OPTIMIZATION OF DECISION-TIMING FOR EARLY TERMINATION OF SSDA-BASED BLOCK MATCHING

Atsushi Hatabu, Takashi Miyazaki, and Ichiro Kuroda

Multimedia Research Laboratories, NEC Corporation
{a-hatabu@bc, miyazaki@cw, kuroda@ah}.jp.nec.com

ABSTRACT

This paper describes an analysis-based method for optimizing the timing of decisions regarding early termination of block matching (BM) in the application of a successive similarity detection algorithm (SSDA). Although the SSDA reduces BM computational costs, making decisions to terminate BM or not consumes additional processor cycles. Here, total costs, including cycles for decisions, are formulated through use of a decision interval, processor-dependent cost factors, and a function which gives probabilities of BM termination. The optimal decision interval is derived by minimizing the cost function. Experiments on MPEG-4 video coding show the proposed method facilitates comparative evaluation of the computational costs of SSDA-based BM for various processors.

1. INTRODUCTION

Block matching (BM) is widely used in such image applications as motion estimation (ME) for MPEG video coding and template matching for image recognition. A number of fast algorithms have been proposed for use in BM because of the need for repeated calculation of the matching error between an input image block and individual candidate blocks in a reference image. The two-step search [1], for example, reduces the number of candidate blocks by dividing the search process into two steps, while the use of sub-sampled images decreases the cost of matching error calculations.

The sequence similarity detection algorithm (SSDA) [2] is also used in many video processing systems to speedup BM, sometimes being combined with the two-step search and/or searches using sub-sampled images. It stops computation of matching error for one candidate block and immediately begins computation for the next candidate block when the partial sum for the first block exceeds a pre-determined threshold, thus reducing computation costs. Improved calculations of the threshold [3, 4], re-ordering of pixels in each block [3, 4], and re-ordering of candidate blocks (e.g., the spiral search [5, 6]) have all been proposed as ways to accelerate the SSDA process.

Another way of improving SSDA would be to optimize the timing by which decisions to terminate or not are made, as these decisions also consume processor cycles. While attempts at such optimization have been mostly limited to trial-and-error approaches, one significant analysis-based solution has recently been provided for use in ME in MPEG-4 video coding [7], with the optimal decision interval formulated by means of a function which represents probabilities of early termination, the cost required for one calculation, and the cost for one decision [8].

In this paper, this solution is reformulated for more general application, and use of numerical analysis to approximate the cost of BM computation provides a more accurate expression of the optimal decision interval. Experimental results on MPEG-4 video codec [8] show that the advantages of this approach can be obtained for a variety of SSDA-based BM and on a variety of processors.

2. FORMULATION OF PROBLEM

2.1. Early Termination in SSDA-based Block Matching

Let H and V be the width and height of the block handled in BM, and let N be $H \times V$. Define the matching error $D(h, v)$ between the current image block and the candidate image block, whose position is indicated by (h, v), as

$$D(h, v) = \sum_{i=0}^{H-1} \sum_{j=0}^{V-1} \|s_{\mathrm{r}}(h+i, v+j) - s_{\mathrm{c}}(i, j)\|, \quad (1)$$

where $s_{\mathrm{c}}(i, j)$ and $s_{\mathrm{r}}(h+i, v+j)$ respectively denote the luminance values of the relevant pixels in the current image and the reference image. The function to measure matching error is denoted by $\|\cdot\|$, which is often expressed as an absolute value or a square. Assume that the goal of the BM described here is to find the minimizer of $D(h, v)$.

Define the n-th partial summation of matching error $D(n)$ $(0 \leq n \leq N)$ as

$$D(n) = \sum_{k=0}^{n-1} \|s_{\mathrm{r}}(h+i_k, v+j_k) - s_{\mathrm{c}}(i_k, j_k)\|, \quad (2)$$

where (i_k, j_k) indicates the pixel whose error is being calculated, $(k+1)$-th. $D(n)$ increases monotonically because $\|\cdot\|$ takes a non-negative value. To reduce the computational cost of BM, the SSDA terminates calculations for $D(h, v)$ early if the obtained value of $D(n)$ satisfies the condition

$$D(n) \geq \mathrm{thresh}(n), \quad (3)$$

where $\mathrm{thresh}(n)$ denotes the threshold for termination. If Eq. (3) is frequently satisfied for smaller values of n, the SSDA will effectively reduce the cost of BM, which requires repeated calculation.

