IEICE TRANS. ELECTRON., VOL.E87–C, NO.4 APRIL 2004

PAPER *Special Section on Low-Power System LSI, IP and Related Technologies*

VLSI-Oriented Motion Estimation Using a Steepest Descent Method in Mobile Video Coding

Masayuki MIYAMA[†a)], **Junichi MIYAKOSHI**[†], **Kousuke IMAMURA**[†], **Hideo HASHIMOTO**[†], ***and* Masahiko YOSHIMOTO**[†], ***Members***

SUMMARY This paper describes a VLSI-oriented motion estimation algorithm using a steepest descent method (SDM) applied to MPEG-4 visual communication with a mobile terminal. The SDM algorithm is optimized for QCIF or CIF resolution video and VLSI implementation. The SDM combined with a subblock search method is developed to enhance picture quality. Simulation results show that a mean PSNR drop of the SDM algorithm processing QCIF 15 fps resolution video in comparison with a full search algorithm is −0.17 dB. Power consumption of a VLSI based on the SDM algorithm assuming 0.18 μm CMOS technology is estimated at 2 mW. The VLSI attains higher picture quality than that based on the other fast motion estimation algorithm, and is applicable to mobile video applications.

key words: *MPEG, motion estimation, gradient based method, steepest descent method, low power, VLSI*

1. Introduction

1.1 Background and Objectives

A mobile terminal by which people can visually communicate with others continues to gain popularity. A low power and high quality video terminal is a key to spreading the visual communication. To produce an ultra low power and high quality MPEG-4 video codec in the terminal, a highly efficient motion estimation processor is essential.

The motion estimator with a conventional full search (FS) shares more than 70% of the total computational complexity in the codec. The FS requires about 200 MOPS computation power for QCIF 15 fps motion estimation. Power Consumption of a 0.18 μm motion estimation processor using the method is about 20 mW. This power consumption is prohibitively large for an IP core in the mobile terminal.

A low power motion estimator with a diamond search algorithm (Cote) has been already reported [1]. Unfortunately, the quality of a predicted picture for high motion video is degraded because of a local minimum problem. An algorithm implemented in a VLSI that yields high quality video with low power consumption is expected.

This paper describes a gradient-based motion estimation algorithm using a steepest descent method for MPEG-4 video encoding. A VLSI based on the algorithm will execute motion estimation of QCIF 15 fps video with power consumption at 1 mW, and give higher quality video than that based on the other fast motion estimation algorithm.

1.2 Motion Estimation Algorithm

In video encoding, the motion estimation is a process to compare an encoding block (template block) in a current frame with blocks in a search range (search window) of a previous frame and find the block that has the smallest error in the search window. Here, a block is a rectangular area consisting of adjacent pixels in a frame. The criterion of a distortion function is usually a mean square error (MSE) or a mean absolute error (MAE) of a block.

A motion vector obtained by the motion estimation is encoded along with the difference between the previous block and the template block. Assuming that the difference is small, the number of bits to code them is also small. A precise motion estimation is essential to obtain high visual quality at the same bit rate.

FS (Full Search) algorithm is well known as the motion estimation algorithm. The FS algorithm evaluates all points in the search range and select the point that has the smallest prediction error. The FS algorithm always find the minimum point in the search range, but the computational complexity is extremely huge. The computational complexity required by the FS algorithm for QCIF resolution video is

$$(16*16)*2*(32*32)*((176*144)/(16*16))*15 \\ = 779MOPS.$$

Two operations to calculate sum of absolute differences (SAD), H:[−16,+15]/V:[−16,+15] of the search range, 176× 144 pixels resolution, 15 fps of the frame rate are assumed in the equation. The computational complexity for search range [−8,7] pel is 220 MOPS.

TSS (Three Step Search) algorithm is the most popular one as a fast motion estimation algorithm. The TSS evaluates all points obtained by a 4:1 subsampling in the horizontal and vertical direction of the search window (Step1.). Then 8 points surrounding the minimum point in the previous step are evaluated (Step2.). The distance between the search point and the center point is 2 pixels. Then 8 points surrounding the minimum point in the previous step are evaluated (Step3.). The distance between the search point and the center point is 1 pixel this time. The minimum point in the final step is a solution. Computational complexity required by the TSS is

Manuscript received September 6, 2003.
Manuscript revised December 25, 2003.

[†]The authors are with the Faculty of Engineering, Kanazawa University, Kanazawa-shi, 920-8667 Japan.
a) E-mail: miyama@t.kanazawa-u.ac.jp

$$(16*16)*2*((32*32)/(4*4)+8+8)$$
$$*((176*144)/(16*16))*15 = 61MOPS.$$

1.3 Gradient Based Methods

Gradient based methods are known as faster motion estimation algorithms than the TSS algorithm. The gradient based methods select an initial motion vector. The next search point is selected according to a gradient of the distortion function. OTS(One at a Time Search) [2], BBGS(Block-Based Gradient descent Search) [3], Cote(5 region diamond search) [4], 1DGDS (1-Dimensional Gradient Descent Search) [5], SDM(gradient based method using a Steepest Descent Method) [6] are known as gradient based methods.

Table 1 summarizes characteristics of the gradient based algorithms. The OTS algorithm repeats 1-dimensional search horizontally and vertically by turns until a minimum is found. The initial point to start searching is the original point (0 vector).

