A FAST SEARCH ALGORITHM FOR SUB-PIXEL MOTION ESTIMATION IN H.264

Dong-kyun Park, Hyo-moon Cho
School of Electrical, University of Ulsan,7-524 san29 Muger-dong Nam-gu ,Ulsan, South Korea

Jong-Hwa Lee
School of Electrical, University of Ulsan,7-524 san29 Muger-dong Nam-gu ,Ulsan, South Korea

Keywords: SAD, H.264, Motion Estimation, Sub-pixel.

Abstract: We propose the advanced sub-pixel block matching algorithm to reduce the computational complexity by using a statistical characteristic of SAD (Sum of Absolute Difference). Generally, the probability of the minimum SAD values is highest when a searching point has one pixel distance from the reference point. Thus, we can overcome high computational complexity problem by reducing the searching area. The main concept of proposed algorithm is one of the fast searching algorithms based on TSS (Three Step Search) method. First, we find three minimal SAD points of all nine searching points on integer distance and then we obtain two minimal SAD points on 1/2-pixel between the second minimal SAD point and the first, and between the third and first, respectively. Finally, we can find matching point by comparing the SAD values among six points which are on triangle from the first minimal SAD point and two 1/2-pixel points including 1/4-pixels. The proposed algorithm in this paper needs only 14 searching points in sub-pixel mode, whereas the conventional TSS method needs total 25 searching points. Therefore, this algorithm improves the processing speed as 51%.

1 INTRODUCTION

The image compression technologies have been researched on the removing the temporal and spatial redundancies. The H.264 video standard has higher computational complexity for traditional video compression methods, though it gives an efficient compression ratio. Thus, the fast and efficient methods of H.264 compression processing are requested for the real-time processing. Specially, the researches for the motion estimation (ME) and motion compensation (MC) in H.264 are carried out actively and widely since those took more than about 70% from the whole H.264 operation [4]. Generally, motion estimation in integer unit cannot obtain accurate motion estimation because the motion of object on video image does not move in integer unit, always. Thus, sub-pixel units such as 1/2-pixel and 1/4-pixel are used in order to accurate motion estimation in advanced video compression standard. For higher accuracy, H.264 standard uses 1/4-pixel ME/MC, whereas MPEG-4 part2 uses only 1/2-pixel. As this reason, the computational complexity of H.264 is remarkably increased by contrast with H.263 or MPEG-4 part 2.

In this paper, we propose the efficient and fast sub-pixel motion-search algorithm reducing the search area and a computational complexity. To evaluate our proposed algorithm, we compared with the test result of JM11.0.

2 MOTION ESTIMATION ALGORITHMS

2.1 Error Value and SAD Characteristic

The SAD is widely used for block-match since it is simple and easy computation method to evaluate the motion estimate without multiplication and division. Its equation is shown in eq. (1).

Park D., Cho H. and Lee J. (2007).
A FAST SEARCH ALGORITHM FOR SUB-PIXEL MOTION ESTIMATION IN H.264.
In *Proceedings of the Second International Conference on Signal Processing and Multimedia Applications*, pages 135-138
DOI: 10.5220/0002139801350138

$$SAD = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \left| c_{ij} - r_{ij} \right| \tag{1}$$

where, N x N is the block size, and c_{ij} and r_{ij} are the sample image signal at position $(i, j)$of the current frame image and the reference frame image, respectively.

$$\begin{aligned} I(x_s, y_s) &= I(x, y)(x+1-x_s)(y+1-y_s) \\ &+ I(x+1, y)(x_s - x)(y+1-y_s) \\ &+ I(x, y+1)(x+1-x_s)(y_s - y) \\ &+ I(x+1, y+1)(x_s - x)(y_s - y) \end{aligned} \tag{2}$$

$I(x, y)$ represents the brightness at integer-pixel (x, y) or sub-pixel (x_s, y_s).

Fig. 1 shows the SAD-error distribution in the range of ± 1 from the reference point. The SAD error value is increased as far off from the reference as shown in Fig. 1. The fact that The SAD error-surface graph as shown Fig. 1 has high probability at quarter-pixel location is known from Table 1 which is obtained by experiments. Therefore, the 8 scores within ±1-pixel locationis have higher value and we could assume it include sub-pixel motion vector.

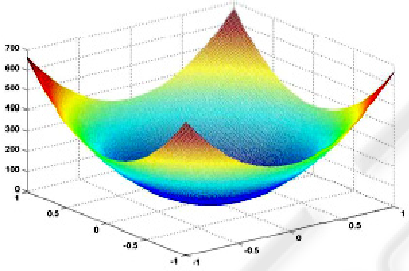

Figure 1: A typical SAD error surface.

Table 1 shows the property of SAD error-surface for several sequences. The findings of experiment indicated that more than 96% had a parabolic distribution of adjacent SAD values around the smallest SAD values. Therefore, pixel point which has the smallest SAD in integer pixel can be found using the parabolic distribution between neighboring pixels of SAD error surface. Then, the pixel point with the smallest SAD in integer pixel can be found using the distribution of SAD values after computing the SAD of values around pixels within the range of 1 based on that position. On the basis of that position, SAD of neighboring pixels within the range of 1 can be computed. Using the distribution property of computed SAD values, pixel position which has the smallest SAD in half pixel and quarter pixel can be forecasted.

