AN FFT-BASED FULL-SEARCH BLOCK MATCHING ALGORITHM USING OVERLAP-ADD METHOD

Hidetake SASAKI, Zhen LI, and Hitoshi KIYA

Department of Information and Communication Systems, Tokyo Metropolitan University
6-6 Asahigaoka, Hino-shi, Tokyo 191–0065, Japan

ABSTRACT

One category in fast full-search block matching algorithms (BMAs) is based on the fast Fourier transformation (FFT). Conventional methods in this category must adjust the macroblock size to the search window size by zero-padding. In the methods, the memory consumption and computational complexity heavily depend on the size difference between the macroblock and the search window. Thus, we propose a novel FFT-based BMA to solve this problem. The proposed method divides the search window into multiple sub search windows to versatilely control the difference between the macroblock and the search window sizes. Simulation results show the effectiveness of the proposed method.

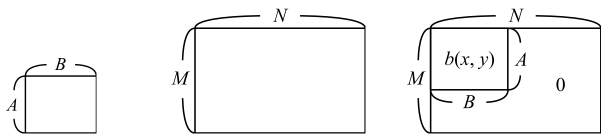
***Index Terms*—** block matching, FFT, overlap-add method, pattern recognition, motion estimation

1. INTRODUCTION

Block matching is widely used in many fields, such as pattern recognition and motion estimation. However, the direct full-search block matching algorithm (with exhaustively searches for every possible candidate in the search window to find the most similar block) imposes a heavy computational load, which makes it almost impossible to use in any applications. Therefore, there is a real need for a fast and high accuracy algorithm.

One category in previous studies bases on the fast Fourier transformation (FFT) [1–4]. Compared to the direct full-search block matching algorithm (BMA), FFT-based BMAs achieve the same accuracy, though they greatly decrease the computational load. These algorithms need to adjust the macroblock size to the search window size by zero-padding. This property arises a problem that the memory consumption and computational complexity increase as the size difference between the macroblock and the search window enlarges.

This paper proposes a novel FFT-based BMA that overcomes the above mentioned problem. The proposed method divides the search window into multiple sub search windows using the overlap-add method (OLA) [5]. This allows us to change the search window size versatilely. Theoretical analysis shows that the proposed method is about 2.0 to 3.4 times more efficient than the conventional method under the optimal conditions. Simulation results also prove the effectiveness of the proposed method.



(a) Macroblock $b(x, y)$. (b) Search window $f(x, y)$. (c) Extended signal $g_b(x, y)$.

Fig. 1. The conceptual diagram for signals.

2. PRELIMINARY

2.1. Block matching

As shown in Figs. 1 (a) and (b), let 2-D signals $b(x, y)$ and $f(x, y)$ be a macroblock and a search window, respectively. Suppose that the search window is bigger than the macroblock. That is,

$$b(x, y), \quad x = 0, 1, \ldots, A-1, \quad y = 0, 1, \ldots, B-1, \tag{1}$$

$$f(x, y), \quad x = 0, 1, \ldots, M-1, \quad y = 0, 1, \ldots, N-1, \tag{2}$$

$$A < M, \quad B < N, \quad x, y, A, B, M, N \in \mathbf{Z},$$

where $\mathbf{Z}$ denotes the set of integer numbers.

Inside the search window, there are $(N-B+1) \times (M-A+1)$ different blocks which each block is the same size as $b(x, y)$. All these different blocks are compared with $b(x, y)$ to find the most similar one. This procedure can be defined as *full-search block matching*.

An FFT-based BMA generally uses the sum of squared differences (SSD) criterion. The SSD can be expressed as follows:

$$\mathrm{SSD}_{b,f}(u, v) = \sum_{x=0}^{A-1} \sum_{y=0}^{B-1} \{f(x+u, y+v) - b(x, y)\}^2, \tag{3}$$

$$u \in [0, M-A+1], \quad v \in [0, N-B+1], \quad u, v \in \mathbf{Z}.$$

Variables u and v are shift amounts. The purpose of full-search block matching is to find (u_0, v_0) that yields the minimum matching error:

$$\mathrm{SSD}_{b,f}(u_0, v_0) = \min_{u,v}\{\mathrm{SSD}_{b,f}(u, v)\}. \tag{4}$$