The BBGS algorithm repeats evaluating 8 points surrounding the minimum point in the previous step until the center point is the minimum. The initial point is the original point.

The Cote algorithm repeats evaluating 4 points surrounding the minimum point in the previous step until the center point is the minimum. The initial point is calculated by a median of predicted motion vectors. The search pattern resembles a diamond shape, so it is called 5 region diamond search.

The 1DGDS algorithm repeats 1-dimensional search horizontally, vertically and diagonally (45, 135 degrees) by turns until the minimum is found. The initial point is selected among the five points indicated by the four predicted vectors and 0 vector. The point with the best motion estimation among the five points is the initial point. The initial direction is the one of the four directions that is the closest one to the direction from the originai point to the initial point.

The SDM algorithm adopts a steepest descent method to the motion estimation. The initial point is selected among the four points indicated by the three predicted vectors and 0 vector. The point with the best motion estimation among the four points is the initial point. The search direction is calculated by the differential coefficients.

The gradient based method requires scene-adaptive computational complexity. The complexity is quite low and does not depend on the search window size. The method also has a drawback that tends to fall into a local minimum. To overcome the problem, the SDM algorithm introduces an initial point selection, a hierarchical search method and a lump search method. The SDM algorithm with these additional methods will achieve high picture quality and low computational complexity simultaneously.

2. SDM Algorithm Optimization

The SDM algorithm is optimized for QCIF or CIF resolution video and VLSI implementation. This section describes the SDM algorithm, optimization techniques and simulation results. It also describes the SDM algorithm combined with a subblock search method to enhance picture quality.

2.1 SDM Algorithm

The SDM algorithm adopts a steepest descent method to the motion estimation. Figure 1 shows an example of the distortion function over the search area for the SDM algorithm. The criterion of the function is the mean square error of a macro block (MB) indicated by a motion vector. The next search starts toward a direction that produces the steepest gradient of the function. The vector with the minimum function value over the search area is the solution to the procedure.

Technical terms are defined here to describe the SDM algorithm. "Template buffer" (TB) is a memory that stores a MB pixel data in a current frame. "Search Window Buffer" (SW) is a memory that stores pixel data in the previous frame. Brightness of the pixel that is located in TB(i,j) is described as $TB_{i,j}$. Brightness of the pixel that is located in

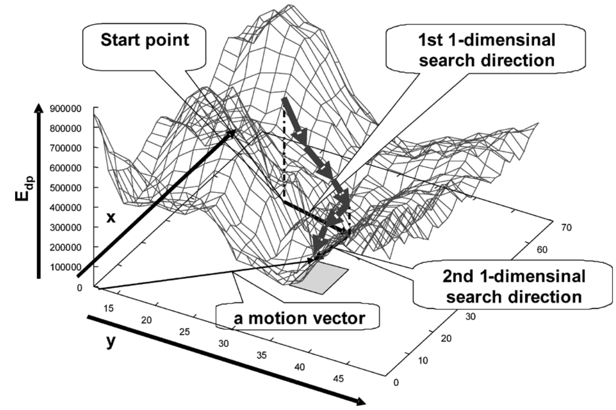


Fig. 1 Distortion function over the search area.

Table 1 Characteristics of gradient based methods.

Algorithm	Initial Vector	Search Direction	Search Dimension	Step Size	Distance Criteria
OTS	the original point	conjugate (2 directions)	1-D	1	MAE
BBGS	the original point	one of 8 surrounding points	2-D	1	MAE
Cote	median of predicted MVs	one of 4 surrounding points	2-D	1	MAE
1DGDS	one of pred. MVs and orig.	conjugate (4 directions)	1-D	2,1(variable)	MSE
SDM	one of pred. MVs and orig.	by differential coefficients	1-D	1	MSE

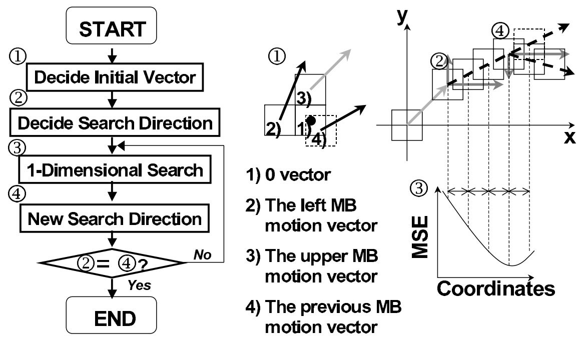


Fig. 2 SDM algorithm.

Fig. 3 Sample pictures.

SW(i,j) is described as $SW_{i,j}$. The search vector is described as (x, y). The flowchart of the SDM algorithm is shown in Fig. 2. The SDM algorithm is described as follows:

Step1. Decide an initial vector

Calculate MSE for the following four vectors. Start searching from the vector that has the smallest MSE among them.

