of 1 bit operation. Moreover, the number of operations is scaled to 8 bits addition and subtraction operations as shown in [8].

First, let us consider the number of operations per one location. We perform $256/8 \times n$ subtractions to compute one MAD for n bits images since the size of a macro block is $16 \times 16 = 256$. On the other hand, to calculate the MAD, we perform 128 n-bits additions, then 64 $(n+1)$-bits additions and so on. Therefore,

Table 2 Combining with other algorithms.

(a) telescopic search (1 Mbps)

	FG	MC	FB
1 bit(Linear)	27.24	27.26	27.19
1 bit(Median)	27.81	27.44	27.90
1 bit(Mean)	27.73	27.53	28.03
2 bits(Linear)	27.89	27.76	27.78
2 bits(Median)	28.14	27.81	28.17
8 bits	28.32	28.05	28.60

(b) telescopic search (128 kbps)

	FG	MC	FB
1 bit(Linear)	24.78	24.60	25.82
1 bit(Median)	24.90	24.63	26.04
1 bit(Mean)	24.92	24.65	26.11
2 bits(Linear)	24.95	24.74	26.06
2 bits(Median)	24.99	24.76	26.14
8 bits	25.06	24.83	26.34

(c) 2D-Logarithmic search (1 Mbps)

	FG	MC	FB
1 bit(Linear)	26.75	26.86	27.08
1 bit(Median)	27.14	26.91	27.41
1 bit(Mean)	27.24	27.06	27.58
2 bits(Linear)	27.46	27.31	27.54
2 bits(Median)	27.47	27.33	27.65
8 bits	28.11	27.63	28.24

FG:flower-garden MC:mobile&calendar
FB:football

the following additions are needed for calculating one MAD.

$128 \times (\quad n \quad)$-bits addition
$64 \times (n+1)$-bits addition
$32 \times (n+2)$-bits addition
$16 \times (n+3)$-bits addition
$8 \times (n+4)$-bits addition
$4 \times (n+5)$-bits addition
$2 \times (n+6)$-bits addition
$1 \times (n+7)$-bits addition

Thus, the amount of operation for computing one MAD with n-bits images are given by

$$\begin{aligned} N(n) &= 32n + \sum_{k=0}^{7} 2^k (n+7-k)/8 \\ &= 32n + (255n + 247)/8 \end{aligned} \tag{8}$$

For example, in case of $n = 1$ and $n = 8$, $N(1)$ is 95 and $N(8)$ is 542 respectively.

It is also assumed that the computational complexity of comparison operation which is used for truncating the values of pixels, is equal to that of 8 bits addition and subtraction. The comparison operation is executed for each macro block in a current frame and each search window in a reference frame. Moreover, we need to take account of the computational complexity for determining threshold values. To determine threshold values for median cut quantization, we have to execute sorting

Table 3 The number of total operations for each macro block.

Algorithm	bits	Linear	Median
Full	8	121950	121950
Full	4	64607	68857
Full	3	50207	54489
Full	2	35807	40057
Full	1	21407	25657
Telescopic	8	40650	40650
Telescopic	4	21557	26107
Telescopic	3	16757	20307
Telescopic	2	11957	16507
Telescopic	1	7156	11707

operation for each macro block. When executing sorting operation by quick sort [14], both of comparison and swap operations are needed. It is also assumed that the computational complexity of swap operation is equal to that of 8 bits addition and subtraction as well as that of comparison operation. Then, we can show that the number of operations for comparison operation is 2467 and that for swap operation is 659 for each macro block.

In Table 3, we compared the number of total operations for each macro block. The number of total operations includes the number of operations for sorting for median cut quantization, truncating the values of pixels and MAD. In addition, we used full search and telescopic search by using linear quantization or using median cut quantization . The maximum motion vector is 7 in both direction and the size of the search window is 30×30.

From the table, it is shown that the computational complexity of motion estimation can be reduced by using lower bits pixels. The number of operations of the algorithms by using linear quantization is fewer than that by using median cut quantization. However, the methods using 1-bit images by median cut quantization can get the same video quality as that using 1-bit images by linear quantization. And the number of operations by using median cut quantization for 1 bit is fewer than that by using linear quantization for 2 bits.

