PAPER *Special Section on Digital Signal Processing*

Motion Estimation Using Edge Enhanced Low-Bit Images for Lowpower MPEG Encoder

Ayuko TAKAGI[†a)], ***Student Member***, **Kiyoshi NISHIKAWA**[†], ***and*** **Hitoshi KIYA**[†], ***Regular Members***

SUMMARY This paper propose a method for improving the image quality of motion estimation (ME) using low-bit images. By using edge-enhanced images for quantization, we can increase the accuracy of the ME and improve the image quality. It is known that using low-bit images for ME is effective for reducing power consumption but it slightly degrades image quality. The quality of the encoded image depends on the thresholds for data quantization, thus, algorithms for determining thresholds are studied. The proposed method uses linear quantization, which simply truncates the least significant bits. This method is simple without any complicated threshold calculations, and the resultant image quality is improved as much as the methods that use threshold calculations. To evaluate the effectiveness, we simulate results for image quality and estimate the power consumption using synthesis results from a VHDL model motion estimator.

key words: *low power, motion estimation, MPEG, low-bit image, edge enhancement*

1. Introduction

We propose a method for improving the image quality of motion estimation (ME) using low-bit images. In the conventional method, complex threshold calculations are needed to improve the image quality for ME using low-bit images. In the proposed method, by enhancing the edges of an image in advance, the resultant encoded-image quality can be improved compared to that when using fewer bits/pixel for linear quantization, which needs no threshold calculation. In this paper we use the term 'low-bit ME' to indicate motion estimation using low-bit images.

In many multimedia applications, video data is often compressed using the MPEG standard. There is considerable temporal redundancy in image sequences, and the MPEG codec uses ME to eliminate the temporal redundancy of video frames. ME is the most time-consuming task in encoding video sequences and it also consumes the most power [1].

Many algorithms have been studied in efforts to reduce the complexity of ME and save power [2]–[8]. These algorithms can be classified into two approaches. The first approach is to reduce the number of search locations in ME and the second is to reduce the amount of computations per location. The second approach uses images with far fewer bits/pixels (low-bit images) compared to the original images. In this paper, we concentrate on the second approach, since it is effective for reducing power consumption [9], [10], and since the two approaches are independent from one another and may be combined. Reducing the bit width of input images reduces power consumption, because the power required for ME is proportional to the bit width of the input image. When preparing low-bit images, we must determine thresholds for quantization. These thresholds influence the encoded image quality, and improving the image quality increases the complexity of threshold calculations.

Many algorithms for preparing low-bit images have been proposed that differ on how to determine the thresholds for deciding whether or not to transform 8-bit/pixel data to data using fewer bits per pixel [2], [3], [5]–[7]. A typical approach uses a linear quantization that truncates the least significant bits. This method is very simple and requires no calculations to determine thresholds, but the ME becomes inaccurate and the image quality is much degraded compared to the following approach. Other approaches calculate the thresholds from the image data to decrease the image degradation. The calculated thresholds may be either the median values [2] or the mean values [3]. By using a low-bit image generated using these quantization methods, the ME is more accurate and the image quality is improved. However, calculating the thresholds is complex because of the need to search through the image data [11]. Therefore, we want a low-bit ME with simple quantization that also causes less image degradation.

In this paper, we propose a method for improving image quality when we use a low-bit ME with linear quantization. In our method, we perform edge enhancement before quantizing the image data. By using edge-enhanced images for quantization, it is shown that the ME becomes more accurate and the quality of the encoded image improves for linear quantization. This edge-enhancement operation is independent of the quantization bit width so that it is also suitable for use with an MPEG encoder with power scalability [9], [10]. To evaluate the effectiveness of this method, we use simulation results and estimate the power consumption

Manuscript received December 6, 2000.
Manuscript revised March 5, 2001.

[†]The authors are with the Department of Electrical Engineering, Graduate School of Engineering, Tokyo Metropolitan University, Hachioji-shi, 192-0397 Japan.
a) E-mail: ayuko@isys.eei.metro-u.ac.jp

using the synthesis results from a VHDL model motion estimator.

2. Preparation

We propose a method for improving the image quality for low-bit motion estimation. In our proposed method, we perform edge enhancement on images. We first briefly review block-matching ME and the method of using low-bit images. We then describe a method for edge enhancement.

2.1 Block-Matching Motion Estimation

The MPEG codec is based on ME, the process that takes most of the time required for video encoding. There are several kinds of algorithms used for ME, and block-matching algorithms are widely used in many kinds of video encoding.

To show how these algorithms work, we assume that a current frame is divided into blocks with a size of 16×16 pixels, called macro blocks. The process of block-matching is to find the macro blocks in a search window of previous frames that are most similar to the macro blocks in the current frame.

