EARLY TERMINATION ALGORITHM FOR ONE-BIT TRANSFORM-BASED MOTION ESTIMATION USING BINOMIAL DISTRIBUTION

Hanjin Park, Changryoul Choi, and Jechang Jeong

Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea.

ABSTRACT

One-bit transform (1BT)-based block motion estimation (ME) has been proposed to reduce the computational complexity by using Boolean exclusive-OR matching of one-bit representations of image frames. In this paper, a novel early termination algorithm for 1BT-based ME is proposed in order to decrease the calculations of the block distortion measure. Unlike the classical early termination schemes, the proposed algorithm utilizes a new approach to reduce computations. It employs the binomial distribution based on the characteristic of binary plane which is composed of only two elements: 0 and 1. Experimental results show that the proposed algorithm keeps its peak signal-to-noise ratio (PSNR) performance very close to the full search algorithm (FSA) while the computational complexity of ME is reduced considerably.

Index Terms—One-bit transform, motion estimation, binomial-distribution, early termination, fast block matching.

1. INTRODUCTION

Motion estimation (ME) plays an important role in digital video compression system because of its remarkable ability to remove temporal redundancies between successive image frames. Generally the ME carries out up to 80% of the computations in the whole video coding systems so it is ceaselessly asked to reduce certain amount of its computations.

The most widely used technique in the ME field is the block-matching algorithm (BMA) [1],[2] which divides each frame into rectangular blocks of equal size and searches for the best matching block within a search window in the previous frame. Full-search algorithm (FSA) is the most suitable method in the various BMAs to find an optimal motion vector. However, its huge computational burden always makes it infeasible for real-time implementations. So, many fast BMAs including one-bit transform (1BT)-based ME are invented to improve the computational burden of FSA.

The author’s e-mail: airjin@gmail.com, denebchoi@gmail.com, and jjeong@ece.hanyang.ac.kr. Hanjin Park is also with Samsung Electronics.

The binary BMAs such as 1BT [3] and two-bit transform (2BT) [4] are the methods of using different matching criteria to achieve reduction in computational complexity. They use Boolean exclusive-OR (XOR) operations instead of mean absolute difference (MAD) which is used in FSA.

1BT-based ME [3] transforms video sequences from a multi-bit to one-bit/pixel representations by comparing the original image frame against a multiband-pass filtered output at first and then carries out classical ME search algorithm such as FSA. A 17×17 multiband-pass filter kernel (K) is given as:

$$K(i,j) = \begin{cases} 1/25, & \text{if } i,j \in [0,4,8,12,16] \\ 0, & \text{otherwise.} \end{cases}$$  \hfill (1)

And the 1BT bit-plane of the image is created as:

$$B(i,j) = \begin{cases} 1, & \text{if } I(i,j) \geq I_p(i,j) \\ 0, & \text{otherwise.} \end{cases}$$  \hfill (2)

$I_p(i,j)$ which is obtained by applying the kernel $K$ to $I(i,j)$ is used as a kind of pixel-wise threshold, i.e. $I_p(i,j)$ represents the filtered version of the image frame $I(i,j)$. After obtaining the 1BT frame $B$, the motion vector of a block is calculated in the basis of the number of non-matching points (NNMP) measure. The NNMP is given as:

$$NNMP(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |B^i(i,j) \ominus B^{i-1}(i+m,j+n)|$$  \hfill (3)

where $(m,n)$ represents the candidate displacement, $s$ decides the search window size, $\ominus$ denotes XOR operation, and $N \times N$ is the macro block (MB) size. $B^i(i,j)$ and $B^{i-1}(i+m,j+n)$ are 1BT representations in the current and previous image frame, respectively. The candidate displacement which has the lowest NNMP value is decided to be the block motion vector.

In this paper, a novel early termination algorithm for 1BT-based ME is proposed. The basic idea of the proposed algorithm is that each MB of current and reference frame can be designed as the binomial distribution which is based on the idea of Bernoulli trial [6], since each bit-plane
consists of only two values: 0 and 1. Of course, it is assumed that bit values of 1BT bit-plane are independent for establishing the early termination criterion even if they are correlated. Using the binomial distribution of $B^t(i,j)$ and $B^{t-1}(i + m, j + n)$, the $NNMP_{EX}(m, n)$ which is the expectation of the $NNMP(m, n)$ is easily obtained by (4) instead of calculating the $NNMP(m, n)$. In the same way, the average ($\mu$) and standard-deviation ($\sigma$) which become the criterion to evaluate the probability of the $NNMP_{EX}(m, n)$ for early termination are also derived from $B^t(i,j)$ . Experimental results show that the proposed algorithm can reduce computational complexity significantly while it has similar peak signal-to-noise ratio (PSNR) performance as the FSA.

The features of binomial distribution are described for better understanding of the proposed algorithm.

