A PARALLEL IMPLEMENTATION METHOD OF FFT-BASED FULL-SEARCH BLOCK MATCHING ALGORITHMS

Toshiyuki DOBASHI and Hitoshi KIYA

Tokyo Metropolitan University, Dept. of Info. and Commun. Systems
6–6 Asahigaoka, Hino-shi, Tokyo, JAPAN

ABSTRACT

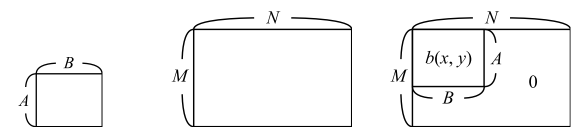
One category of fast full-search block matching algorithms (BMAs) is based on the fast Fourier transformation (FFT). This paper proposes a parallel implementation method of FFT-based full-search BMAs. The FFT-based full-search BMAs are much faster than the direct full-search BMA, and its accuracy is as same as the direct full-search BMA. However, these are not designed for parallel processing. The proposed method divides the search window into multiple sub search windows using the overlap-save method, and the FFT-based full-search BMA is applied to each sub search window. These sub search windows are processed in parallel. By dividing the search window, the method can not only process in parallel, but also select the efficient FFT size. Furthermore, the method can also calculate two cross-correlations at the same time. These properties also contribute to speeding up of the block matching. The experimental results shows that the method on 6 cores CPU is about 11 times faster than the conventional FFT-based full-search BMA.

Index Terms— Block matching, FFT, Overlap-save method, Parallel processing, Fast algorithm

1. INTRODUCTION

A block matching is widely used in many fields, including pattern recognition, object tracking, motion detection, computer vision, motion estimation, inpainting and image denoising [1–6]. Because of its efficiency and simplicity, it has been also widely adopted in many video coding standards, such as H.263, H.264, MPEG-2, and MPEG-4. 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 application. To solve this problem, many fast block matching algorithms (BMAs) have been developed. Their basic approaches can be generally divided into three types.

The first type uses an approximative search window instead of a direct full search window. For example three-step search [7], four-step search [8], and diamond search [9] are based on this approach. While the computational load is decreased, the accuracy is less than that of a full-search BMA, and the initial value affects the results.



(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.

The second type has the same performance as a direct full-search BMA in terms of accuracy, but imposes a lighter computational load so processing speed is higher. The successive elimination algorithm (SEA) [10] and the fast full-search block matching algorithm [11] are representatives of this type. However, the degree to which the computational load can be reduced depends on the input signal.

These first two types operate in the spatial domain. The third type shifts the spatial domain problem into the frequency domain by using phase correlation [12] or cross-correlation [13–16]. The third type of BMAs ensures the same accuracy as a direct full search does. The same time, because of using FFT approach, its computational load is low and does not depend on the input signal. In all these FFT-based full-search BMAs, the algorithm of Kiya et al. [16] has the highest processing speed. However, these algorithms are not designed for parallel processing.

This paper proposes a parallel implementation method for the FFT-based full-search BMAs. The proposed method divides the search window into multiple sub search windows using the overlap-save method (OLS) [17], and then, the FFT-based full-search BMA is applied to each sub search window. These sub search windows are processed in parallel. Dividing the search window enables to select the efficient FFT size. Moreover, the method can calculate two cross-correlations at the same time. The experimental results shows that the proposed method is 10.71 times faster than the conventional FFT-based full-search BMA [16] on 6 cores CPU under the optimal conditions.

2. PREPARATION

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 the *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 BMA

This section describes the conventional FFT-based full-search BMA [16]. First, by zero-padding $b(x, y)$, signal $g_b(x, y)$ which is the same size as $f(x, y)$is generated as shown in Fig. 1(c). Then, Eq. (3) can be rewritten 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,

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

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

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

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

$$(u_0, v_0) = \arg\max_{u,v}\{\mathrm{SSD'}_{b,f}(u, v)\}. \tag{9}$$

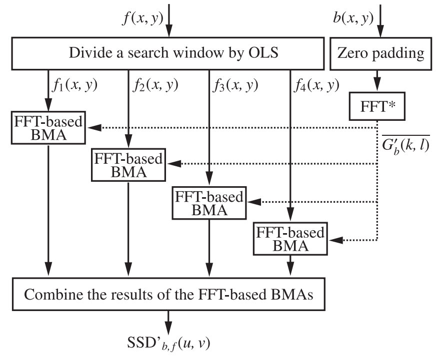


Fig. 2. The block diagram of the method. ($L = 4$, FFT* means complex conjugate of FFT).

where,

$$\mathrm{SSD'}_{g'_b,f}(u, v) = 2\mathrm{cor}_{g,f}(u, v) - S_{f^2}(u, v). \tag{10}$$

$S_{f^2}(u,v)$ can efficiently be carried as the recursive summation [19,20]. An example of the recursive summation is as follows. For simplicity, this example considers 1-D signals.

$$S_{f^2}(u) = \begin{cases} \sum_{k=0}^{A-1} f^2(u+k), & u = 0, \\ S_{f^2}(u-1) - f^2(u-1) + f^2(u+A-1), & u > 0. \end{cases} \tag{11}$$

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.