Table 1: Probability of finding the optimal quarter-pixel location.

Sequence	# of blocks	# of single valley blocks	Ratio
Claire	48807	47140	96.58%
News	29601	28908	97.66%
Salesman	44352	44106	99.45%
Trevor	14751	14199	96.26%

2.2 Ordinary Motion Estimation

Ordinary motion estimation used FS (Full Search). FS is the algorithm to search all $(2R+1)^2$ number of candidates. FS method has problems in applying to real time video clip compression system because heavy computation load is required, though it enables the most optimal motion vector by searching for all positions within the search range. Therefore, a lot of high speed algorithms have been proposed to resolve those problems.

2.3 Three Step Search

This algorithm is most widely known in its three step form, the 'three step search (TSS)', but it can be carried out with other numbers of steps. For a search window of $\pm(2^N-1)$ pixels, the TSS algorithm is as follows;

1. Search location (0,0)
2. Set S = $(2^N - 1)$
3. Search eight locations $+/-S$ pixels around (0,0)
4. From the nine locations searched so far, pick the location with the smallest SAD and make this the new search origin.
5. Set S = S/2
6. Repeat stages 3-5 until S=1.

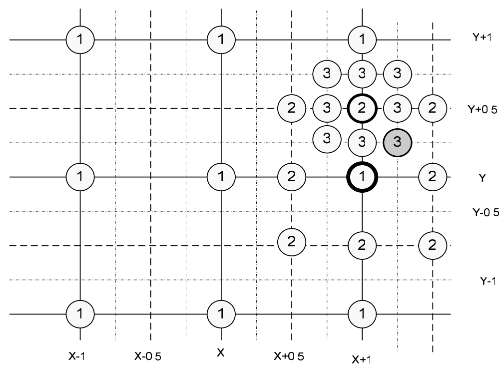

Figure 2: Three Step Search (TSS).

Figure 2 illustrates the procedure for a search window of $\pm$ 1.75(i.e. N=3). The first 'step' involves searching location (0, 0) and eight locations ± 1 pixels around the origin. The second 'step' searches ± 0.5 pixel around the best match from the first step(highlighted in bold) and the third step searches ± 0.25 pixels around the best match from the second step(again highlighted). The best match from this third step in chosen as the result of the search algorithm. With a search window of $\pm$ 1.75, three repetitions (steps) are required to find the best match. A total of (9+8+8) = 25, search comparisons are required for the TSS, compared with (15 $\times$ 15) = 225 comparisons for the equivalent full search. In general, (8N+1) comparisons are required for a search area of $\pm(2^N - 1)$ pixels.

3 PROPOSED ALGORITHM

In this paper, the second smallest SAD and the third smallest SAD can be found in integer pixel 8 points around the point which possess the smallest SAD in integer pixel to identify the sub pixel motion vector in that direction, using distribution property of SAD error surface, as shown in Fig. 1. Therefore, 1/2-pixel and 1/4-pixel motion vector can be obtained through integer pixel SAD. In this regard, 1/2-pixel and 1/4-pixel motion vector can be obtained through the position of integer pixel SAD value.

As shown in Equation (3), it is found that both side points next to pixel have the biggest effect because of the characteristics of H.264, which interpolates 1/2-pixel with 6-tap FIR filter. Therefore, 1/2-pixel and 1/4-pixel motion estimation at a position between the second smallest SAD and the third smallest SAD where SAD was the smallest-was proposed, using the following property.

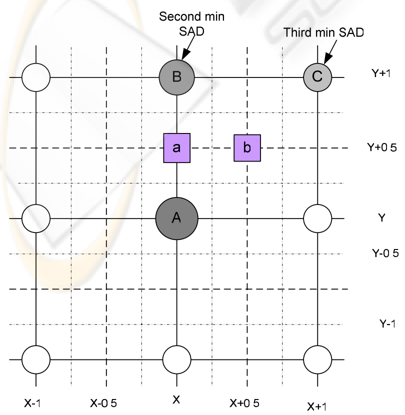

Figure 3: Proposed motion estimation in half pixel.

If SAD is minimum at (x,y), 1/2-pixel motion vector can be obtained in the following way:

Step 1: Find (x,y) with the smallest SAD in integer pixel, and find the point which has the second smallest SAD and the third smallest SAD in the neighboring integer pixel point within

Step 2: Find 1/2-pixel SAD point between the second smallest SAD point and the third smallest SAD point.

Considering that the 1/2-pixel searching point has the motion estimation in the direction from the minimum SAD to the second largest and the third largest SAD, two searching points are enough.

In addition, 1/4-pixel searching point can be found using 1/2-pixel searching point, as shown in Fig. 4.