2.2. FFT-based full-search block matching algorithm

Here, we describe the conventional FFT-based BMA [3]. First, by zero-padding $b(x, y)$, we can have signal $g_b(x, y)$ which is the same size as $f(x, y)$, as shown in Fig. 1(c). Then, we can rewrite Eq. (3) as

$$\mathrm{SSD}_{b,f}(u, v) = C_{g_b} - 2\mathrm{cor}_{g_b,f}(u, v) + S_{f^2}(u, v), \tag{5}$$

where,

$$C_{g_b} = \sum_{y=0}^{B-1}\sum_{x=0}^{A-1}\{b(x, y)\}^2, \tag{6}$$

$$\mathrm{cor}_{g_b,f}(u, v) = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} g_b(x, y)f(x + u, y + v), \quad \text{and} \tag{7}$$

$$S_{f^2}(u, v) = \sum_{y=0}^{B-1}\sum_{x=0}^{A-1} f^2(x + u, y + v). \tag{8}$$

Since C_{g_b}, the first term on the right in Eq. (5), is independent of (u, v), it would not affect the result of block matching. Then, (u_0, v_0) that yields the minimum matching error is calculated by

$$(u_0, v_0) = \arg\max_{u,v}\{2\mathrm{cor}_{g,f}(u, v) - S_{f^2}(u, v)\}. \tag{9}$$

$S_{f^2}(u,v)$ can efficiently be carried as the recursive summation [6, 7]. An example of the recursive summation is expressed as below, where we consider 1-D signals for simple explanation:

$$S_{f^2}(u) = \begin{cases} \sum_{k=0}^{A-1} f^2(u + k), & u = 0, \\ S_{f^2}(u-1) - f^2(u-1) + f^2(u+A-1), & u > 0. \end{cases} \tag{10}$$

Moreover, $\mathrm{cor}_{g_b,f}(u, v)$, a cross-correlation between $g_b(x, y)$ and $f(x, y)$, can be calculated by using FFT. Thus, (u_0, v_0) in Eq. (9) can easily be found.

However, if the search window is much larger than the macroblock, a huge zero-padded region would lead to an increase in the memory consumption and the computational load. To solve this problem, we propose a new FFT-based block matching algorithm in the next section that the proposed method divides the search window into multiple sub search windows.

3. PROPOSED METHOD

In this section, we first describe a cross-correlation calculation by overlap-add method, and then propose a new FFT-based BMA using the OLA-based cross-correlation calculation. Furthermore, a concurrent processing of FFT-based BMA is proposed.

3.1. Calculate a cross-correlation by OLA

Overlap-add is one of the method for calculating a block convolution [5]. We show an efficient method to calculate a cross-correlation by OLA for 1-D signals.

As shown in Fig. 2, we divide the search window $f(x)$ into L of M_1-length multiple adjoining sub search windows. By zero-padding sub search windows, we can have new signals $f_i(x)$'s which size of each signal M_2 is $M_1 + A - 1$. Subscript i denotes the index of sub search windows $(i = 1, 2, \ldots, L)$.

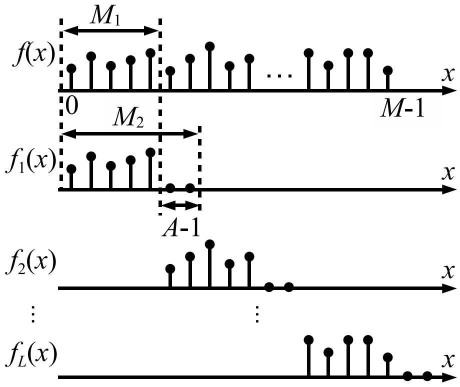


Fig. 2. The conceptual diagram for dividing a signal.

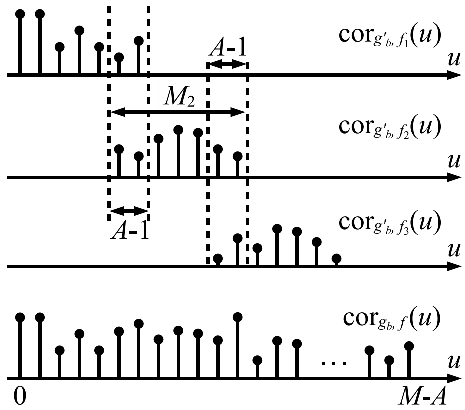


Fig. 3. The conceptual diagram for combining signals.

