28th Picture Coding Symposium, PCS2010, December 8-10, 2010, Nagoya, Japan

FFT-BASED FULL-SEARCH BLOCK MATCHING 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 of fast full-search block matching algorithms (BMAs) is based on the fast Fourier transformation (FFT). In conventional methods in this category, the macroblock size must be adjusted to the search window size by zero-padding. In these 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.

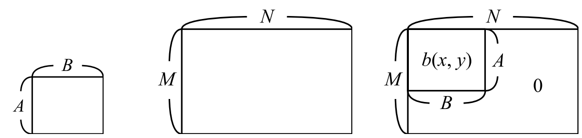
Index Terms— block matching, FFT, overlap-add method, pattern recognition, motion estimation

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 exhaustive searches for every possible candidate in the search window to find the most similar block of the macroblock) imposes a heavy computational load, which makes it almost impossible to use in applications. Therefore, there is a need for a fast and high-accuracy algorithm.

One category in previous studies is based on the fast Fourier transformation (FFT) [1–5]. Compared to the direct full-search block matching algorithm (BMA), FFT-based BMAs greatly decrease the computational load, though they achieve the same accuracy. However, these algorithms need to adjust the macroblock size to the search window size by zero-padding. This increases the memory consumption and computational complexity as the size difference between the macroblock and the search window enlarges.

We developed an FFT-based BMA that overcomes the above problem. The proposed method divides the search window into multiple sub search windows using the overlap-add method (OLA) [6]. This enables us to change the size difference versatilely. Theoretical analysis showed that the proposed method is about 2.0 to 3.4 times more efficient than the conventional method under optimal conditions. Simulation results also demonstrate 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. 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 **Z** denotes the set of integer numbers.

Inside the search window, there are $(N-B+1)\times(M-A+1)$ different blocks, each of which 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*.

The sum of squared differences (SSD) is generally used as the similarity measurement in FFT-based BMAs. The SSD can be expressed as

$$\mathrm{SSD}_{b,f}(u,v)=\sum_{x=0}^{A-1}\sum_{y=0}^{B-1}\{f(x+u,y+v)-b(x,y)\}^2, \tag{3}$$
$$u\in[0,M-A+1], \quad v\in[0,N-B+1], \quad u,v\in\mathbf{Z}.$$

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:

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

978-1-4244-7133-1/10/$26.00 ©2010 IEEE

2.2. FFT-based full-search block matching algorithm

Here, we describe the conventional FFT-based BMA [3, 4]. 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

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

where

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

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

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

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

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

In addition, $S_{f^2}(u,v)$ can efficiently be carried as the recursive summation [7, 8]. An example of the recursive summation is expressed below, where we consider 1-D signals for a simple example:

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

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 increase the memory consumption and the computational load. To solve this problem, we describe our FFT-based block matching algorithm in the next section. 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 that uses the overlap-add method and then describe our new FFT-based BMA using the OLA-based cross-correlation calculation. Furthermore, concurrent processing of the FFT-based BMA is proposed.

3.1. Calculate a cross-correlation by OLA

Overlap-add is one method for calculating a block convolution [6]. We show an efficient method to calculate a cross-correlation by OLA for 1-D signals.

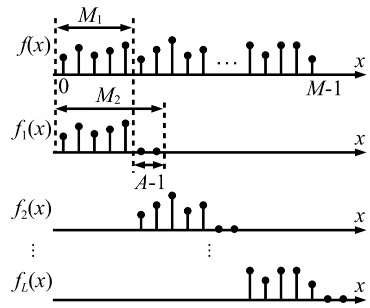


Fig. 2. Conceptual diagram for dividing a signal.

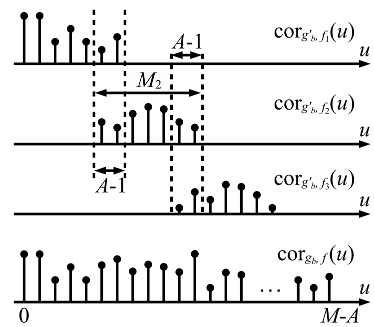


Fig. 3. Conceptual diagram for combining signals.

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 the sub search windows, we can have new signals $f_i(x)$'s which each have an M_2 signal size of $M_1 + A - 1$. Subscript i denotes the index of sub search windows $(i = 1, 2, \ldots, L)$.