1. 0 vector
2. The left MB motion vector
3. The upper MB motion vector
4. The motion vector of the MB that is located in the same position of the previous frame

Here, a displaced prediction difference DPD and a distortion function E_{dp} are defined as:

$$DPD(x+i, y+j) = TB_{i,j} - SW_{x+i,y+j} \quad (1)$$

$$E_{dp}(x, y) = \sum_i \sum_j [DPD(x+i, y+j)]^2. \quad (2)$$

Step2. Decide a search direction

Calculate x and y differential coefficients of the distortion function at the point indicated by the initial vector. Differential coefficients of DPD approximate to the brightness gradient between adjacent pixels.

$$\frac{\partial E_{dp}}{\partial x} = 2 \sum_i \sum_j DPD \frac{\partial DPD}{\partial x} \quad (3)$$

$$\frac{\partial DPD}{\partial x} = -(SW_{x+i+1,y+j} - SW_{x+i-1,y+j})/2 \quad (4)$$

$$\frac{\partial E_{dp}}{\partial y} = 2 \sum_i \sum_j DPD \frac{\partial DPD}{\partial y} \quad (5)$$

$$\frac{\partial DPD}{\partial y} = -(SW_{x+i,y+j+1} - SW_{x+i,y+j-1})/2 \quad (6)$$

$$\tan\theta = \frac{\frac{\partial E_{dp}}{\partial y}}{\frac{\partial E_{dp}}{\partial x}} \quad (7)$$

Step3. One dimensional search

- Search vectors toward the direction corresponding to the angle θ with step width λ.
- The λ is 1 pixel.
- Continue to search vectors until MSE increase.
- The vector whose MSE is minimum is a temporary solution.

Step4. Decide to repeat or not

- Calculate differential coefficients and new direction θ' at the point obtained in Step3..
- If θ does not equal to θ', then go to Step3., and search in the new direction corresponding to the angle θ'.
- If θ equals to θ', finish the procedure. The latest temporary solution is taken as the final solution.

The SDM algorithm introduces a hierarchical search method and a lump search method not to fall into a local minimum. The hierarchical search method generates multiresolution images. It predicts a large-scale motion vector in a coarse resolution layer and to refine the vector in a finer resolution layer. The lump search method evaluates the fixed number of points in a search direction regardless of the MSE increase. The algorithm evaluates 8 half-pel points surrounding the minimum point in the previous integer-pel search.

2.2 Optimization for QCIF and CIF Resolution Video

The SDM algorithm is optimized for QCIF and CIF resolution video about the following items:

- search range minimization
- the number of hierarchies
- repeat number minimization for 1-dimensional search
- search points minimization in a lump search

Search range should be minimized because Search Window (SW) RAM accounts for a significant amount of power consumption. The minimum search range that maintains picture quality within a mean PSNR drop of −0.1 dB is obtained by simulation. The simulator is not an MPEG encoder. It executes motion estimation only. Simulation conditions are summarized as:

- sample picture (Fig. 3)
  - Salesman (sale)
  - Susie (ssie)

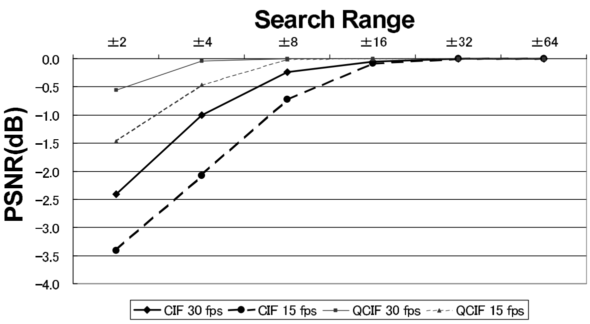


Fig. 4 Search range minimization.

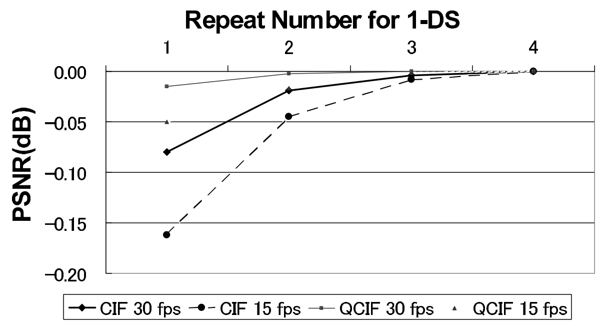


Fig. 6 Repeat number minimization for 1-DS.

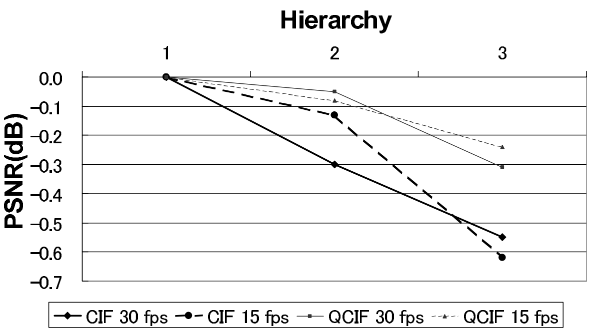


Fig. 5 Number of hierarchies.

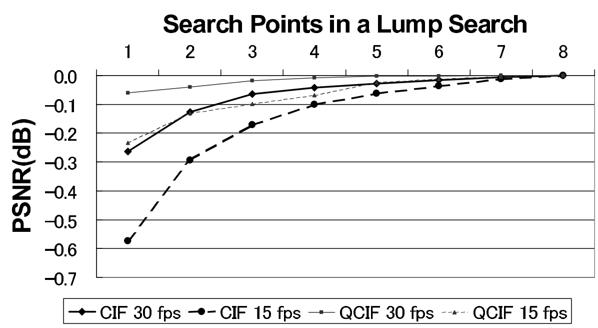


Fig. 7 Search points minimization in a lump search.

