# Adaptive Unithreshold Two-Bit Block Motion Estimation

Dongjin Jung and Jechang Jeong
Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea
Email: jdj850@gmail.com, jjeong@hanyang.ac.kr

***Abstract*—Image compression is essential technique to realize real time communication and motion estimation is the slowest part in the video processing. To reduce the complexity, one bit and two bit transform algorithms are presented past, but this algorithms are some problems, particularly when each pixel transform into binary, it increase encoding time. In this work, we improve the 2 bit transform algorithm and reduce the complexity while keeping subjective quality. When we do bit transforming, we compare the current block and previous frame's collocated block. If the block is almost same, we just calculate current block's average and variance. If it is not, we find the mean and variance separately. By doing this method, we can approximately gain 10 % complexity decrease.**

***Index Terms*—video coding, motion estimation, fast motion, bit transform, block matching**

## I. INTRODUCTION

Digital video compression is the most important part to store the video data and transmit those in many broad casting services, and the video coding standard such as MPEG-x, H.264 has been standardized for reducing the data. Motion estimation (ME) is the most efficient part in the video coding and has a 50% complexity of entire process [1].

Full search (FS) is the fundamental algorithm and used widely in ME. It has the limited search area and finds the best candidate in that area. This algorithm exhausts all search points and has a mount of complexity; however, we can select best candidate, and it is easy to realize in hardware. To resolve these problems, a lot of algorithms have been proposed. ME is classified into 3 method largely: successive elimination algorithm (SEA), partial distortion search (PDS), bit transform (BT). First, SEA removes special search point if any specific condition is satisfied [2]. It has no loss of subjective quality compared with FS. Second, PDS is the early termination algorithm by taking some sample in the candidate block and predicting entire block distortion [3], [4]. Third, BT changes 8 bit image into binary image and revises the conventional block matching to process parallelly [5]-[9].

## II. CONVENTIONAL ALGORITHMS

### *A. FS Algorithm*

FS is the most famous technique in ME and use the block distortion of current block and candidate block. While searching the entire search point, it find the best search point having the minimum sum of difference (SAD) in the limited area. It exhausts all candidate points and takes a long time; however, we can get the optimal motion vector.

$$SAD(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|f^{t}(i,j)-f^{t-1}(i+m,j+n)\right|, \quad where -s \le m,n \le s \tag{1}$$

where, f is the intensity of pixel, (i,j) means the pixel position. N-1 is the block size, and (m,n) is the translation of motion vector ,and s is the search area.

### *B. One-Bit Binary Block Matching ME*

Binary block matching ME was developed for decrease the complexity by converting 8bit pixel into a few bits. It speeds up by applying data packing; however, it may have the wrong motion vector because its distortion range from 0 to 64 in case of 1BT 8x8 block matching (FS distortion range: 0~16320).

1BT use the next kernel which is the 17x17 band-pass filter.

$$K_{i,j}=\begin{cases}\frac{1}{25} & \text{, if i,j} \in [0,4,8,12,16] \\ 0 & \text{, otherwise}\end{cases} \tag{2}$$

where, i,j is the index of x, y axis. By (2), each image is filtered, and then we make a binary image using (3)

$$B(i,j)=\begin{cases}1 & \text{,if F(i,j)} \ge \hat{F}(i,j) \\ 0 & \text{,otherwise.}\end{cases} \tag{3}$$

where F is an original image, and $\hat{F}$ is a filtered image, and B means a binary image. Above process is called 1BT. After transforming, we can calculate the number of non-matching point ( NNMP) which is the block matching criteria.

$$NNMP(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}B^{t}(i,j)\oplus B^{t-1}(i+m,j+n). \quad where -s \le m,n \le s \tag{4}$$

where, $\oplus$ is the exclusive operator.

Manuscript received October 1, 2013; revised January 11, 2014.