If $X \sim binomial (N, p)$, then

$$EX(\mu) = Np$$

$$VarX(\sigma^2) = Np(1 - p)$$

$$Stdev(\sigma) = \sqrt{Np(1 - p)}$$

If $N$ is large enough, then the skew of distribution is not too great, we can regard the binomial distribution as the normal distribution approximately:

$$binomial (N, p) \sim normal (Np, Np(1 - p))$$

(5)

**2. PROPOSED ALGORITHM**

In this section, we derive a new early termination technique based on the binomial distribution. In order to eliminate the non-possible candidate motion vectors (CMVs), the proposed algorithm compares the $NNMP_{EX}(m, n)$ with $\mu$ and $\sigma$ before calculating true $NNMP(m, n)$. According to the result, we decide whether the MB should be discarded or calculated.

### 2.1. $NNMP_{EX}(m, n)$

As we mentioned in Section 1, each binary plane such as $B^t(i,j)$ and $B^{t-1}(i + m, j + n)$ can be designed as the binomial distribution. So the result of $B^t(i,j) \oplus B^{t-1}(i + m, j + n)$ also follows the binomial distribution. To derive the $NNMP_{EX}(m, n)$, we define two vectors: $x$ and $y$.

Let $x$ denotes a binary vector of length $N \times N$ of the current MB. The vector is defined as follows:

$$x = (x_0, x_1, \ldots, x_{255})$$

(6)

Then we define a random variable $X$ as a total number of 1’s occurrence in 256 trials which is derived from MB ($16 \times 16$); a random variable $X$ follows the binomial distribution. The features of $X$ are given as:

$$X \sim binomial \left(256, \frac{w_y(x)}{256}\right)$$

(7)

where $w_y(\cdot)$ denotes the Hamming weight which is the number of nonzero components.

In the same way we define a vector $y$ which is a binary vector of length $N \times N$ of the reference MB and a random variable $Y$. The features of $y$ and $Y$ are given as:

$$y = (y_0, y_1, \ldots, y_{255})$$

(8)

$$Y \sim binomial \left(256, \frac{w_y(y)}{256}\right).$$

(9)

In order to calculate the $NNMP_{EX}(m, n)$, let $z$ denote a binary vector of length $N \times N$ of the XOR results of $x$ and $y$. The features of $z$ and $Z$ are given as:

$$z = x \oplus y = (x_0 \oplus y_0, x_1 \oplus y_1, \ldots, x_{255} \oplus y_{255})$$

(10)

$$Z \sim binomial (256, A).$$

Note that the $NNMP_{EX}(m, n)$ is the expectation of the number of nonzero components, so $A$ should be considered the probability to be 1. Because of the characteristic of XOR operation, $x$ and $y$ must be the different value each other in order to be 1. For example, $x_0 \oplus y_0 = (0 \oplus 1) = (1 \oplus 0) = 1$. So $A$ which is the probability of becoming 1 is calculated as:

$$A = \left(\frac{256 - w_y(x)}{256}\right) \times \left(\frac{w_y(y)}{256}\right) + \left(\frac{w_y(x)}{256}\right) \times \left(\frac{256 - w_y(y)}{256}\right).$$

(11)

In terms of computation, if we have computed the probability ($w_y(\cdot)/256$) and ($256 - w_y(\cdot)))/256 according to all 256 cases which are arbitrary integer value between 0 and 256, and stored to look-up table in advance, $A$ is easily obtained by two multiplications and one addition. Thus, the expectation of random variable $Z$ which is identical to the $NNMP_{EX}(m, n)$ is obtained as:

$$NNMP_{EX} = n \times A = 256 \times A.$$
the μ and σ, let w denote a binary vector of length $N \times N$ of the XOR results of $x$ and $x'$. The features of $w$ and $W$ are given as:

$$w = x \oplus x'$$

$$W \sim \text{binomial}(256, B).$$

In the same way, $x$ and $x'$ must be a different value to each other in order to be 1. So $B$ which is the probability of becoming 1 is calculated as:

$$B = \left( \frac{256 - w_i(x)}{256} \right) \times \left( \frac{w_i(x')}{256} \right) \times \left( \frac{256 - w_i(x')}{256} \right).$$

(13)

Equation (13) can be expressed in the form

$$B = \frac{512 \times w_i(x) - 2 \times w_i(x)^2}{256^2}$$

(14)

where $w_i(x) = w_i(x')$.

In the same way the μ and σ are easily obtained by (4). It is helpful to use probability look-up table which is calculated by the number of $w_i(x)$ for reducing computational burden.

$$\mu(\text{Average}) = n \times B = 256 \times B$$

$$\sigma^2(\text{var}) = n \times B \times (1 - B)$$

$$\sigma(\text{sd}) = \sqrt{\sigma^2} \approx 15 + 0.0125 \times \sigma^2$$

(15)

where the third term of (15) is derived from [4].

After finding all the components: the $w_i(x)$, $w_i(x')$, and $B_{ij}$, the proposed algorithm’s regular procedure starts to find the best CMV. During the searching process, the proposed algorithm compares the $NNMP_{EX}(m, n)$ with μ and σ before measuring the true distortion of the $NNMP_{EX}(m, n)$. Under the assumption that the most suitable CMV is located near the μ and distributed by the normal distribution (5), the proposed algorithm’s criterion for early termination is defined as:

$$(\mu - k\sigma) \leq NNMP_{EX}(m, n) \leq (\mu + k\sigma)$$  \hspace{1cm} (16)

where $k$ is the σ’s coefficient to decide the range of the lower and upper bounds. The range of $k$ is located between 0 and 3 due to the characteristic of the standard-deviation.