  – Mobile and Calender (mbcl)
  – Flower Garden (flow)
  – Bus (bus1)

- resolution: QCIF(176x144), CIF(352x288)
- number of frames: 75 (15 fps), 150 (30 fps)
- frame rate: 15 fps, 30 fps
- forward prediction
- half pel prediction

Figure 4 shows the relation between the search range and predicted picture quality. It represents that the minimum search range degrading the quality within −0.1 dB PSNR in regard to all cases is [−16:15.5] pixels.

The optimum number of hierarchies is obtained by simulation. The simulated number of hierarchies is 1 (SDM_mb_h1), 2(SDM_mb_h2) and 3(SDM_mb_h3). The initial vector calculations are executed in a layer from which the search starts. The simulation results are summarized in Fig. 5. They indicate that SDM_mb_h1 produces the best predicted picture quality for all sequences. The simulation results represent that QCIF and CIF resolution video does not require a hierarchical search method to enhance the quality.

It has been confirmed that the hierarchical method has a good effect on the motion estimation using a CCIR601 picture [6]. The CCIR601 is an interlaced format that is constructed by even and odd fields. The surface of the distortion function applied to the interlaced picture is like waves in accordance with even and odd fields. The search in a higher layer avoids falling into the local minimum this time. The hierarchical method for CIF and QCIF has no effect because they are non-interlaced formats.

The SDM_mb_h1 yields the best picture quality even though the lowest computational complexity. The SDM_mb_h1 algorithm does not require extra memory space storing the hierarchical image nor circuits to generate low resolution images when implemented in a VLSI.

The number of cycles to estimate a motion vector for a macroblock is usually fixed in an MPEG hardware encoder because a unit of pipelined processing is a macroblock. In such a situation, it is required to decide the upper limit number of search points, or to stop estimation at the time limit. This study takes the former approach.

The minimum number to repeat 1-dimensional search and the minimum number of points in a lump search are investigated on the condition that the picture quality is maintained within a mean PSNR drop of −0.1 dB in regard to all cases. The simulation results are summarized in Fig. 6 and Fig. 7. The results show the number to repeat 1-dimensional search is 2, and the number of points in a lump search is 4.

2.3 Optimization for VLSI Implementation

The SDM algorithm is optimized for VLSI implementation about the following items.

- differential coefficients calculation
- search direction rounding

The differential coefficients calculation requires adjacent pixels to the edge of a reference block. These extra pixels result in a complicated design of a VLSI datapath. The calculations for the block edge are modified as:

$$\frac{\partial DPD(x+0,y+j)}{\partial x} = -(SW_{x+1,y+j} - SW_{x+0,y+j})/2 \quad (8)$$

$$\frac{\partial DPD(x+15,y+j)}{\partial x} = -(SW_{x+15,y+j} - SW_{x+14,y+j})/2 \quad (9)$$

$$\frac{\partial DPD(x+i,y+0)}{\partial y} = -(SW_{x+i,y+1} - SW_{x+i,y+0})/2 \quad (10)$$

$$\frac{\partial DPD(x+i,y+15)}{\partial y} = -(SW_{x+i,y+15} - SW_{x+i,y+14})/2. \quad (11)$$

It is more appropriate to divide DPD by 1 in the above equations because the distance between pixels is 1. The dedicated hardware for the differential coefficient has to choose 1 or 2 as the divisor in the case. It makes the hardware more complicated, so the divisor is fixed at 2.

The search direction is rounded in 8 directions to simplify the address generation. The rounding calculations can be simplified by a boundary definition illustrated in Fig. 8. By the definition, the rounding calculation becomes simpler, so that the multiplication can be eliminated from the calculation and substituted by shift operation.

Figure 9 shows the relation between the rounding methods and picture quality. Modification to the equations to calculate differential coefficients causes almost no degradation of the quality because only the ratio of y to x differential coefficient is used. Rounding the search direction into one of 8 directions upgrades the quality. The horizontal and vertical motion is often the preferred motion direction. The rounding operation probably prevents the algorithm from choosing a wrong search direction. The boundary definition illustrated in Fig. 9 also upgrades the quality. The definition slightly prefers the horizontal and vertical direction to the diagonal direction. It has a good effect on the quality.

A processor with a SIMD datapath that has 16 processor elements (PEs) corresponding to 16 pixels in a row of a MB is a typical architecture to execute the SDM algorithm efficiently. The processor consists of some processing stages. They are vector generation, address generation, read operation of image data, square difference calculation, summation of the calculation results, and accumulation of the summation. The processor needs an image data cache that has 16 read ports to operate 16 PEs continuously. The pipelined processor with 16 pipelined PEs can calculate 16 pixels per 1 clock cycle. It requires 16 clock cycles per 1 MB calculation. Assuming that the number of pipeline stages is 8, the number of clock cycles to evaluate 4 points one by one is as follows: $(8 + 16) * 4 = 96$. In a lump search, the next MB can start calculating before the completion of the previous MB and the pipeline does not stall. The number of clock cycles to evaluate 4 points in the lump search is as follows: $8 + 16 * 4 = 72$. The lump search reduces clock cycles by 25% this time. Introduction of the lump search makes the pipeline more efficient.