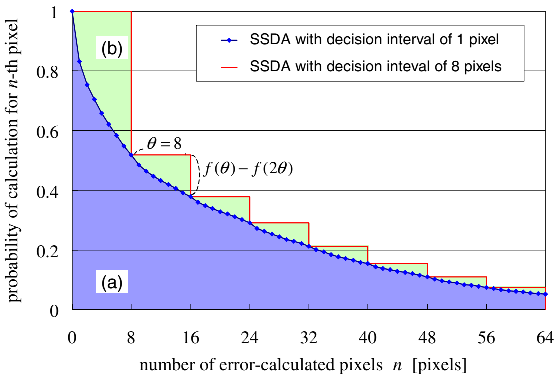


Fig. 1. Example of calculation probability function $f(n)$ in video coding

To estimate the performance of the SSDA, suppose that current images and reference images are generated on the basis of a probability model. This allows function $f(n)$ to be defined as

$$f(n) = \Pr\{D(n) < \mathrm{thresh}(n)\}, \tag{4}$$

where $f(n)$ represents the probability that Eq. (3) is not satisfied, or the probabilities that it is necessary to calculate matching error for the n-th pixel. Figure 1 shows an example of $f(n)$.

Obtaining $f(n)$ values for a given SSDA enables the computational load of the BM algorithm to be estimated and the optimal timing of decisions for early termination to be determined. Here, $f(n)$ depends on the method by which $\mathrm{thresh}(n)$ is determined, on the error measurement function $\|\cdot\|$, on the ordering of pixels (i_k, j_k), on the ordering of candidate blocks (h, v), etc. A rapid decrease in $f(n)$ values indicates that the given SSDA effectively reduces the cost of BM.

2.2. Optimizing Interval for Decisions in order to Minimize Calculation Costs

To evaluate the total cost for BM, including the cost of decisions regarding early termination, cost C is defined as

$$C = c_1 T_1 + c_2 T_2, \tag{5}$$

where T_1 denotes the average number of pixels whose errors are calculated before termination, T_2 denotes the average number of decisions as to whether to terminate or not, and c_1 and c_2 are processor-dependent weight parameters representing the number of cycles consumed for a calculation or a decision.

To simplify our discussion about the timing of decisions regarding termination, suppose that such decisions are made when matching errors for every θ pixels have been calculated. Let θ be termed the “decision interval.” BM cost C is formulated, using function f and the decision interval θ, described below. First, since the first θ errors are calculated with a probability of $f(0) = 1$, and the second θ errors are calculated with a probability of $f(\theta)$, etc., T_1 may be written as

$$T_1 \simeq \sum_{k=0}^{\lceil N/\theta - 1 \rceil} f(k\theta)\theta. \tag{6}$$

This equation show that T_1 increases with an increasing θ (see Fig. 1, where T_1 for $\theta = 8$ is illustrated as the sum of areas (a) and (b).). Since the first decision is made with a probability of $f(0) = 1$, and the next is with a probability of $f(\theta)$, etc.,

$$T_2 = \sum_{k=0}^{\lceil N/\theta - 1 \rceil} f(k\theta) \tag{7}$$

is obtained. In contrast to T_1, T_2 decreases with an increasing θ.

Since the optimal decision interval θ_* for SSDA-based BM minimizes C, a value of θ_* can be obtained by simply calculating C using values of c_1, c_2 and $f(n)$ for each value of θ. This solution, however, does not help us understand the relationship among θ_*, c_1, c_2 and $f(n)$.

3. ANALYSIS WITH APPROXIMATION

3.1. Approximation of Calculation Costs

Assume that interval θ is significantly smaller than N, and that $f(n)$, a function of integer n, naturally expands to $f(x)$, a significantly-smooth, differentiable function of an non-negative real number. Let F denote $\int_0^N f(x)dx$, which represents area (a) in Fig. 1. By applying the trapezoidal formula, with its interval set at θ, F may be calculated as

$$F \simeq \sum_{k=0}^{N/\theta - 1} \frac{1}{2}\{f(k\theta) + f(k\theta + \theta)\}\theta \tag{8}$$

$$= \sum_{n=0}^{N/\theta - 1} f(k\theta)\theta - \frac{1}{2}\{1 - f(N)\}\theta. \tag{9}$$