The accuracy of ME depends on the matching criteria, and one of the most popular criteria is the sum of absolute difference, or SAD, given by

$$SAD(k,l) = \sum_{i=1}^{16}\sum_{j=1}^{16} |P_t(i,j) - P_{t-1}(i+k, j+l)|, \tag{1}$$

where (k,l) is the location in the search window, $P_t(i,j)$ is a pixel at (i,j) in the current frame, and $P_{t-1}(i,j)$ is a pixel in the previous frame. When the value $SAD(k,l)$ is minimum, (k,l) is the motion vector of the macro block.

A full-search block-matching algorithm exhaustively examines all the locations in a search window. This algorithm enables us to find the minimum-error block, but it requires a large amount of computations. There are other algorithms which reduce the number of search locations to reduce ME the computation. These algorithms are generally executed using 8-bit images. One way to reduce the complexity of ME is to use low-bit images for the ME, as described in the next section.

2.2 Motion Estimation Using Low-Bit Images

In general, 8-bit/pixel data is used for ME. To reduce the complexity of ME, there are methods that use low-bit images with less than 8bits per pixel. In these methods, the pixel data is quantized with the thresholds calculated for each macro block in the current frame. The images generated this way are used to calculate the motion vector implicit in the minimum solution of Eq. (1). Using low-bit images reduces the calculation time required for ME.

The image quality of the encoded image differs among the various methods for making low-bit images. A typical approach is linear quantization that truncates the least significant bits. Linear quantization is easy to implement on hardware architecture, but it is well known that ME becomes inaccurate and image degradation is quite large. Other methods calculate the thresholds from the image data to decrease the image degradation. The calculated thresholds may be either the median values or the mean values [2], [3]. Using these quantization methods, ME becomes more accurate and the image quality improves. However, since these quantization methods require that the threshold be calculated, implementation is difficult [11].

2.3 Edge Enhancement

Edge enhancement is generally used to sharpen a blurred image. The most common method of edge enhancement is to generate a Laplacian image and subtract it from the original image.

The Laplacian is an approximation to the linear second derivative of the image given as

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}, \tag{2}$$

where $f(x,y)$ is the pixel value at position (x,y). This Laplacian is invariant to rotation, and hence insensitive to the direction in which the discontinuity runs. This highlights the points, lines, and edges in the image and suppresses uniform and smoothly varying regions. Therefore, this method is widely used for edge detection and edge enhancement.

For digital signals, Eq. (2) is applied using a 2-dimensional convolution of image $f(i,j)$ and filter $h(i,j)$

$$\begin{aligned} \nabla f(i,j) &= \sum_{k_1}\sum_{k_2} h(k_1,k_2) f(i-k_1, j-k_2) \\ &= h(i,j) * f(i,j) \end{aligned} \tag{3}$$

$$h(i,j) = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}, \tag{4}$$

where $*$ denotes the convolution and $h(i,j)$ is the 3×3 Laplacian operator, which is used to detect the edges. By adding the Laplacian from the original image, we can make the slope of the edge steeper and obtain a sharpened image $g(i,j)$. The equation is given by

$$g(i,j) = f(i,j) + \alpha \cdot h(i,j) * f(i,j), \tag{5}$$

where α is a constant indicating the enhancement rate. In this paper, we perform edge enhancement based on Eqs. (4) and (5).

3. Proposed Method Using Edge Enhancement

We next describe the proposed method for improving the image quality of low-bit ME.

3.1 Concept of the Proposed Method

When performing low-bit ME, it becomes difficult to predict the motion vector and some motion vectors cannot be predicted even though the motion vector originally exists. This is caused by the reduction of the dynamic range of ME, and the image quality depends on the generated low-bit images, which differ by their thresholds for quantization. By using quantization methods that calculate the thresholds, such as median quantization, we can limit the deterioration of image quality compared to using linear quantization. This is because linear quantization does not consider the characteristics of the image.

When using a quantization method that calculates the thresholds, the calculation is done by searching through the original image data. This ensures that the characteristics of the original image are retained in the generated low-bit images. Compared to this type of quantization method, the linear quantization method simply truncates the LSB data from the original image, fading the characteristics of the original image from the low-bit image. Although quantization methods that calculate the thresholds maintain image quality, they are difficult to implement because of the calculation requirements. Linear quantization, on the other hand, is easy to implement but it degrades image quality. Therefore, we propose a method for improving the image quality for linear quantization while keeping the advantage of its simplicity for implementation.

In our proposed method, we improve the image quality by performing edge enhancement on the original image before linear quantization. This enables us to enhance the characteristics of the original image and pass them on to the low-bit image. Edge information is a special characteristic among image properties and has a favorable effect on ME. Compared to a low-bit image generated directly from the original image, a low-bit image generated from an edge-enhanced image has stronger edge information. This enables us to predict more motion vectors and increase the accuracy of the ME which leads to the improvement of image quality.