Intuitively we recognize that the bigger the $k$ is, the better the PSNR performance is; the amount of calculation increase. On the other hand, if the $k$ is close to the zero, the amount of calculations decreases substantially and likewise the PSNR performance decreases. Fig. 2 which is derived from Table 1 shows the PSNR performance of ‘Akiyo’ and ‘Football’ sequences with variations of $k$. According to the result the static picture such as ‘Akiyo’ has a strong tolerance of PSNR performance in small $k$; motional picture such as ‘football’ has a weak tolerance. So we can achieve better PSNR performance using adaptive $k$ which varies according to the block activity. In order to judge the amount of motion, the $NNMP(0,0)$ is utilized as a good indicator for block activity [5]. The $NNMP(0,0)$ is given as:

$$NNMP_{(0,0)} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B_i(i, j) \oplus B'(i, j).$$

(17)

After comparing the $NNMP(0,0)$ with α and β which are lower and upper bounds of block activity respectively, the values of each MB’s $k$ are classified into three: 0.08, 0.25, and 1. At the end, if the $NNMP_{EX}(m, n)$ is not satisfied with the criterion (16), we regard the $NNMP(m, n)$ as non-possible CMV so discard this search position ($m, n$) and go to the next search position. Otherwise, the conventional FSA-1BT is applied. Fig. 1 shows overall flowchart of the proposed algorithm with adaptive $k$. 

![Fig.1. Flowchart of the proposed algorithm](image-url)
Table 1. Experimental results for PSNR(dB) and ANSP

<table>
<thead>
<tr>
<th>Test Sequence</th>
<th>FSA-1BT</th>
<th>Proposed (adaptive $k$)</th>
<th>Proposed (fixed $k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Akiyo</td>
<td>PSNR</td>
<td>42.03</td>
<td>42.05</td>
</tr>
<tr>
<td></td>
<td>ANSP</td>
<td>984</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>ANSP</td>
<td>984</td>
<td>558</td>
</tr>
<tr>
<td>Football</td>
<td>PSNR</td>
<td>22.66</td>
<td>22.62</td>
</tr>
<tr>
<td></td>
<td>ANSP</td>
<td>984</td>
<td>678</td>
</tr>
<tr>
<td>Hall</td>
<td>PSNR</td>
<td>32.51</td>
<td>32.63</td>
</tr>
<tr>
<td></td>
<td>ANSP</td>
<td>984</td>
<td>434</td>
</tr>
<tr>
<td>Mobile</td>
<td>PSNR</td>
<td>23.52</td>
<td>23.47</td>
</tr>
<tr>
<td></td>
<td>ANSP</td>
<td>984</td>
<td>424</td>
</tr>
<tr>
<td>Average</td>
<td>PSNR</td>
<td>29.17</td>
<td>29.18</td>
</tr>
<tr>
<td></td>
<td>ANSP</td>
<td>984</td>
<td>449</td>
</tr>
</tbody>
</table>

Fig.2. Normalized PSNR performance with variations of $k$

3. EXPERIMENTAL RESULTS

The simulations of the proposed algorithm were conducted in five CIF ($352 \times 288$, 100 frames) sequences: ‘Akiyo’, ‘Stefan’, ‘Football’, ‘Hall’ and ‘Mobile’. And only the luminance component was considered. The MB size and the size of search range were chosen as $16 \times 16$ and $\pm 16$, respectively. The searching process was in spiral order. We selected two measures, the average number of search positions per block (ANSP) and PSNR of the reconstructed sequences, to evaluate the performances of the proposed algorithm compared with the exhaustive search FSA-1BT. The experimental results are given in Table 1.

In terms of the fixed $k$, the proposed algorithm has a powerful tool which can control the computational complexity gain by varying the coefficient ($k$) of the current MB’s standard-deviation. As seen from the results PSNR performance is degraded slightly within maximum 0.85dB which is very close to FSA-1BT. On the other hand, the ANSP gain is improved dramatically up to 78% when the $k$ is 0.08. In terms of the adaptive $k$, although the average rate of PSNR performance is almost same but the computational gain is 54% on average over FS-1BT when $\alpha$ and $\beta$ are 15 and 30, respectively. Moreover, in some sequences such as Akiyo and Hall, the proposed algorithm outperforms the FSA-1BT.

4. CONCLUSION

This paper proposed an effective early termination scheme for 1BT-based ME to reduce the computational complexity. Using the modeling of the binomial distribution, the proposed algorithm can establish the condition for 1BT matching criterion to eliminate the non-possible CMV and attain a good computational reduction. The experimental results show that the proposed algorithm can decrease computational complexity substantially while maintaining its PSNR performance very close to the FSA-1BT. It means that the proposed algorithm is suitable for real-time implementations.

5. REFERENCES