Since the first term on the right side of Eq. (9) nearly equals T_1(see Eq. (6)),

$$T_1 \simeq F + \frac{1}{2}\{1 - f(N)\}\theta. \tag{10}$$

T_1 may be approximated as a linear function of θ, i.e., the interval of decisions regarding termination. Similarly, T_2 may be expressed as

$$T_2 \simeq \frac{1}{\theta}T_1 \simeq \frac{F}{\theta} + \frac{1}{2}\{1 - f(N)\}. \tag{11}$$

This is a fractional function of θ, one which decreases with an increasing θ. Consequently, the total cost for BM may be approximated as

$$C(\theta) \simeq \left(c_1 + \frac{c_2}{\theta}\right)\left[F + \frac{1}{2}\{1 - f(N)\}\theta\right]. \tag{12}$$

3.2. Optimal Decision Interval for Early Termination

To find minimizer θ_* of $C(\theta)$, suppose that θ takes a non-negative real number. Differentiating Eq. (12) with respect to θ gives Eq. (13) as

$$C'(\theta) \simeq \frac{1}{2}c_1\{1 - f(N)\} - \frac{c_2 F}{\theta^2}. \tag{13}$$

Solving $C'(\theta_*) = 0$ yields minimizer θ_* as

$$\theta_* \simeq \sqrt{\frac{c_2}{c_1}}\sqrt{\frac{2F}{1 - f(N)}}. \tag{14}$$

The minimized cost C_* is obtained as

$$C_* \simeq \left|\sqrt{c_1 F} + \sqrt{\frac{1}{2}c_2\{1 - f(N)\}}\right|^2, \tag{15}$$

where the second term in this square represents the overhead created by decisions for termination.

For all practical purposes, interval θ will take only an integer. Since C_* and θ_* can be substituted into Eq. (12) to obtain

$$C(\theta) \simeq C_* + \frac{1}{2}c_1\{1 - f(N)\}\frac{(\theta - \theta_*)^2}{\theta}, \tag{16}$$

the effective optimal value of θ will be the integer that minimizes the second term of the right side of Eq. (16).

As may be seen in Eq. (14), which gives θ_*, the interval θ should be made larger when the cost of decisions is larger or when the probabilities of early termination are smaller. Among the factors in Eq. (14) and the terms in Eq. (15), only c_1 and c_2 depend on the processor on which the SSDA is executed. This means that new optimal interval θ_* and minimum costs C_* for different processors can be estimated simply by substituting new values for c_1 and c_2.

3.3. Approximation of Higher Degree

The size of θ_* in Eq. (14) is of the order $\sqrt{N}$. Under the assumption that $f^{(k)}(x) = O(N^{-k})$, an approximation with higher accuracy can be derived by applying the Euler-Maclaurin formula to F, with its interval set at θ, as

$$\int_0^N f(x)dx = \sum_{k=0}^{N/\theta} f(k\theta)\theta - \frac{1}{2}\{f(0) - f(N)\}\theta + \frac{1}{12}\{f'(N) - f'(0)\}\theta^2 + O(N\theta^4 f^{(4)}). \tag{17}$$

In similar way to that described in Section 3.1, $C(\theta)$ may be expressed as

$$C(\theta) = \left(c_1 + \frac{c_2}{\theta}\right)\left(\alpha N + \frac{\beta}{2}\theta + \frac{\gamma\theta^2}{12N}\right) + O\left(\frac{1}{N}\right), \tag{18}$$

where $\alpha = F/N$, $\beta = f(0) - f(N)$, and $\gamma = N\{f'(N) - f'(0)\}$. Here, γ can be estimated from $f(n)$ values as

$$\gamma \simeq \frac{N^2}{N-1}[\{f(N) - f(N-1)\} - \{f(1) - f(0)\}]. \tag{19}$$

Table 1. Approximation coefficients α, β, and γ

Error evaluation function	N	α	β	γ
Mean square error (MSE)	64	0.265	0.947	8.89
Mean absolute error (MAE)	64	0.373	0.949	2.17