3.2 Procedure of the Proposed Method

The ME procedure for the proposed method is as follows (The flow of the method is shown in Fig. 1):

Step 1. Perform edge enhancement on the original images.

Step 2. Calculate thresholds for quantization from the edge-enhanced image.

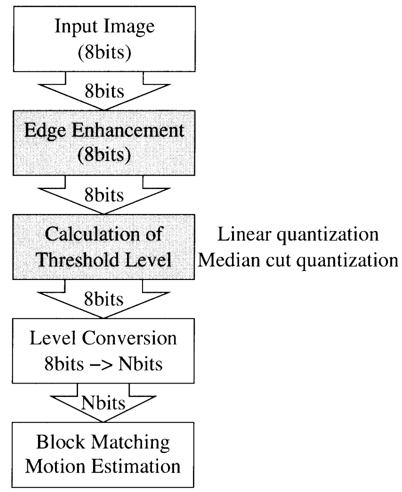


Fig. 1 Proposed method for low-bit ME.

Step 3. Generate a low-bit image using the thresholds.

Step 4. Perform ME using the low-bit image generated in Step 3.

The only difference from the conventional method is the edge enhancement in Step 1. Just adding this single step significantly improves the image quality for low-bit ME, and the accuracy of the ME increases.

The ME procedure shown above is a general case of our method. Since our goal is to improve the image quality of low-bit ME with simple quantization, here, we show the procedure using linear quantization.

Step 1. Perform edge enhancement on the original images.

Step 2. Generate a low-bit image by truncating the LSB data.

Step 3. Perform ME using the low-bit image generated in Step 2.

When using linear quantization, we can omit the step for calculating the thresholds so that the ME procedure is simplified. By using linear quantization, the effects of edge enhancement are more clear than when using a complex quantization method, such as median cut quantization.

4. Discussion on Computational Complexity

In this section, we compare the computational complexity based on a bit-serial operation. We assume that the computational complexity of each operation is linear to the number of bits used as well as in other papers [2], [7], [12]. For instance, the complexity of 8-bit operation is eight times as large as that of 1-bit operation. Moreover, the number of operations is scaled to 8-bit addition and subtraction operations. Here, we consider the number of operations when using n bits/pixel data for low-bit ME.

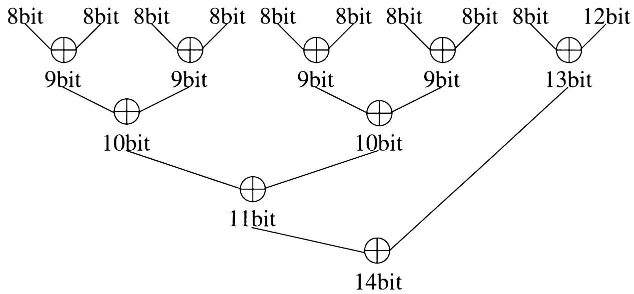


Fig. 2 Binary tree structure for summation of edge enhancement ($n = 8$).

4.1 Computation of Eq. (5)

First, we consider the computational complexity for making an edge-enhanced image from Eq. (5). The Laplacian filter we use is Eq. (4). Here, we set $\alpha = 1$ for simplicity, and in this case Eq. (5) is defined as

$$\begin{aligned} & g(i,j) \\ &\quad = f(i,j) - f(i-1,j-1) - f(i,j-1) \\ &\qquad - f(i+1,j-1) - f(i,j-1) + 8f(i,j) \\ &\qquad - f(i+1,j) - f(i+1,j-1) - f(i+1,j) \\ &\qquad - f(i+1,j+1) \end{aligned} \tag{6}$$

where $8f(i,j)$ can be implemented by a 3-bit shift. Therefore, Eq. (6) can be considered as a summation for $9{\times}(n\text{bit})$ data and an $(n+3)$bit data. The number of operations for this summation can be calculated as a binary tree structure and is illustrated in Fig. 2 for $n = 8$. As a result, the total number of operations for Eq. (6) becomes

$$\begin{array}{rl} & 4 \times (n)/8 \\ & 1 \times (n+3)/8 \\ & 2 \times (n+1)/8 \\ & 1 \times (n+2)/8 \\ +) & 1 \times (n+4)/8 \\ \hline \text{total} & (9n+13)/8 \end{array} \tag{7}$$

It is noted that this edge enhancement is independent from the quantization bit width and can be implemented by a filtering operation. In contrast, the operation for calculating thresholds depends on the quantization bit width and it is calculated for each macro block. This characteristic of computation especially gives us an advantage for implementing ME with power scalability [9], [10].