The cross-correlation $\mathrm{cor}_{g'_b,f_i}(u)$ between $f_i(x)$ and $g'_b(x)$ is calculated as

$$\mathrm{cor}_{g'_b,f_i}(u) = \sum_{x=0}^{M_2-1} g'_b(x)f_i(x + u), \qquad u \in [0, M_2 - 1], \tag{11}$$

where $g'_b(x)$ is a zero-padded macroblock whose length is M_2. Then, we can overlap and add all $\mathrm{cor}_{g'_b,f_i}(u)$'s together to reconstruct the cross-correlation $\mathrm{cor}_{g_b,f}(u)$ as shown in Fig. 3.

3.2. FFT-based BMA using OLA

As in the case for 1-D signals, we split the search window $f(x, y)$ into multiple adjoining sub search windows which the sub windows size is $M_1 \times N_1$. By zero-padding each sub search window and the macroblock, we can have signals, $f_i(x, y)$'s and $g'_b(x, y)$, whose sizes are all $M_2 \times N_2$. Here, M_2 is $M_1 + A - 1$ and N_2 is $N_1 + B - 1$. The cross-correlation between them can be written as,

$$\mathrm{cor}_{g'_b,f_i}(u, v) = \frac{1}{M_2N_2}\sum_{k=0}^{M_2-1}\sum_{l=0}^{N_2-1}\{\overline{G'_b(k, l)}F_i(k, l)\}W_{M_2}^{-uk}W_{N_2}^{-vl}, \tag{12}$$

$u \in [0, M_2 - 1], \quad v \in [0, N_2 - 1],$

where $G'_b(k,l)$ and $F_i(k,l)$ are the DFTs of $g'_b(x,y)$ and $f_i(x,y)$, respectively. $\overline{G'_b(k,l)}$ represents the complex conjugate of $G'_b(k,l)$ and $W_N^n = e^{-j2\pi n/N}$, where j is the square root of -1.

The cross-correlation $\mathrm{cor}_{g_b,f}(u,v)$ is reconstructed by overlapping and adding all $\mathrm{cor}_{g'_b,f_i}(u,v)$'s together. The size of the overlapped area is $(A-1)\times(B-1)$. By using this method, we can select FFT size more freely than conventional methods [1–4].

3.3. Concurrent processing of FFT-based BMA

In most cases, macroblock and search window in the block matching are both real signals, whereas the FFT approach is designed for complex signals. Therefore, we can calculate two cross-correlations at the same time.

First, we combine $f_i(x,y)$ and $f_{i+1}(x,y)$ to create a new complex signal $\hat{f}(x,y) = f_i(x,y) + jf_{i+1}(x,y)$. The DFT of $\hat{f}(x,y)$ can be written as $\hat{F}(x,y) = F_i(x,y) + jF_{i+1}(x,y)$. Therefore, the cross-correlation between $\hat{f}(x,y)$ and $g'_b(x,y)$ can be written as,

$$\begin{aligned}\mathrm{cor}_{g'_b,\hat{f}}(u,v) &= \frac{1}{M_2N_2}\sum_{k=0}^{M_2-1}\sum_{l=0}^{N_2-1}\{\overline{G'_b(k,l)}\hat{F}(k,l)\}W_{M_2}^{-uk}W_{N_2}^{-vl}\\ &= \frac{1}{M_2N_2}\sum_{k=0}^{M_2-1}\sum_{l=0}^{N_2-1}\{\overline{G'_b(k,l)}(F_i(k,l)+jF_{i+1}(k,l))\}W_{M_2}^{-uk}W_{N_2}^{-vl}\\ &= \frac{1}{M_2N_2}\sum_{k=0}^{M_2-1}\sum_{l=0}^{N_2-1}\{\overline{G'_b(k,l)}F_i(k,l)\}W_{M_2}^{-uk}W_{N_2}^{-vl}\\ &\quad + j\frac{1}{M_2N_2}\sum_{k=0}^{M_2-1}\sum_{l=0}^{N_2-1}\{\overline{G'_b(k,l)}F_{i+1}(k,l)\}W_{M_2}^{-uk}W_{N_2}^{-vl}\\ &= \mathrm{cor}_{g'_b,f_i}(u,v) + j\mathrm{cor}_{g'_b,f_{i+1}}(u,v). \end{aligned} \tag{13}$$

Separating Eq. (13) into the real and imaginary parts, we can get $\mathrm{cor}_{g'_b,f_i}(u,v)$ and $\mathrm{cor}_{g'_b,f_{i+1}}(u,v)$ at the same time. This property leads to a reduction in computational cost of the proposed FFT-based BMA. Fig. 4 illustrates the block diagram of the whole proposed method.