6. Conclusions

In this paper, we described the method for determining threshold values by median cut quantization and a motion estimation algorithm using less gray level images. Our method is executed with full search and moreover can be combined with most of other fast algorithms. The proposed method can reduce the computational complexity of motion estimation. In addition, it was also shown that median cut quantization is applicable to multibit images and gives motion vectors with less expense of video quality.

We plan to evaluate the effectiveness of power consumption of the proposed method as a future work.

References

[1] K. Matsuda, T. Tsuda, T. Ito, and S. Maki, "A new motion compensation coding scheme for video conferences," IEEE Int. Commun. Conf. pp.234–237, 1984.
[2] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol.COM-29, pp.1799–1808, Dec. 1984.
[3] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans. Commun., vol.38, pp.950–953, July 1990.
[4] B. Liu and A. Zaccarin, "New fast algorithm for the estimation of block motion vectors," IEEE Trans. Circuits & Syst. for Video Technol., vol.3, pp.148–157, April 1993.
[5] A.M. Tekalp, "Digital video processing," Prentice Hall, NJ, 1995.
[6] J. Feng, K.-T. Lo, H. Mehrpour, and A.E. Karbowiak, "Adaptive block matching motion estimation algorithm using bit-plane matching," IEEE Int. Conf. Image Proc., Wash. D.C, pp.496–499, 1995.
[7] B. Natarajan, B. Vasudev, and K. Konstantinides, "Low-complexity algorithm and architecture for block-based motion estimation via one-bit transforms," IEEE ICASSP'96, pp.3244–3247, 1996.
[8] L. Jiang, K. Ito, and H. Kunieda, "Bits truncation adaptive algorithm for motion estimation of MPEG2," IEICE Trans. Fundamentals, vol.E80-A, no.8, pp.1438–1445, Aug. 1997.
[9] Y. Sasajima and T. Enomoto, "Snapping-off motion vector estimation algorithm using reduced pixel data," Proc. 1998 Information and Systems Society Conference of IEICE, pp.117, 1998.
[10] S. Lee and S.-I. Chae, "Two-step motion estimation algorithm using low-resolution quantization," IEEE ICIP'96, vol.3, pp.795–798, 1996.
[11] S. Lee and S.-I. Chae, "New motion estimation algorithm and its block-matching criteria using low-resolution quantization," ITC-CSCC'98, vol.1, pp.175–178, 1998.
[12] P. Heckbert, "Color image quantization for frame buffer display," Computer Graphics, vol.16, no.3, July 1982.
[13] MPEG Software Simulation Group, "MPEG-2 encoder/decoder version 1.2," [Online] Available WWW: http://www.mpeg.org/tristan/MPEG/MSSG
[14] A.V. Aho, J.E. Hopcraft, and J.D. Ullman, "Data structures and algorithms," Addison-Wesley, 1987.

Hitoshi Kiya was born in Yamagata, Japan, on November 16, 1957. He received the B.E. and M.E. degrees in electrical engineering from Nagaoka University of Technology, Niigata, Japan, and the D.E. degree in electrical engineering from Tokyo Metropolitan University, Tokyo, Japan, in 1980, 1982, and 1987, respectively. In 1982, he joined Tokyo Metropolitan University, where he is currently an Associate Professor of Department of Electrical Engineering. His research interests are in digital signal processing, multirate systems, adaptive filtering, image processing, and efficient algorithms for VLSI implementation. He is an Associate Editor of Trans. on Signal Processing of IEEE and Trans. of IEICE. He is a Member of the IEEE, the Image Electronics Engineers of Japan and the Institute of Television Engineers of Japan.

Jun Furukawa was born in Fukushima, Japan, on December 24, 1974. He received the B.E. degree in electronics and information engineering from Tokyo Metropolitan University in 1997. He is currently a candidate for the M.E. degree at Tokyo Metropolitan University. His research interest is in image processing.

Yoshihiro Noguchi was born in Okayama, Japan, on March 2, 1962. He received the B.E. degree from Kobe University in 1985. In 1985, he joined Asahi Chemical Ind., where he is currently a candidate for D.E. degree at Tokyo Metropolitan University. He used to be a visiting scholar at EECS of UC Berkeley from 1994 to 1996. His research interests are in image processing and VLSI implementation. He is a member of IEEE.