4.2 Computation of SAD [2], [7], [12]

Here, we show the number of operations needed to calculate SADs from Eq. (1). When using $n$bits/pixel data, the computation for each operation is $n/8$. Since the macro block size is 16×16, the number of operations needed are $256 \times n/8$ for subtractions, $256 \times n/8$ for calculating the absolute value, and summation of these 256 values. For the summation of these 256 values, we perform 128 $n$bits additions, then $64(n+1)$bits additions, and so on. And the following numbers of additions are need for summation:

$$\begin{array}{rl} & 128 \times (n)/8 \\ & 64 \times (n+1)/8 \\ & 32 \times (n+2)/8 \\ & 16 \times (n+3)/8 \\ & 8 \times (n+4)/8 \\ & 4 \times (n+5)/8 \\ & 2 \times (n+6)/8 \\ +) & 1 \times (n+7)/8 \\ \hline \text{total} & (255n+247)/8. \end{array} \tag{8}$$

The total number of operations to calculate one SAD is

$$\begin{aligned} N(n) &= (255n + 255n + 255n + 247)/8 \\ &= (767n + 247)/8. \end{aligned} \tag{9}$$

4.3 Number of Operations for Edge Enhancement and ME

To show the computational complexity, we consider the number of operations for computing one motion vector. The results are shown in Tables 1 and 2.

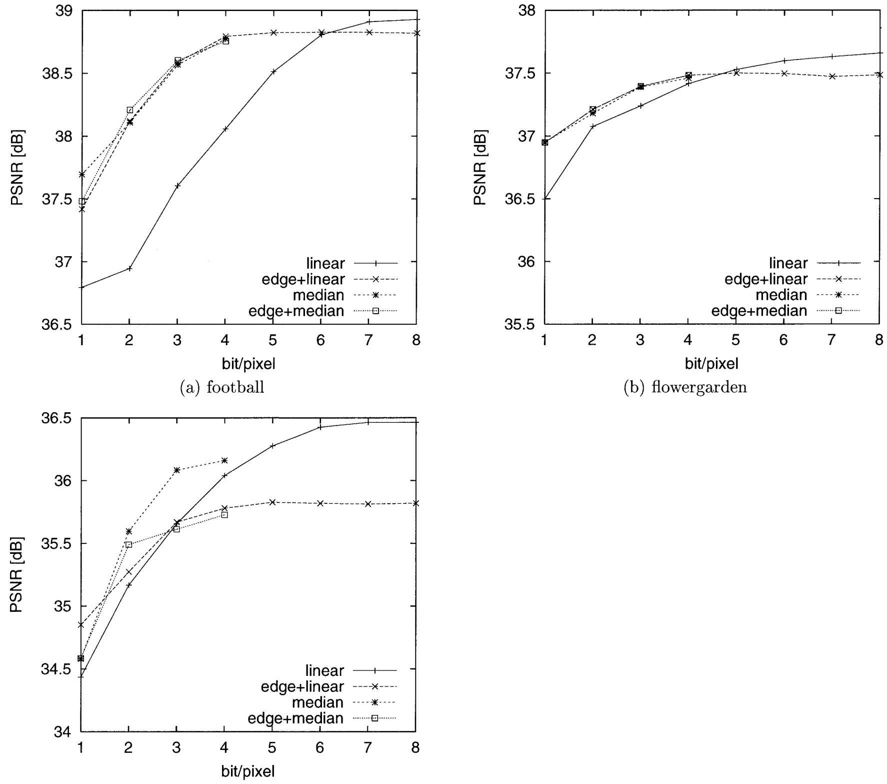
For edge enhancement, the number of operations to calculate a pixel is shown in Eq. (7). Since the edge enhancement is performed with an 8-bit/pixel image ($n = 8$), the total number of operations is 85/8. When performing ME, we need two macro blocks, one from

Table 1 The number of operations for computing edge enhancement.

bits	number of operations
8	5440

Table 2 The number of operations for ME when calculating one motion vector.

bits	number of operations
8	204256
7	179712
6	155168
5	130624
4	106080
3	81536
2	56992
1	32448



(a) football

(b) flowergarden

(c) mobile & calendar

Fig. 3 PSNR comparison for low-bit ME using an edge-enhanced image with different quantization methods. ($\alpha = 1$ for edge enhancement), ("linear" and "median" stand for linear quantization and median cut quantization, "edge+" means the usage of an edge-enhanced image)

the current frame and the other from the reference frame. Thus, the total number of operation for making an edge-enhanced image is $16 \times 16 \times 2 \times 85/8 = 5440$.

Next, we calculate the number of operations for ME. To compute a motion vector, we need to calculate the SADs for each macro block within the search window. When the search window size is 16×16, the number of operations needed for computing a motion vector is

$$N(n) = 16 \times 16 \times (767n + 247)/8. \quad (10)$$

The number of operations for edge enhancement is about 2.7% compared to those for 8-bit ME.