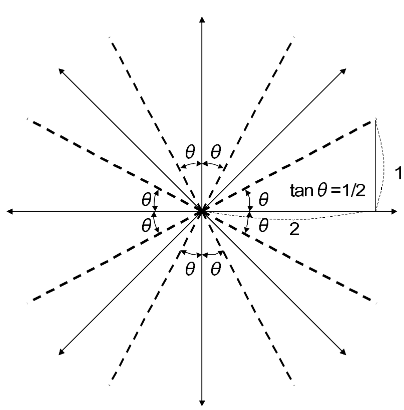


Fig. 8 Direction rounding.

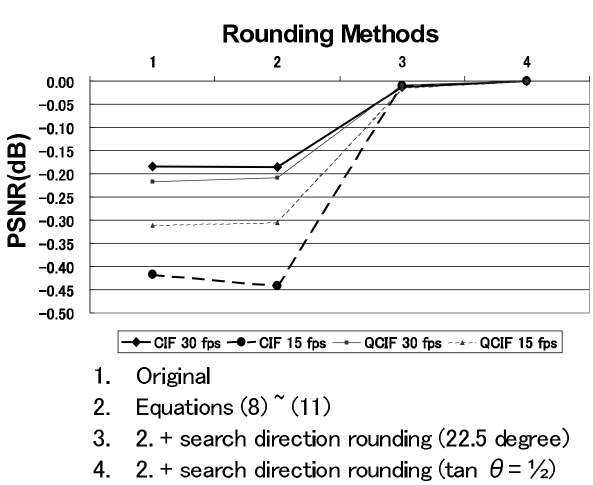


Fig. 9 Rounding methods and picture quality.

2.4 Subblock Search Method

The SDM algorithm is combined with a subblock search method (SDM-SB) to enhance picture quality. A flowchart of the SDM-SB algorithm is depicted in Fig. 10. The SB search method is described as follows:

Step1. Divide one MB into four SBs

A MB (16×16 pixels) indicated by the initial vector is divided into 4 SBs (8×8 pixels).

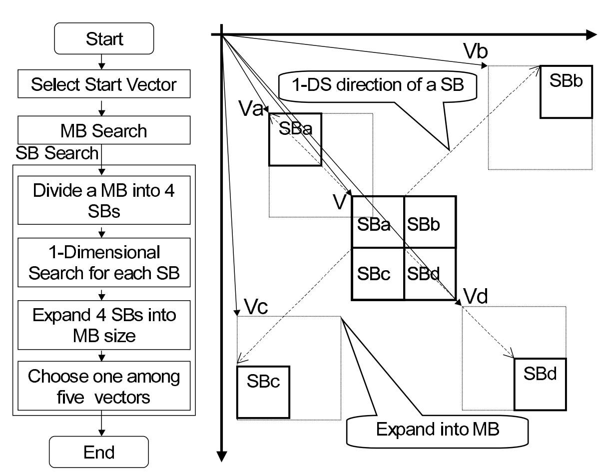


Fig. 10 Subblock search method.

Step2. One dimensional search for each SB

The differential coefficients are calculated for each SB at the point indicated by the MB initial vector. Then, the 1-DS for each SB are executed toward a search direction corresponding to the differential coefficients. The criterion is the MSE of a SB constructed by 8×8 pixels. As a result of 1-DS, four SB vectors are obtained as temporal solutions.

Step3. Expand into MB size

Four SBs indicated by SB vectors are expanded into MB size in such a way as shown in Fig. 10. The MSE of each expanded MB is calculated.

Step4. Decide a motion vector

A final motion vector is decided from 5 vectors obtained by a MB search (V) and four SB searches (Va, Vb, Vc, Vd). The SDM algorithm with the SB search decides a motion vector that indicates a MB having the smallest MSE during the MB search and the SB search. Therefore, the algorithm always attains higher or equal picture quality comparing with the original algorithm.

3. Simulation Results

The SDM algorithm and the other algorithms are simulated to analyze computational complexity and picture quality. The algorithms simulated here are as follows:

- FS
- TSS
- Cote
- SDM_mb_h1
- SDM_sb_h1 (SDM_mb_h1 combined with a subblock method).

The FS, TSS and Cote algorithms search integer-pel points first, then 8 half-pel points surrounding the minimum integer-pel point. The distortion function of the FS, TSS and Cote algorithm is a mean absolute error. The VLSI based on each algorithm usually adopts these search methods. The simulation conditions are the same as the previous section except the resolution and the frame rate are QCIF 15 fps and CIF 30 fps only, which are our targets. The parameters for the SDM algorithm are the same as the previous section except the number of points in a lump search is 3.

Tables 2 and 3 list average PSNRs of the predicted picture generated by the algorithms for each sequence. The TSS algorithm results in a mean PSNR drop of −0.52 dB for QCIF and −0.74 dB for CIF resolution video in comparison with the FS algorithm. The Cote algorithm results in a mean PSNR drop of −0.58 dB for QCIF and −0.57 dB for CIF. The TSS and Cote algorithms have sequences with a mean PSNR drop of −1 dB or less (“Bus” or “Flower Garden”).