©2014 Engineering and Technology Publishing
doi: 10.12720/ijeee.2.1.41-44

$$K = \frac{1}{16}\begin{bmatrix}
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

Figure 1. Multiplication-free kernel

### C. Multiplication-Free 1-Bit Binary Block Matching ME

In One-bit algorithm, we cannot avoid the multiplication operation which increases the complexity of ME. To resolve this problem, we make the number of non-zeros be a power of 2 which can be realized in shift operation; however, the modified kernel generated in loss. The kernel shape is like a diamond and is shown in Fig. 1. The filter has an addition of 16 pixels is shifting the result, so it can be processed in integer operation.

### D. Two-Bit Binary Block Matching ME

Seeing (4), we can notice that the distortion of block matching limited to 64, so it has possibility to find wrong motion vector. To improve this problem, 2BT was proposed. 2BT cover the 40x40 window block by block and calculate the mean and variance of the window using next equation.

$$\begin{aligned} \mu &= E[I_{tw}] \\ \sigma^2 &= E[I_{tw}^2] - E^2[I_{tw}] \end{aligned} \tag{5}$$

where, $I_{tw}$ is the intensity of pixel in window, and E means the average. However, as the standard deviation has high computational burden, we simply approximate to this equation.

$$\sigma_a = 15 + 0.0125\sigma^2. \tag{6}$$

where, $\sigma_a$ is the approximated standard deviation. Using these values, we change the pixel into binary.

$$\begin{aligned} B_1(i,j) &= \begin{cases} 1 & \text{,if } I(i,j) \ge \mu \\ 0 & \text{,otherwise.} \end{cases} \\ B_2(i,j) &= \begin{cases} 1 & \text{,if } I(i,j) \ge \mu + \sigma_a \text{ or } I(i,j) \le \mu - \sigma_a \\ 0 & \text{,otherwise.} \end{cases} \end{aligned} \tag{7}$$

where $B_1(i,j)$ and $B_2(i,j)$ are each bit plane image. Using (7), we can get the block matching distortion.

$$\begin{aligned} NNMP(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} & \left\{ B_1^t(i,j) \oplus B_1^{t-1}(i+m, j+n) \right\} \\ & \| \left\{ B_2^t(i,j) \oplus B_2^{t-1}(i+m, j+n) \right\} \\ & where -s \le m,n \le s \end{aligned} \tag{8}$$

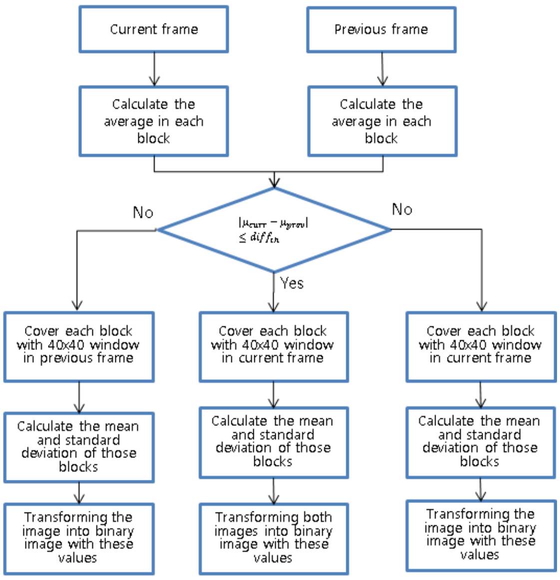


Figure 2. The flow chart of proposed algorithm.