4. COMPUTATIONAL COMPLEXITY OF THE PROPOSED METHOD

Here we analyze the computational complexity of the proposed method. For ease of explanation, let us consider the 1-D situation. For M_2-point FFT/IFFT, $M_2\log_2 M_2$ complex additions and $(M_2/2)(\log_2 M_2 - 1)$ complex multiplications are needed. The number of FFT/IFFT operations in the proposed method is $L+1$. Thus, additions per one output, A_p, and multiplications per one output, M_p, for all FFTs/IFFTs in the proposed method can be written as,

$$A_p = \frac{M_2(3\log_2 M_2 - 1)(L+1)}{M-A+1}, \tag{14}$$
$$M_p = \frac{2M_2(\log_2 M_2 - 1)(L+1)}{M-A+1}, \tag{15}$$

respectively. Here, let r be the ratio of the zero-padding signal size to the macroblock size, i.e., $r = M_2/A$. Then, we can rewrite Eqs. (14) and (15) as,

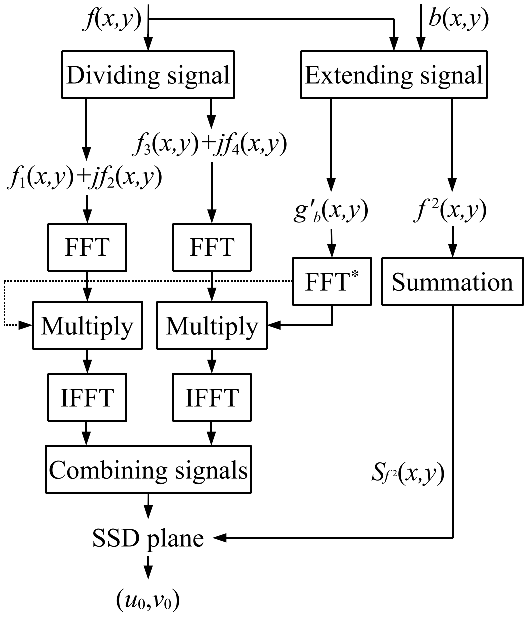


Fig. 4. The whole proposed method. (L = 4, FFT* means complex conjugate of FFT).

Table 1. The optimal r for M_p.

(M,A)	Optimal r	M_p for optimal r	M_p for non-division	Efficiency
(512,16)	2.94	14.64	49.45	3.38
(512,32)	2.97	19.80	51.09	2.58
(512,64)	2.98	27.98	54.74	1.96

$$A_p = \frac{Ar(3\log_2 Ar - 1)(L+1)}{M-A+1}, \tag{16}$$
$$M_p = \frac{2Ar(\log_2 Ar - 1)(L+1)}{M-A+1}. \tag{17}$$

Meanwhile, the relationship between the division number L and the ratio r can be expressed as,

$$L+1 = \frac{M-A+1}{Ar-A+1} + 1. \tag{18}$$

$Ar(3\log_2 Ar - 1)$ and $2Ar(\log_2 Ar - 1)$ increase with r does, whereas $L+1$ decreases with r increases. It means particular r exists which minimizes the computational complexity.

Fig. 5 illustrates M_p versus r. It explicitly shows that there is optimal r. Table 1 lists the optimal r and computational costs for several sized different macroblocks. As shown in Table 1, the proposed method is expected to improve the computational efficiency up to about 2.0–3.4 times in theory. Since A_p gives similar results to M_p in the computational efficiency, the results for A_p are omitted here.

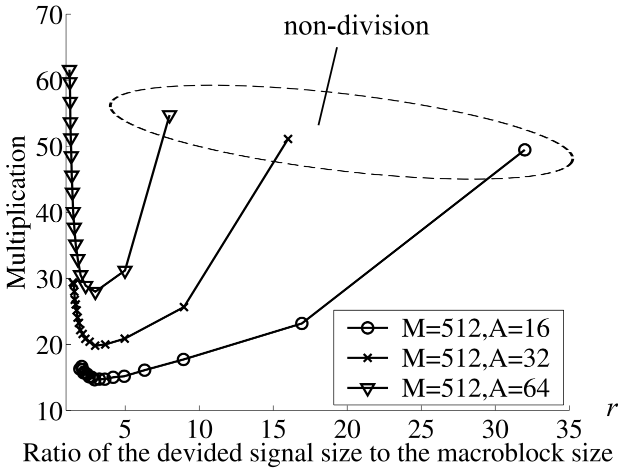


Fig. 5. Theoretical value of computational complexity per one output of proposed method in 1-D block matching.