The SDM_mb_h1 and SDM_sb_h1 algorithms have no such sequence with significantly degrading quality. They yield good results for all sequences. The SDM_mb_h1 algorithm results in a mean PSNR drop of −0.30 dB for QCIF and −0.16 dB for CIF. The SDM_sb_h1 algorithm results in a mean PSNR drop of −0.17 dB for QCIF and −0.05 dB for CIF.

Figures 11 and 12 show PSNR comparison on prediction errors for the algorithms using the sequence “Bus.” The SDM_mb_h1 and SDM_sb_h1 algorithms attains higher picture quality than the TSS and Cote algorithm, especially in the case of CIF 30 fps. The SDM_sb_h1 algorithm enhances the quality of the SDM_mb_h1 algorithm. In addition, there are frames that the PSNR of the SDM algorithm is higher than that of the FS algorithm because of the MSE error criterion for a block matching.

In this simulation, FS algorithm evaluates all points in the search range using the MAE criterion as described above. The SDM algorithm evaluates some points in the search range using the MSE criterion. The MSE is preferred

Table 2 Simulation results (QCIF 15 fps, PSNR (dB)).

algorithm	bus1	flow	mbcl	sale	ssie	mean diff.
FS	23.89	24.34	25.58	36.62	35.60	0
TSS	23.00	23.05	25.52	36.57	35.28	−0.52
Cote	21.99	23.78	25.51	36.57	35.27	−0.58
SDM_mb_h1	23.17	23.64	25.59	36.82	35.33	−0.30
SDM_sb_h1	23.34	23.91	25.59	36.86	35.49	−0.17

Table 3 Simulation results (CIF 30 fps, PSNR (dB)).

algorithm	bus1	flow	mbcl	sale	ssie	mean diff.
FS	27.55	28.08	25.71	35.98	39.52	0
TSS	26.19	27.21	25.11	35.73	38.90	−0.74
Cote	25.35	27.81	25.71	35.95	39.15	−0.57
SDM_mb_h1	27.19	27.75	25.74	36.05	39.31	−0.16
SDM_sb_h1	27.44	27.83	25.77	36.07	39.47	−0.05

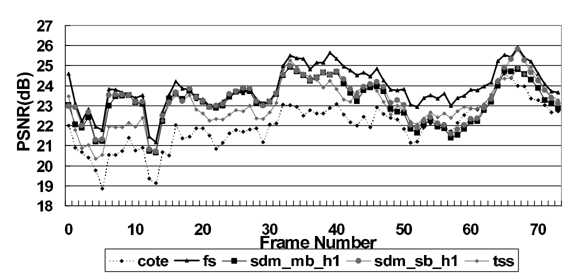


Fig. 11 Simulation results (QCIF 15 fps, Bus).

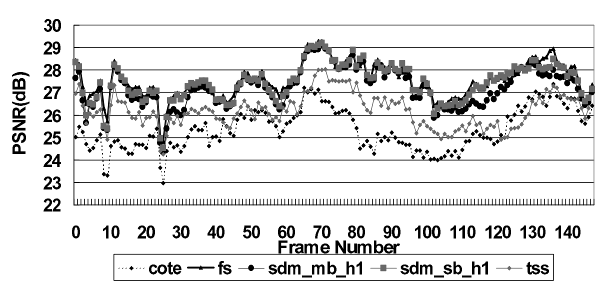


Fig. 12 Simulation results (CIF 30 fps, Bus).

than the MAE to choose the best motion vector because sum of square differences is used in the PSNR calculation. It is possible that the SDM algorithm has higher picture quality than the FS algorithm.

4. Analysis

4.1 Computational Complexity Analysis

Computational complexitiy for each algorithm is analyzed. The FS and TSS complexities can be calculated as above. The FS complexity is 779 MOPS for QCIF and 6229 MOPS for CIF resolution video. The TSS complexity is 67 MOPS for QCIF and 535 MOPS for CIF. The average and worst complexities are equal here.

The average complexity of the Cote is measured by simulation. The average complexity of the Cote is 9 MOPS for QCIF and 83 MOPS for CIF. Two operations to calculate sum of absolute differences are assumed here. The maximum number of block-matching iterations per 1 MB obtained by simulation is 72 for QCIF and 88 for CIF. The worst complexity of the Cote is 55 MOPS for QCIF and 535 MOPS for CIF.

The average complexity of the SDM is also measured by simulation. The average complexity of the SDM_mb_h1 is 15 MOPS for QCIF and 131 MOPS for CIF. The average complexity of the SDM_sb_h1 is 27 MOPS at QCIF and 268 MOPS for CIF. The worst complexity of the SDM can be calculated from the maximum number of points to evaluate. The number is fixed in the algorithm. The worst complexity of the SDM_mb_h1 is 27 MOPS for QCIF and 213 MOPS for CIF. The worst complexity of the SDM_sb_h1 is 53 MOPS for QCIF and 423 MOPS for CIF. One operation for an addition, subtraction, and multiplication is assumed here.