TABLE I. EXPERIMENTAL ENVIRONMENT

| | |
|---|---|
| Software | Off Codec |
| Test Sequence | CIF (352x288) |
| Block size | 8, 16 |
| Search area | ± 8, ± 16 |
| Kernel size | 40x40, 80x80 |
| Complexity | The number of operations |

©2014 Engineering and Technology Publishing 

TABLE II. THE PSNR COMPARISON WITH CONVENTIONAL ALGORITHMS

| PSNR | FS | 2BT | Proposed | FS | 2BT | Proposed |
|---|---|---|---|---|---|---|
| COASTGUARD | 31.70 | 30.49 | 30.51 | 30.68 | 29.98 | 29.99 |
| FOREMAN | 33.07 | 30.08 | 30.35 | 32.33 | 30.22 | 30.26 |
| MOBILE | 25.87 | 24.96 | 24.98 | 25.18 | 24.83 | 24.85 |
| STEFAN | 25.48 | 24.32 | 24.35 | 24.92 | 24.35 | 24.35 |
| TABLE | 34.08 | 32.42 | 32.54 | 32.77 | 31.52 | 31.54 |
| CONTAINER | 38.43 | 37.03 | 37.72 | 38.34 | 37.97 | 37.98 |
| HALL | 36.01 | 33.96 | 34.07 | 34.83 | 33.78 | 33.88 |
| AVERAGE | 32.09 | 30.46 | 30.64 | 31.29 | 30.38 | 30.41 |

TABLE III. THE COMPLEXITY COMPARISON WITH CONVENTIONAL ALGORITHMS

| Complexity | FS | 2BT | Proposed | FS | 2BT | Proposed |
|---|---|---|---|---|---|---|
| COASTGUARD | 77163 | 74778 | 60059 | 1135827 | 716394 | 655903.2 |
| FOREMAN | 77163 | 74778 | 61494 | 1135827 | 716394 | 660972.2 |
| MOBILE | 77163 | 74778 | 60043 | 1135827 | 716394 | 655479.2 |
| STEFAN | 77163 | 74778 | 63687 | 1135827 | 716394 | 664633.5 |
| TABLE | 77163 | 74778 | 60357 | 1135827 | 716394 | 657326.6 |
| CONTAINER | 77163 | 74778 | 59467 | 1135827 | 716394 | 655230.4 |
| HALL | 77163 | 74778 | 59565 | 1135827 | 716394 | 655430.5 |
| AVERAGE | 77163 | 74778 | 60667 | 1135827 | 716394 | 657854 |

where, $||$ is the bit operator “OR”. Equation (8) is revised from (4) and consider two bit plane and increase the distortion range, so we can get the better motion vector than 1BT.

### E. Proposed Algorithm

2BT extends the distortion range to acquire more accurate motion vector than 1BT algorithms; however, it is not ignorable to process the mean and variance. Through this algorithm, we do not only increase the encoder speed, but also keep its subjective quality. The procedure of my algorithm is shown in Fig. 2. In advance, cover every block with 40x40 window each frame and simply calculate the average of each block each frame. If the difference between current and previous block’s average is in the threshold value, we just calculate the average and variance of current blocks and then transform into binary image with those values. Otherwise, we calculate separately those values.

## III. EXPERIMENTAL RESULTS

The Table I shows the experimental setting. Test sequences which have a different motion are used and the size of it is 352x288 (CIF). We experiment in the block size of 8x8, 16x16, and Depending on block size the search range is $\pm 8$, $\pm 16$ . To find the best motion vector, we used the spiral method which search from the point of (0,0) to neighboring point. For the comparison with original and reconstructed image, we adopted the PSNR measure which is used widely for the image processing. We also used the complexity method which is measured by following equation

$$Complexity = Add + Mul * 8 + Bit\_op + Com \quad (9)$$

In (9), Add means add operation and the Mul is multiplication, and Bit_op a shift operation and Com a comparison. The Table II shows the PSNR results of proposed and conventional algorithms. Compared to FS, 2BT changes the 8bit pixels into 2bit pixel, so the PSNR of 2BT dropped to 1dB. Our proposed algorithm increased to 0.2dB in PSNR.

The Table III compared the complexity of algorithms and the complexity is measured as operations. FS has a large burden because it finds every search point and calculates the distortion of the block. The 2BT improved the complexity using bit operation and reducing the data of pixels.