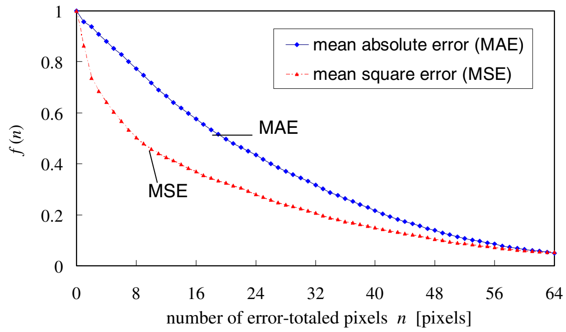


Fig. 2. $f(x)$ values obtained in ME for video coding

Minimizing Eq. (18) gives the optimal decision interval θ_* and the minimum cost C_* for BM as

$$\theta_* = \sqrt{\frac{c_2}{c_1}\frac{2\alpha N}{\beta} - \frac{c_2}{c_1}\frac{\alpha\gamma}{3\beta^2}} + O\left(\frac{1}{\sqrt{N}}\right), \tag{20}$$

$$C_* = \left|\sqrt{c_1\alpha N} + \sqrt{\frac{1}{2}c_2\beta}\right|^2 + \frac{c_2\alpha\gamma}{6\beta} + O\left(\frac{1}{\sqrt{N}}\right). \tag{21}$$

These expressions also show that θ_* and C_* derived in Section 3.2 include approximation errors of $O(1)$.

4. EXPERIMENTAL RESULTS ON VIDEO CODING

4.1. Application of Approximations

Experiments on ME of video coding were performed on an MPEG-4 video encoder [8] running on a 16-bit fixed point DSP (NEC μPD77210). This encoder utilizes a two-step search with sub-sampled images, and it employs an SSDA in the first of the two steps. For this SSDA, values of calculation probability $f(n)$ were determined on the basis of an empirical distribution of n for which $D(n) < T(n)$ was satisfied, while the encoder handled three video sequences: foreman, mobile, and flower. Figure 2 shows results for two cases: using either mean absolute error (MAE) or mean square error (MSE) for $\|\cdot\|$. MSE can be seen to reduce computational costs more than does MAE.

On the basis of $f(n)$ values, $C(\theta)$ was computed in three ways: first-degree approximation (described in sections 3.1 and 3.2), second-degree (in Section 3.3), and direct calculations (in Section 2.2). Table 1 lists the obtained coefficients for approximations, Fig. 3 shows T_1 and T_2, and Fig. 4 shows the trade-off in the total costs $C(\theta)$. In Figs. 3 and 4, dots represent the results by direct calculation, and curves express approximations of these results. Second-degree approximations were clearly more precise. Even with first-degree MSE, however, which shows the largest error, in the area of interest (around minimum C) the error is small enough for good optimization.

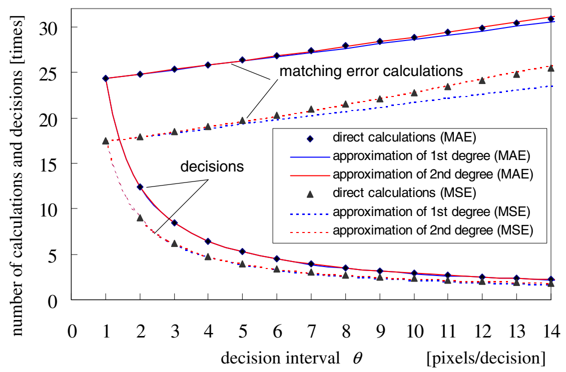
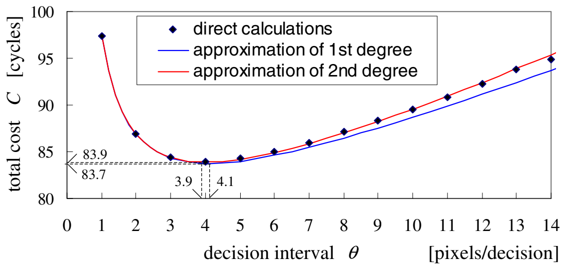
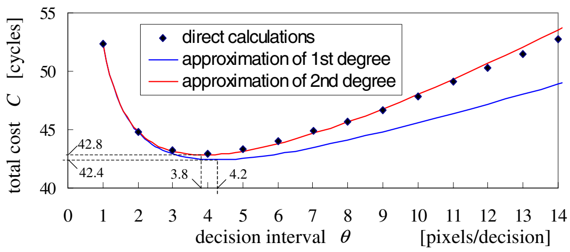


Fig. 3. Number of calculations and decisions (T_1, T_2)



(a) MAE



(b) MSE

Fig. 4. Total cost function $C(\theta)$

4.2. Estimation for Other Processors

Optimal decision interval θ_* for different processors can easily be estimated by changing coefficients c_1 and c_2. Table 2 lists θ_* and C_* for two DSPs, NEC μPD77210 and NEC μPD77050 [9], which has higher performance, as well as for an embedded CPU (ARM940T[10]). Here, c_1 represents the throughput when only matching error calculations were computed. c_2 is the increase in the number of cycles made by inserting a decision regarding termination.