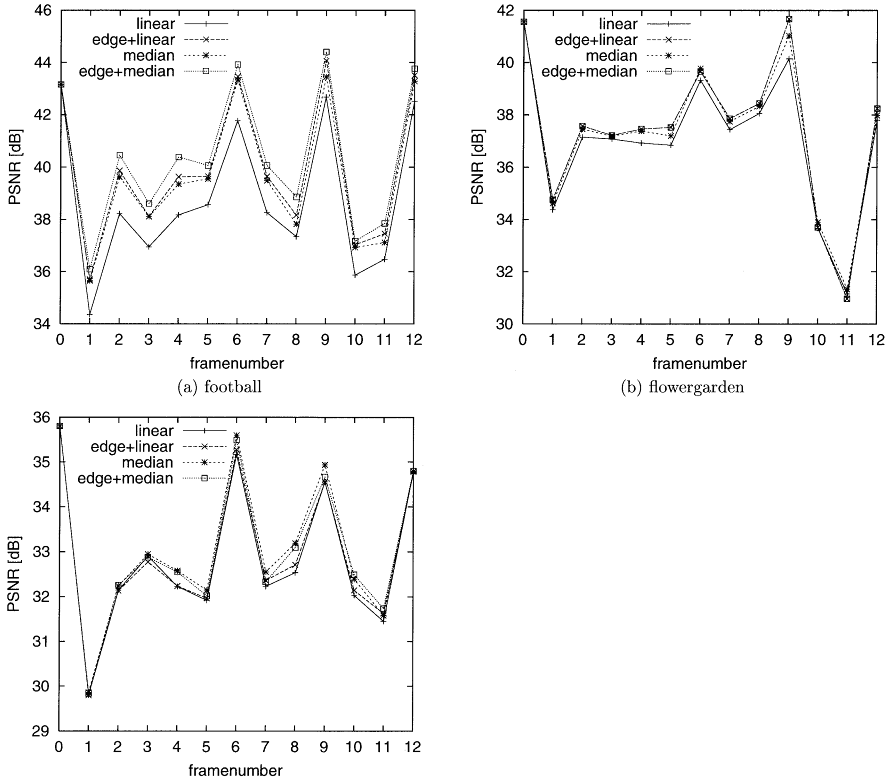
5. Simulation Using the Proposed Method

We next discuss the simulation results for the proposed method.

5.1 PSNR Comparison

In our simulation, we used the MPEG-2 codec [13] with grayscale video sequences of "football," "flowergarden" and "mobile & calendar" (size: 704×240 pixels). The search window size was 31×31, and the GOP structure was IBBPBBPBBPBBI where I, P, and B stand for intra-coded picture, predictive-coded picture, and bidirectionally predictive-coded picture. The video data was 8 bits/pixel with no signs. We performed the edge enhancement, reduced the 8-bit/pixel data using linear quantization, and simulated its processing at 10 Mbps. For ME we used a full-search block-matching algorithm. The full-search block-matching algorithm is the basic search algorithm used for ME so that we can verify the isolated effectiveness of the proposed method using the edge enhancement.

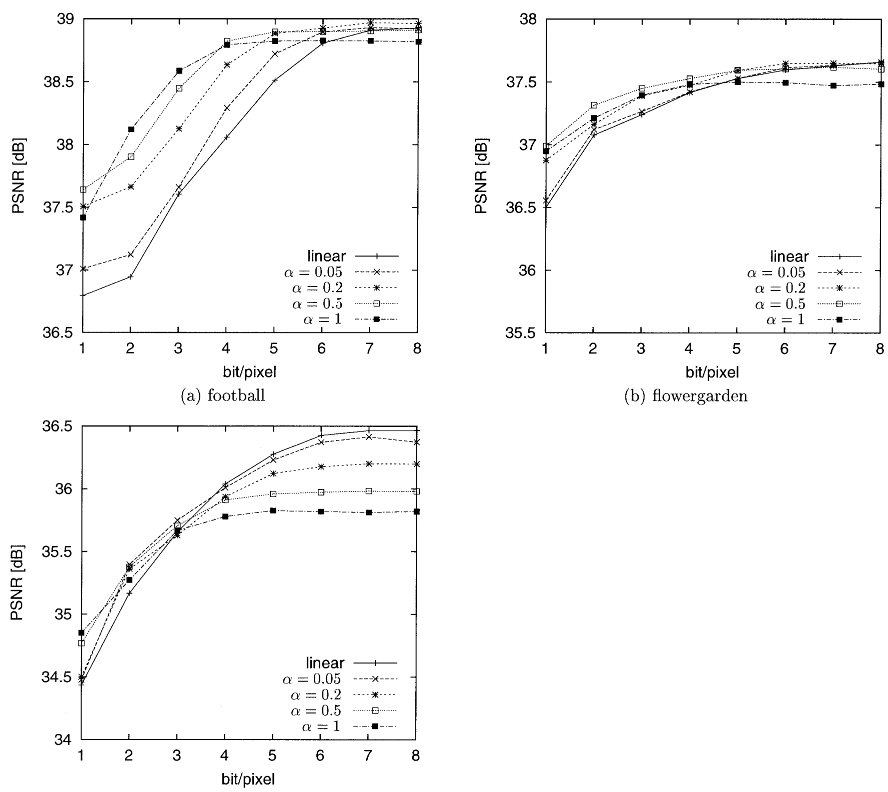
Figure 3 shows a PSNR comparison of encoded images using the proposed method for linear quantization and median cut quantization. The enhancement rate was $\alpha = 1$ in this simulation. We compared the difference between the proposed ME, with edge enhancement, and the conventional ME, without edge enhancement. The quantization methods we used are linear quantization and median quantization. The image quality deteriorates when data with fewer bits per pixel is used. This shows that using the edge-enhanced image improves the image quality for linear quantization. This means that the image quality of the conventional method can be obtained using fewer bits/pixel data with the proposed method, although the amount of quality improvement depends on video sequences.