## IV. CONCLUSION

Our algorithm can calculate one block’s variance and average depending on the similarity between current and previous blocks, and the complexity is reduced compared with 2BT. Its trend is same as 16x16 block case. The

©2014 Engineering and Technology Publishing

proposed algorithm calculates one block's average and variance in transforming binary values if the current and past frame's block is almost same. Our algorithms show not the improvement of PSNR but also reduce the complexity of algorithm, and it can be adopted for a variety of improved 2BT algorithms.

## ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT & Future Planning), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) **(NIPA-2013-H0301-13-1011)**

## REFERENCES

[1] Z. L. He, C. Y. Tsui, K. K. Chan, and M. L. Liou, "Low-power VLSI design for motion estimation using adaptive pixel truncation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 669–678, Aug. 2000.

[2] Y. Wang and G. Tu, "Successive elimination algorithm for binary block matching motion estimation," *Electronics Letters*, vol. 36, pp. 2007-2008, Nov. 2003.

[3] C. K. Cheung and L. M. Po, "Normalized partial distortion search algorithm for block motion estimation," *IEEE Trans. CSVT*, vol. 10, pp. 417-422, Apr.2000.

[4] C. H. Cheung and L. M. Po, "Adjustable partial distortion search algorithm for fast block motion estimation," *IEEE Trans. CSVT*, vol. 13, pp. 100-110, Jan. 2003.

[5] A. Erturk and S. Erturk, "Two-bit transform for binary block motion estimation," *IEEE Trans. Circuits and Algorithms Systems for Video Tech*., vol. 15, no. 7, pp. 938-946, July 2005.

[6] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *IEEE Trans. Circuits and Systems for Video Tech*., vol. 7, no. 5, pp. 702-706, Aug. 1997.

[7] S. Ertuk, "Multiplication-free one-bit transform for low-complexity block-based motion estimation," *IEEE Signal Processing Letters*, vol. 14, no. 2, pp. 109-112, Feb. 2007.

[8] H. Lee, S. Jin, and J. Jeong, "Early termination scheme for 2BT block motion estimation," *Electronics Letters*, vol. 45, Apr. 2009.

[9] O. Urhan and Sarp Erturk, "Constrained one-bit transform for low complexity block motion estimation," *IEEE Trans. Circuits and Systems for Video Tech*., vol. 17, pp. 478-482, Apr. 2007.

**Jung** received BS degree in Dept. of wireless Communication Engineering from kwangwoon University, Korea, in 2010 and MS degree in in electronics and computer engineering from Hanyang University, Korea, in 2012. He is currently pursuing the PhD degree in electronics and computer engineering at Hanyang University. His research interests are in the fields of image processing and compression, including H.264 video coding, multi-view video coding (MVC), and scalable video coding (SVC).

**Jechang Jeong** received a BS degree in electronic engineering from Seoul National University, Korea, in 1980, an MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 1982, and a Ph.D degree in electrical engineering from the University of Michigan, Ann Arbor, in 1990. From 1982 to 1986, he was with the Korean Broadcasting System, where he helped develop teletext systems. From 1990 to 1991, he worked as a postdoctoral research associate at the University of Michigan, Ann Arbor, where he helped to develop various signal-processing algorithms. From 1991 through 1995, he was with the Samsung Electronics Company, Korea, where he was involved in the development of HDTV, digital broadcasting receivers, and other multimedia systems. Since 1995, he has conducted research at Hanyang University, Seoul, Korea. His research interests include digital signal processing, digital communication, and image and audio compression for HDTV and multimedia applications. He has published numerous technical papers. He received the Scientist of the Month award in 1998, from the Ministry of Science and Technology of Korea, and was the recipient of the 2007 IEEE Chester Sall Award and 2008 ETRI Journal Paper Award. He was also honored with a government commendation in 1998 from the Ministry of Information and Communication of Korea Information and Communication of Korea Chester Sall Award and 2008 ETRI Journal Paper Award. He was also honored with a government commendation in 1998 from the Ministry of Information and Communication of Korea.

©2014 Engineering and Technology Publishing 