With MSE, Processor 2 consumes twice as much as Processor 1 for decisions, while it offers 2.7 times higher performance Processor 1. As a result, θ_* is doubled, and C_*/c_1N, which represents the reduction of computational costs by SSDA, increases by 0.3 times. θ_* for Processor 3 is smaller than that for Processor 1 because c_1 for Processor 3 is smaller. Using MAE gives similar results to these under the proviso that θ_* with MAE is smaller than with MSE because of a larger α. Comparing minimum cost C_* for MAE and MSE shows that MSE reduces computational costs more than MAE on all evaluated processors. Note that, with respect to Processor 2, if the cost of decision making were not

Table 2. Optimal interval θ_* on various processors

(a) MAE

	Processor	c_1	c_2	θ_*	C_*	C_*/c_1N	$C_*\|_{c_2=0}$
1	μPD77210	3	1	4.00	83.9	0.44	73.0
2	μPD77050	1/2	4	17.67	24.1	0.75	12.2
3	ARM940T	7	1	2.64	185.5	0.41	170.4

(b) MSE

	Processor	c_1	c_2	θ_*	C_*	C_*/c_1N	$C_*\|_{c_2=0}$
1	μPD77210	2	1	3.79	42.8	0.33	34.9
2	μPD77050	3/4	2	7.44	21.5	0.45	13.1
3	ARM940T	6	1	2.30	116.5	0.30	104.7

considered, MAE would appear to be the better option (see $C_*|_{c_2=0}$ in Table 2), while MSE is actually the better option in terms of total cost for BM.

5. CONCLUSION

An analysis-based method has been presented for optimizing the interval of termination decisions for SSDA, a fast computation algorithm used in BM. With this method, it is possible to clarify the dependence of the optimum on the processor and the algorithm. Experiments on ME for MPEG-4 video coding has shown the effectiveness of the analysis for comparative evaluation of computational costs for SSDA as applied to a variety of algorithms and on various processors.

6. REFERENCES

[1] T. Koga, et al., "Motion Compensated Interframe Coding for Video Conferencing," in *Proc. Nat. Telecommun. Conf.*, pp. G5.3.1-5.3.5, 1981.

[2] D. I. Barnea and H. F. Silverman, "A Class of Algorithm for Fast Digital Image Registration," *IEEE Trans. Comput.*, vol. 21, C-21, no. 2, pp.179–186, Feb. 1972.

[3] J. Cooper, et al., "Early Jump-Out Corner Detectors," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 15, no. 8, pp. 823-828, 1993.

[4] H.-C. Huang, et al., "Adaptive Early Jump-Out Technique for Fast Motion Estimation," in *Proc. ICPR'96*, vol. 2, pp. 864-868, 1996.

[5] Telenor Research, "TMN (H.263) Encoder/Decoder, Version 2.0," June 1997.

[6] T. Moriyoshi, et al., "Real-Time Software Video Codec with a Fast Adaptive Motion Vector Search," *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 29, no. 3, Nov. 2001.

[7] ISO/IEC 14496-2:1999, "Information technology - Coding of audio-visual objects - Part 2: Visual," 1999.

[8] A. Hatabu, et al. "QVGA/CIF Resolution MPEG-4 Video Codec based on a Low-power and General-purpose DSP," in *Proc. of IEEE Signal Processing Systems (SiPS)*, pp. 15-20, Oct. 2002.

[9] T. Kumura, et al., "VLIW DSP for Mobile Applications," *IEEE Signal Processing Mag.*, vol. 19, no. 4, pp. 10-21, Jul. 2002.

[10] "ARM940T Technical Reference Manual," ARM Limited, http://www.arm.com/arm/TRMs/, 2000.