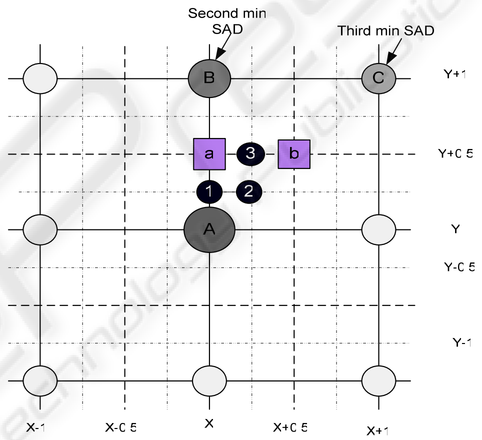

Figure 4: Proposed motion estimation in quarter pixel.

Find 1/4-pixel SAD value can be found by using two points and minimum integer SAD found in 1/2-pixel as shown in Fig 4. Equation (3) is an example in which 1/4-pixel motion estimation in Fig 4 was used.

$$\begin{aligned} 1 &= \partial(a + A)/2 \\ 2 &= \partial(a + b)/2 \\ 3 &= \partial(A + b)/2 \end{aligned} \tag{3}$$

Therefore, the strength of the proposed method is that the computation was made simpler and the processing time was shortened because only 2 searching points were sought in 1/2-pixel and 3 searching points in 1/4-pixel, unlike the previous way which required 8 searching points for 1/2-pixel and 1/4-pixel.

4 IMPLEMENTATION BASED ON THE PROPOSED ALGORITHM

The test was conducted using JM11.0 open source. Fast search algorithm embodied in JM11.0 were used exactly for integer pixel, and a comparison was made between the method which reduces the number of searching points for half and quarter pixel and the method which searches for in whole under existing method, and the result was summarized. The proposed algorithm is considered to embody the function nearly on a par with PSNR, compared with previous algorithm, to evaluate the image quality.

A comparison was made between the method which searches 5 points under existing method and the proposed method, in order to validate the efficiency of the proposed method for sub pixel motion estimation.

Table2: Performance comparison about propose algorithm.

	Algorithm	PSNR (dB)	Bit-rate (Kbit/s)	Encoder processing time (sec)	Sub-pixel processing time (sec)
foreman	JM 11.0	38.7	85.2	48.2	7.68
	Proposed	37.6	80.6	36.3	3.70
akiyo	JM 11.0	40.5	44.4	44.2	7.91
	Proposed	39.2	37.6	33.3	3.86
news	JM 11.0	40.2	87.8	46.9	8.12
	Proposed	39.8	79.2	36.5	3.96
table tennis	JM 11.0	39.2	87.8	47.8	8.49
	Proposed	39.0	77.2	36.2	4.99
Mobile	JM 11.0	38.2	105.1	49.9	8.52
	Proposed	37.8	91.2	38.0	4.23

5 CONCLUSIONS

This paper proposed the method of motion estimation in less complicated sub-pixel to ensure the efficient motion estimation process which comprises the largest portion of calculation in H.264, the next generation image compression codec with superb compression efficiency. The proposed method is to forecast the direction which has less error range in sub pixel to search for motion vector, using the error surface property and pixel interpolation property of SAD. Searching only in the direction forecast on the basis of the difference in SAD value, the searching point was remarkably reduced to two for half pixel and five for quarter pixel respectively, unlike existing method which required 8 searching points.

The result of test based on the proposed method indicated that the time that it took to calculate was reduced by approximately 24% for encoder processing and by approximately 51% for sub pixel motion estimation, compared to the function of existing JM 11.0. Therefore, the efficiency of proposed method for motion estimation was validated.

ACKNOWLEDGEMENTS

The work was supported by the Korea Research Foundation Grant funded by the Korean Government and 2 level BK21 (EVERDEC (e-Vehicle Research & human Resource Development Center)) in MOE (Ministry of Education & Human Resources Development). (KRF-2006-D00292)

REFERENCES

Cheng Du, Yun He, and Junli Zheng, "PPHS: A Parabolic Prediction-Based, Fast Half-Pixel Search Algorithm for Very Low Bit-Rate Moving-Picture Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No.6, June 2003.

Bo Zhou, Jian Chen, "A Fast Two-Step Search Algorithm for Half-Pixel Motion Estimation", *ICECS 2003, Proceedings of the 2003 10th IEEE International Conference*, Vol 2, pp 14-17, Dec 2003.

H. M. Wong, O. C. Au, J. Huang, S. Zhang and W. Z. Yan, "Sub-Optimal Quarter- Pixel Inter-Prediction Algorithm (SQIA)," *ICASSP '05. IEEE International Conference on Volume 2*, pp. 921-924, March 18-23, 2005.

Iain E, G. Richardson, "Video Codec Design" pp102~ 10 5, willey

Richardson, "H.264 and MPEG-4 Video compression", Willey

W Zhenyu, J Baochen, Z Xudong, C Yu, "A New Full-pixel and Sub-pixel Motion Vector Search Algorithm for Fast Block-matching Motion Estimation", *Image and Graphics, 2004. Proceedings. Third International Conference , ICIG'04*, pp18~20, Dec 2004 .