(a) football (b) flowergarden

(c) mobile & calendar

Fig. 4 PSNR comparison of GOP structure when using 2-bits/pixel data for low-bit ME. ($\alpha = 1$ for edge enhancement)



(a) football (b) flowergarden

(c) mobile & calendar

Fig. 5 PSNR comparison for low-bit ME using an edge-enhanced images with different enhancement rates. (quantization method is the linear type)

The effect of the edge enhancement can also be verified for lower bits/pixel such as 1 or 2bits/pixel, even though the amount of quality improvement is small for Figs. 3(b) and (c). We consider that the quality improvement depends on the characteristic of the original image. Figure 4 shows the result for the GOP when using 2-bits/pixel data for ME. The enhancement rate was set to $\alpha = 1$. From the results, we can see the effect of the edge enhancement in every frame.

Figure 5 shows the result using the edge-enhanced image with different enhancement rates using linear quantization. Here, the results are for $\alpha = 0.05, 0.2, 0.5$, and 1. It is shown that the effect of edge enhancement depends on α. When using lower bits/pixel such as 1 or 2 bits/pixel, we may select a larger α. In these cases, more power is reduced. In contrast, when using 3 or 4 bits/pixel, smaller α is recommended and they have better image quality. We should note that when the selection of α is too small, such as $\alpha = 0.05$, the effect of edge enhancement decreases so that the amount of improvement on the image quality decreases, too.

Now, let us consider the effectiveness of the proposed method from a different view point. We use the ratio as the reference here, which is given by

$$R = \frac{N}{H \times V/(16 \times 16)} \times 100[\%] \tag{11}$$

where N is the number of macro blocks which the motion vector has predicted, H and V are the horizontal and vertical sizes of a frame, respectively. The video sequences we used here are "football," "flowergarden," and "mobile & calendar." The results are shown in Table 3. In the conventional method, it becomes difficult to predict the motion vector when using low-bit images and some motion vectors can not be predicted even though the motion vector originally exists. By using edge enhancement, it makes it easy to predict more motion vectors compared to the conventional method when lower bits/pixel data are used. But, when the edge enhancement is strong, prediction error occurs and motion vectors are predicted for the macro blocks for which motion vectors do not originally exist. This causes the image degradation.

Figure 6 shows the percentage of the correct motion vectors for frames no. 3 and no. 6 in the "football" video sequence. Note that we define the term correct-motion vector as the one that is predicted with 8-bit/pixel-image data. We used the full-search block-matching algorithm with the accuracy of one pixel. The enhancement rate used was $\alpha = 0.5$. As the results show, the proposed method is effective when using lower bits/pixel. We can say that this tendency is the same as those shown in Fig. 3. We consider that this small difference between Figs. 3 and 6 is caused by quantization and rate control used in the MPEG-2 codec.