Figures 13 and 14 illustrate the relation between the complexity and PSNR. They represent that the SDM algorithm attains both higher picture quality and lower complexity than the TSS algorithm. The SDM algorithm also attains higher picture quality than the Cote algorithm, and the average complexity is a little bit higher than that of the Cote algorithm. The worst complexity is lower than that of the Cote.

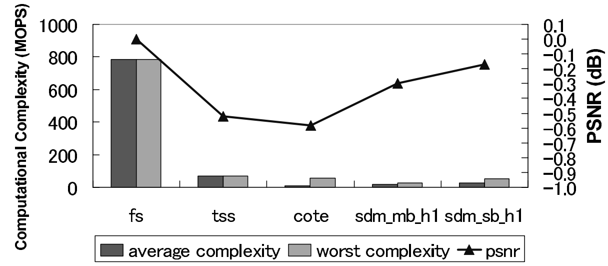


Fig. 13 ME algorithms comparison (QCIF 15 fps).

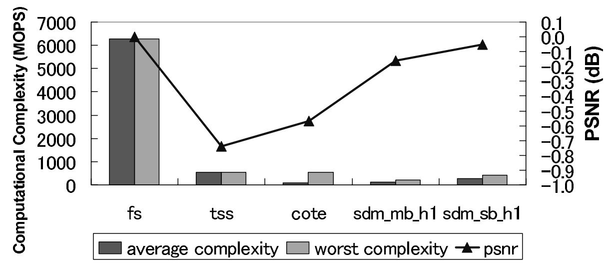


Fig. 14 ME algorithms comparison (CIF 30 fps).

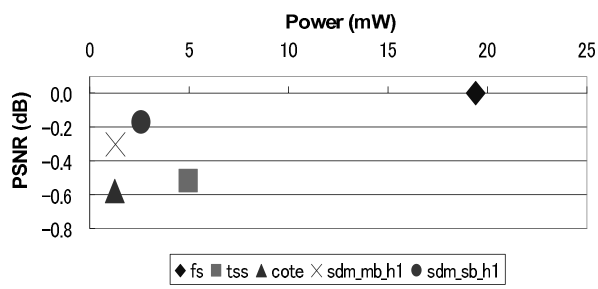


Fig. 15 ME algorithms comparison (QCIF 15 fps).

4.2 Power Consumption Analysis

Power consumption of a VLSI based on each algorithm is analyzed next. It is assumed that the VLSI is fabricated by 0.18 μm CMOS process technology and the power supply voltage is 1.8 V. Power consumption can be derived from that of the existing VLSI with modifications according to the assumption. Modifications to operating frequency, capacitance and supply voltage are necessary to calculate the power consumption. Search range, picture resolution and frame rate have to be considered in the modification to the operating frequency. Search range and process technology have to be considered in the modification to the capacitance. Power consumption of a VLSI based on the FS algorithm is derived from that of [7]. The power consumption is estimated at 19 mW for QCIF and 155 mW for CIF resolution video. Power consumption of a VLSI based on the TSS algorithm is also derived from that of [7]. The power consumption is estimated at 5 mW for QCIF and 40 mW for CIF resolution video. Power consumption of a VLSI based on the Cote algorithm is derived from that of [1]. The power consumption is estimated at 1 mW for QCIF and 10 mW for CIF resolution video. Power consumption of a VLSI based on the SDM algorithm is derived from [8]. The power consumption of a VLSI based on the SDM_mb_h1 algorithm is estimated at 1 mW for QCIF and 10 mW for CIF resolution video. The power consumption of a VLSI based on the SDM_sb_h1 algorithm is estimated at 2 mW for QCIF and 20 mW for CIF resolution video.

Figures 15 and 16 show the relation between power consumption and picture quality of a VLSI based on each algorithm. They represent that the SDM algorithm attains higher picture qualiy than the TSS algorithm, and the VLSI based on the SDM algorithm consumes lower power than that based on the TSS algorithm. The SDM algorithm also

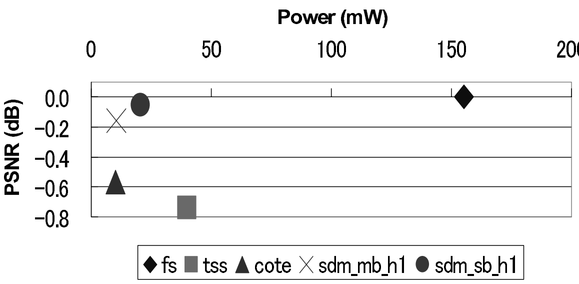


Fig. 16 ME algorithms comparison (CIF 30 fps).

attains higher picture quality than the Cote algorithm, and the VLSI based on the SDM algorithm consumes almost equal or a little bit higher power than that based on the Cote algorithm.

5. Conclusion

A highly efficient motion estimation algorithm is essential to produce a low power MPEG-4 video codec with superior visual quality. A VLSI-oriented motion estimation algorithm using a steepest descent method (SDM) is studied for this purpose. The SDM algorithm is optimized for QCIF or CIF resolution video and VLSI implementation. The SDM combined with a subblock search method is developed to enhance picture quality. Simulation results show that a mean PSNR drop of the SDM algorithm processing QCIF 15 fps resolution video in comparison with the FS algorithm is −0.17 dB. Power consumption of a VLSI based on the SDM algorithm assuming 0.18 μm CMOS process technology is estimated at 2 mW. The VLSI outperforms others based on the TSS or Cote algorithm from a view point of picture quality, and is applicable to mobile video applications. The VLSI using 0.18 μm process is now under development.