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

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

where $g'_b(x)$ is a zero-padded macroblock of length 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 we showed for 1-D signals, we split the search window $f(x,y)$ into multiple adjoining sub search windows with sub windows of size $M_1 \times N_1$. 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

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

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 [1–5].

3.3. Concurrent processing of FFT-based BMA

In most cases, the macroblock and the 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. Figure 4 is a 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 as shown in Figs. 2 and 3. For an M_2-length signal $f_i(x)$, an FFT or IFFT needs $M_2\log_2 M_2$ complex additions and $(M_2/2)(\log_2 M_2 - 1)$ complex multiplications, where each complex addition consists of two real additions and two real multiplications, and each complex multiplication consists of four real multiplications. For L-division, as shown in Fig. 2, $L+1$ times FFT and IFFT operations are required to obtain $(M-A+1)$-length $\mathrm{cor}_{g_b,f}(u)$. Thus, for calculating

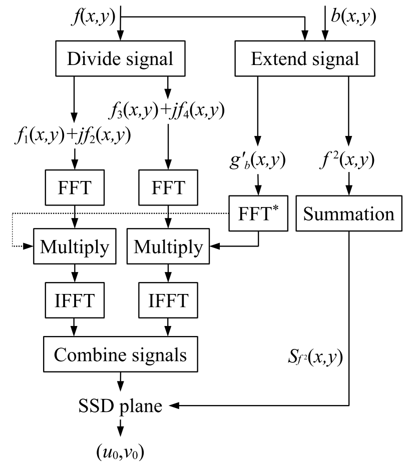


Fig. 4. Proposed method ($L = 4$, FFT* means complex conjugate of FFT).

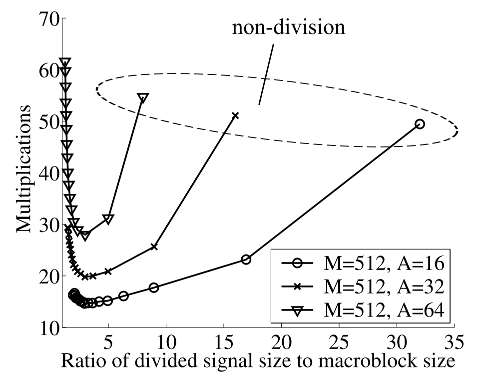


Fig. 5. Theoretical value of computational complexity per one output of proposed method in 1-D block matching.

a single point in $\mathrm{cor}_{g_b,f}(u)$, A_p of real additions and M_p of real multiplications are needed, where

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

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

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

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

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

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,

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

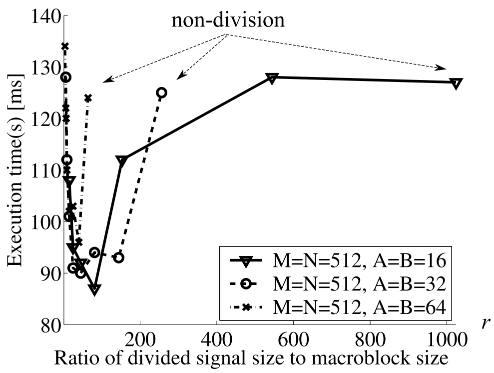


Fig. 6. Block matching speed against the ratio of the divided signal size to the macroblock size ($r = M_2N_2/AB$).

whereas $L+1$ decreases as r increases. This means a particular r exists which minimizes the computational complexity.

Figure 5 illustrates M_p versus r. It explicitly shows that there is an optimal r. Table 1 lists the optimal r and computational costs for several different sized macroblocks. As shown in Table 1, the proposed method is expected to improve the computational efficiency up to about 2.0 to 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.

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 of RAM.

5.2. Results

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

Table 2 lists the 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 increased, the efficiency of the proposed method was enhanced.

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

6. CONCLUSIONS

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

ACKNOWLEDGMENTS

This work has been partly supported by Grant-in-Aid for Scientific Research (C) No.20560361 from the 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,"An FFT-based full-search block-matching algorithm with sum of squared difference criterion," *IEICE Trans. Fundamentals*, to be published.

[5] ——,"Double-search-window block matching using the fast Fourier transform," in *Proc. IEEE ICASSP*, 2010, pp.1418–1421.

[6] J.G. Proakis and D.G. Manolakis, *Digital signal processing: principles, algorithms, and applications*, 2nd ed., Macmillan, New York, 1992.

[7] 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.

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