Table 3 The percentage of motion compensated macro blocks in frame no. 3 [%].

	bit per pixel	linear quantization without edge enhancement	linear quantization with edge enhancement
football	1	91.4	100
	2	94.1	100
	4	99.4	100
	8	100	100
flower-garden	1	80.8	88.6
	2	85.9	95.8
	4	95.8	100
	8	98.9	100
mobile & calendar	1	88.3	95.5
	2	95.9	97.7
	4	98.0	99.4
	8	98.9	99.7

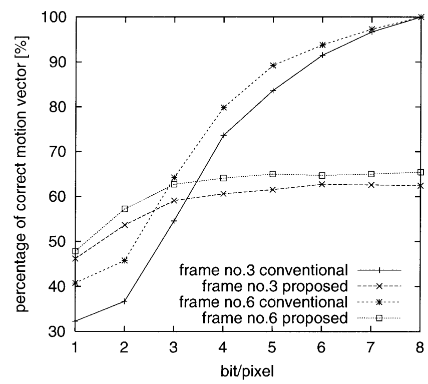


Fig. 6 The percentage of the correct motion vector[%]. ("linear" is the conventional method and "linear+edge" is the proposed method using edge-enhancement, respectively)

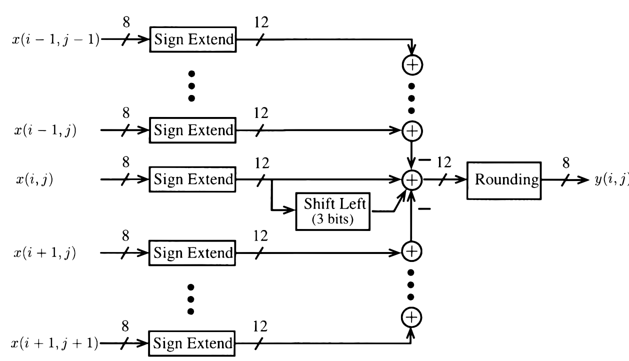


Fig. 7 The block diagram of the edge enhancement. ($x(i, j)$ is the pixel value of the input image and $y(i, j)$ is the pixel value of the edge enhanced image, respectively)

5.2 Power Estimation

To estimate the power consumption, we used the low-bit motion estimator [9], [10] and the architecture for edge enhancement designed by VHDL as an edge-triggered synchronous system. The block diagram of the edge enhancement is shown in Fig. 7. In this simu-

Table 4 Switching power estimation for ME [mW]. ("with edge enhancement" is the proposed method using edge enhancement and "without edge enhancement" is the conventional method)

with edge enhancement				without edge enhancement
1 bit/pixel	2 bits/pixel	4 bits/pixel	8 bits/pixel	8 bits/pixel
46.2264 (27.03%)	66.7584 (39.03%)	105.2792 (61.55%)	179.7849 (105.11%)	171.0414 (100%)

Table 5 Switching power estimation for edge enhancement and median cut quantization [11] [mW]. (Bit width is 1bit/pixel for median cut quantization)

edge enhancement	median cut quantization
0.0317	87.3

lation, we use linear quantization and the edge enhancement rate was set to $\alpha = 1$. The design was synthesized using the Synopsys design tools Ver. 1998.02 [14] with the linear model of the standard cell library EXDLIB provided by VDEC for 0.5-μm CMOS technology [15]. The environments are as follows:

- No driving cell is set to CLK. Inverters are set to other inputs as driving cells.
- An inverter is set to each output as the load.
- The operating condition is set to "WCCOM."
- The wire load is set to "20 × 20."

"WCCOM" stands for Worst Case COMmercial. This is the worst operating condition for commercial use which is defined in EXDLIB.

The constraints are as follows:

- CLK is constrained to $t_{clk} = 80$ [nsec]
- Area is not constrained.

Switching power is the major component of the overall power consumption, and for the research reported in this paper we estimated only the switching power. We annotate the switching information from the input image (1 bit/pixel, 2 bits/pixel, 4 bits/pixel, 8 bits/pixel), which was calculated using an ME software model. The results are shown in Table 4. This shows that using the low-bit images is very effective for saving power. The power of the proposed method using 8 bits/pixel has been increased because of the following two reasons. One is that the proposed method consumes power by performing the edge enhancement. Second is that the switching probability of the input image data has changed and this too increases power consumption. Since the improvement of the image quality is about 2 bits/pixel, we can select a 2 bits lower bits/pixel to obtain the same image quality for the proposed method. Thus, the power for the ME can be saved.

Table 5 is the switching power estimation result of edge enhancement and the median cut quantizer [11]. It is shown that the power used edge enhancement is extremely small compared to the power used for ME. Furthermore, the power required for edge enhancement is smaller than that for median cut quantization. This edge enhancement operation is independent from the quantization bit width, thus, it is also suitable for implementing an MPEG encoder with power scalability [9], [10].

6. Conclusion

We proposed a method for improving the image quality of low-bit motion estimation (ME) when using linear quantization. By using an edge-enhanced image for quantization, we can enhance the characteristics of the original image and pass them on to the low-bit image. This increases the accuracy of the ME and improves the image quality. We used a PSNR comparison of the encoded image to illustrate the effectiveness of our method, and we also estimated the power consumption. For future work, we would like to study the method of determining α and combining the method proposed in this paper with other algorithms.