Acknowledgement

This study has been supported by STARC (Semiconductor Technology Academic Research Center).

References

[1] H. Nakayama, T. Yoshitake, H. Komazaki, Y. Watanabe, H. Araki, K. Morioka, J. Li, L. Peilin, S. Lee, H. Kubosawa, and Y. Otobe, "An MPEG-4 video LSI with an error-resilient codec core based on a fast motion estimation," Proc. ISSCC 2002, 22-2, 2002.

[2] R. Srinivasan and K.R. Rao, "Predictive coding based on efficient motion estimation," IEEE Trans. Commun., vol.COM-33, no.8, pp.888–896, Aug. 1985.

[3] L.-D. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. Circuits Syst. Video Technol., vol.6, no.4, pp.419–422, Aug. 1996.

[4] P. Kuhn, Algorithms, complexity analysis and VLSI architectures for MPEG-4 motion estimation, Kluwer Academic Publishers, 1999.

[5] O.T.-C. Chen, "Motion estimation using a one-dimensional gradient descent search," IEEE Trans. Circuits Syst. Video Technol., vol.10, no.4, pp.608–616, June 2000.

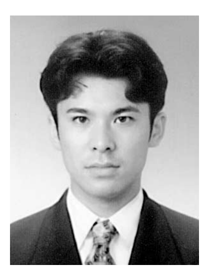
[6] M. Takabayashi, K. Imamura, and H. Hashimoto, "A fast motion vector detection based on gradient method," IEICE Technical Report, IE2001-74, Sept. 2001.

[7] R.S. Richmond II, and D.S. Ha, "A low-power motion estimation block for low bit-rate wireless video," ISLPED, pp.60–63, Huntington Beach, CA, Aug. 2001.

[8] M. Miyama, O. Tooyama, N. Takamatsu, T. Kodake, K. Nakamura, A. Kato, J. Miyakoshi, K. Imamura, H. Hashimoto, S. Komatsu, M. Yagi, M. Morimoto, K. Taki, and M. Yoshimoto, "An ultra low power motion estimation processor for MPEG2 HDTV resolution video," IEICE Trans. Electron., vol.E86-C, no.4, pp.561–569, April 2003.

Masayuki Miyama was born on March 26, 1966. He received the B.S. degree in computer science from University of Tsukuba in 1988. He Joined PFU Ltd. in 1988. He received the M.S. degree in computer science from Japan Advanced Institute of Science and Technology in 1995. He joined Innotech Co. in 1996. He is a research assistant in the Department of Electrical and Electronic Engineering at Kanazawa University. His present research focus is low power design technique for multimedia VLSI.

Junichi Miyakoshi was born on February 22, 1980, in Niigata Prefecture, Japan. He received B.E. degree in electrical and information engineering from Kanazawa University, Ishikawa, Japan, in 2002. He is currently a master's student at Kanazawa University. His research interests include system VLSI design and implementaion of multimedia communication system.



Kousuke Imamura received the B.S., M.S. and Dr. Eng. degrees in Electrical Engineering and Computer Science in 1995, 1997 and 2000, respectively, and all from Nagasaki University. He has been a research assistant of Information and Systems Engineering, Kanazawa University. His research interests are high efficiency image coding and image processing.

Hideo Hashimoto received the B.S., M.S. and Dr. Eng. degrees in Electronic Engineering in 1968, 1970 and 1975, respectively, all from Osaka University. He joined Electrical Communication Laboratories of Nippon Telegraph and Telephone Corporation (NTT) in 1975. Since 1993, he has been a professor of Information and Systems Engineering, Kanazawa University. His research interests are video coding, moving object segmentation and visual communication.

Masahiko Yoshimoto was born in Japan, on January 25, 1953. He received the B.S. degree in electronic engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1975, and M.S. degree in electronic engineering from Nagoya University, Nagoya, Japan, in 1977. He received Ph.D. degrees in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. He joined the LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan, in April 1977. From 1978 to1983 he had been engaged in the design of NMOS and CMOS static RAM including a 64K full CMOS RAM with divided-word-line structure. Since 1984 he had been involved in the research and development of a digital NTSC decoder LSI with adaptive filtering called VSP, an image compression DSP called DISP, a 100 MHz DCT processor, MPEG2 video encoder/decoder LSIs, 3-D graphics processor, multimedia ULSI systems for the digital broadcasting and the digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology and so on. Since 2000, he has been a professor of Dept. of Electrical & Electronic System Engineering in Kanazawa University, Japan. His current activity is focused on the research and development of multimedia and ubiquitous media VLSI systems including an ultra low power image compression processor and a low power wireless interface circuit. He holds on 70 registered patents. He has served on the program committee of the IEEE International Solid State Circuit Conference from 1991 to 1993. Also he served as Guest Editor for special issues on Low-Power System LSI, IP and Related Technologies of IEICE Transactions in 2004. He received the R&D100 awards from the R&D magazine for the development of the DISP and the development of the realtime MPEG2 video encoder chipset in 1990 and 1996, respectively.