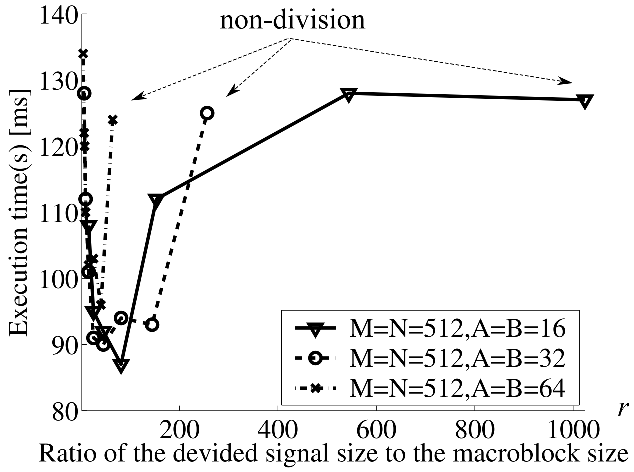


Fig. 6. Block matching speed against the ratio of the divided signal size to the macroblock size ($r = M_2N_2/AB$).

Table 2. The optimal ratio r (= M_2N_2/AB).

$(M \times N, A \times B)$	Optimal r	Time(s) [ms]	Non-division time(s) [ms]	Efficiency
$(512 \times 512, 16 \times 16)$	81.0	87	127	1.46
$(512 \times 512, 32 \times 32)$	45.0	90	125	1.39
$(512 \times 512, 64 \times 64)$	40.3	96	124	1.29

5. SIMULATION

5.1. Conditions

We evaluated the proposed method under the simplest conditions for block matching. There were only one macroblock and one search window. The search window size was fixed at 512×512. There were three macroblock sizes: 16×16, 32×32, and 64×64. All the simulations were carried out using MATLAB 6.5 on a computer with an Intel Core2 2.4-GHz CPU and 2-Gibytes RAM.

5.2. Results

Fig. 6 shows the execution time of the proposed method against $r = M_2N_2/AB$. As shown in Fig. 6, it is confirmed optimal r exists which minimizes the execution time in this simulation.

Table 2 lists optimal r and runtime for three different sized macroblocks. As shown in Table 2, as the size difference between the original search window and the macroblock increases, the efficiency of the proposed method is enhanced.

6. CONCLUSIONS

In this paper, we proposed a novel full-search block matching algorithm that uses overlap-add. By dividing the search window to multiple sub search windows, the proposed method allows us to choose the FFT size appropriate for the environments. The simulation results show that the proposed method is about 1.3 to 1.5 times faster than the conventional method.

ACKNOWLEDGMENT

This work has been partly supported by the Grant-in-Aid for Scientific Research (C), No.20560361, from Japan Society for the Promotion of Science.

REFERENCES

[1] S.L. Kilthau, M.S. Drew, and T. Moller, “Full search content independent block matching based on the fast fourier transform,” *in Proc. IEEE ICIP*, 2002, pp.669–672.

[2] F. Essannouni, R.O.H. Thami, D. Aboutajdine, and A. Salam, “Simple noncircular correlation method for exhaustive sum square difference matching,” *Opt. Eng.*, vol.46, pp.107004-1–107004-4, Oct. 2007.

[3] Z. Li, A. Uemura, and H. Kiya,“An FFT-based full-search block-matching algorithm with SSD criterion,” *in Proc. APSIPA ASC*, 2009, pp.457–460.

[4] Z. Li and H. Kiya,“Double-search-window block matching using the fast fourier transform,” *in Proc. IEEE ICASSP*, 2010, no.IVMSP-P11.PG8, pp.1418–1421.

[5] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications, 2nd ed.*, Macmillan, New York, 1992.

[6] J.W. Adams and A.N. Willson, “A new approach to FIR digital filter design with fewer multipliers and reduced sensitivity,” *IEEE Trans. Circuits Syst.*, vol.30, pp.277–283, May 1983.

[7] J.W. Adams and A.N. Willson,“Some efficient digital prefilter structure,” *IEEE Trans. Circuits Syst.*, vol.31, pp.260–265, May 1984.