Acknowledgement

The design estimation in this study has been derived by the VLSI design tools licenced via the VLSI Design and Education Center (VDEC), the University of Tokyo. We used the CMOS standard cell libraries developed in the VDEC program [15]. We would like to appreciate their cooperation.

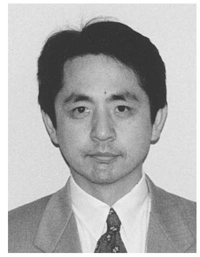
References

[1] B. Furht, J. Greenberg, and R. Westwater, Motion Estimation Algorithms for Video Compression, Kluwer Academic Publishers, 1997.
[2] H. Kiya, J. Furukawa, and Y. Noguchi, "Block matching motion estimation based on median cut quantization for MPEG video," IEICE Trans. Fundamentals, vol.E82-A, no.6, pp.899–904, June 1999.
[3] S. Lee, J.-M. Kim, and S.-I. Chae, "New motion estimation algorithm using adaptively quantized low bit-resolution image and its VLSI architecture for MPEG2 video encoding," IEEE Trans. Circuits & Syst. for Video Technol., vol.8, pp.734–744, Oct. 1998.
[4] S. Lee and S.-I. Chae, "New motion estimation algorithm and its block-matching criteria using low-resolution quantization," Proc. ITC-CSCC, vol.1, pp.175–178, 1998.
[5] 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 Processing, pp.496–499, Wash.D.C, 1995.
[6] B. Natarajan, B. Vasudev, and K. Konstantinides, "Low-complexity algorithm for motion estimation and architecture for block-based motion estimation via one-bit transform," IEEE ICASSP'96, pp.3244–3247, Atlanta, 1996.

[7] M. Kurosaki, S. Muramatsu, and H. Kiya, "A motion estimation method with two images of less gray level," IEICE Trans., vol.J83-A, no.12, pp.1434–1440, Dec. 2000.
[8] X. Song, T. Chiang, X. Lee, and Y.-Q. Zhang, "New fast binary pyramid motion estimation for MPEG2 and HDTV encoding," IEEE Trans. Circuits & Syst. Video Technol., vol.10, pp.1015–1028, Oct. 2000.
[9] A. Takagi, S. Muramatsu, and H. Kiya, "Motion estimation with power scalability and its VHDL model," IEICE Trans. Fundamentals, vol.E83-A, no.8, pp.1608–1613, Aug. 2000.
[10] A. Takagi, S. Muramatsu, and H. Kiya, "Motion estimation with power scalability and its VHDL model," IEEE Int. Conf. Image Proc., Vancouver, Sept. 2000.
[11] S. Muramatsu, H. Kiya, and A. Yamada, "A bit-operation algorithm of the median-cut quantization and its hardware architecture," IEICE Trans. Fundamentals, vol.E83-A, no.2, pp.320–328, Feb. 2000.
[12] L. Jiang, K. Ito, and H. Kunieda, "Bits truncation adaptive pyramid algorithm for motion estimation of MPEG2," IEICE Trans. Fundamentals, vol.E80-A, no.8, pp.1438–1445, Aug. 1997.
[13] MPEG Software Simulation Group, "Mpeg-2 encoder/decoder version 1.2," [Online] Available WWW: http://www.mpeg.org/MPEG/MSSG.
[14] Synopsys, Inc., 700 East Middlefield Road, Mountain View, CA 94042 USA, Synopsys online documentation version 1998.02, 1998.
[15] K. Shiomi, K. Okino, T. Kawasaki, T. Ishihara, and H. Yasuura, "Development of a standard cell library for vdec," IEICE Technical Report, CAS97-31,VLD97-31, DSP97-46, 1997.

Ayuko Takagi was born in Tokyo, Japan, on March 15, 1976. She received the B.E. and M.E. degrees in electronics and information engineering from Tokyo Metropolitan University in 1998, and 2000. She is currently a candidate for the D.E. degree at Tokyo Metropolitan University. Her research interest is in image processing.

Kiyoshi Nishikawa received the B.E., the M.E., and the D.E. degrees in electrical engineering from Tokyo Metropolitan University in 1990, 1992 and 1996 respectively. From 1992 to 1993, he was at the Computer Systems Laboratory, Nippon Steel Corp. as a researcher. In 1993, he joined Tokyo Metropolitan University, where he is currently a Lecturer in the Department of Electrical Engineering. His research interest includes the adaptive signal processing and digital communications. He is a member of IEEE SP, CAS, Communications, and Computer Societies.



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 Professor of Electrical Engineering, Graduate School of Engineering. He was a visiting researcher of the University of Sydney in Australia from Oct. 1995 to March 1996. 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. of IEICE and served as an Associate Editor of Trans. on Signal Processing of IEEE from 1998 to 2000. He is a senior Member of the IEEE, and a member of the Image Electronics Engineers of Japan and the Institute of Television Engineers of Japan.