3. PROPOSED METHOD

The proposed method divides the search window into multiple sub search windows to parallel processing. Then, the FFT-based full-search BMA is applied to each sub search window. These sub search windows are processed in parallel. After all the processes are finished, the results of the FFT-based BMAs are combined. Fig. 2 shows the outline of the method.

By dividing the search window, the method can not only process in parallel, but also select the efficient FFT size. Furthermore, the method can also calculate two cross-correlations at the same time. This section describes about dividing the search window by OLS, the FFT-based BMA for each sub seach window, and concurrent calculation of the cross-correlations.

3.1. Dividing the search window by OLS and the FFT-based full-search BMA

The overlap-save method (OLS) is one of the calculation method for a block convolution [17,18].

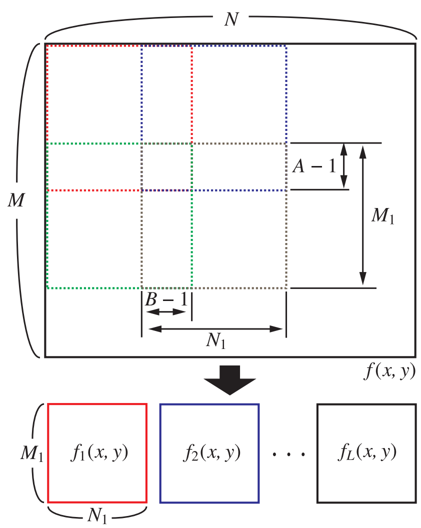


Fig. 3. The conceptual diagram for dividing the search window.

First, as shown in Fig. 3, the search window is divided into L of $M_1 \times N_1$ sized multiple sub search windows using OLS. Each sub search window is overlapped $A-1$ and $B-1$ points with neighbor one. Subscript i denotes the index of sub search windows ($i = 1, 2, \ldots, L$).

Then, $\mathrm{SSD'}_{g'_b, f_i}(u, v)$ is calculated as

$$\mathrm{SSD'}_{g'_b, f_i}(u, v) = 2\mathrm{cor}_{g'_b, f_i}(u, v) - \mathrm{S}_{f_i^2}(u, v), \quad u \in [0, M_1 - 1], \quad v \in [0, N_1 - 1], \tag{12}$$

where $g'_b(x, y)$ is a zero-padded macroblock whose size is $M_1 \times N_1$. The cross-correlation $\mathrm{cor}_{g'_b, f_i}(u, v)$ between $g'_b(x, y)$ and $f_i(x, y)$ can be calculated easily by using FFT. It is written as,

$$\mathrm{cor}_{g'_b, f_i}(u, v) = \frac{1}{M_1 N_1} \sum_{k=0}^{M_1-1} \sum_{l=0}^{N_1-1} \{\overline{G'_b(k, l)} F_i(k, l)\} W_{M_1}^{-uk} W_{N_1}^{-vl}, \tag{13}$$

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.

Finally, $\mathrm{SSD'}_{b,f}(u, v)$ is obtained by combining all the $\mathrm{SSD'}_{g'_b, f_i}(x, y)$ (see Fig. 4). Fig. 5 shows the block diagram of the FFT-based BMA which applied to each sub search window.

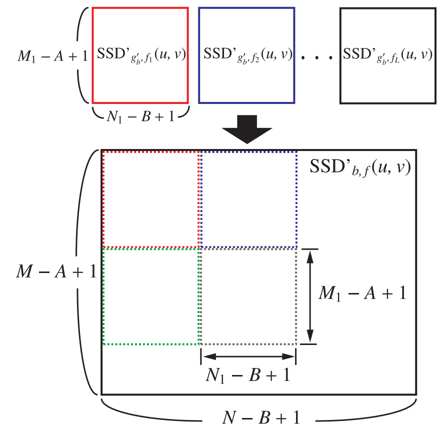


Fig. 4. The conceptual diagram for combining results of the FFT-based BMAs.

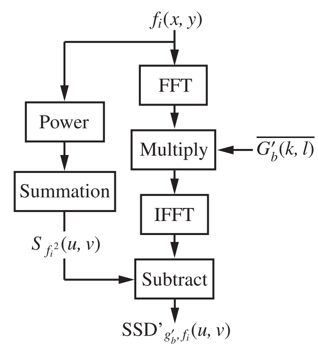


Fig. 5. The block diagram of the FFT-based BMA.

3.2. Concurrent calculation of the cross-correlations

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

First, $f_i(x, y)$ and $f_{i+1}(x, y)$ are combined 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,

$$\begin{aligned}
\mathrm{cor}_{g'_b, \hat{f}}(u, v) &= \frac{1}{M_1 N_1} \sum_{k=0}^{M_1-1} \sum_{l=0}^{N_1-1} \{\overline{G'_b(k, l)} \hat{F}(k, l)\} W_{M_1}^{-uk} W_{N_1}^{-vl} \\
&= \frac{1}{M_1 N_1} \sum_{k=0}^{M_1-1} \sum_{l=0}^{N_1-1} \{\overline{G'_b(k, l)} (F_i(k, l) + jF_{i+1}(k, l))\} W_{M_1}^{-uk} W_{N_1}^{-vl} \\
&= \frac{1}{M_1 N_1} \sum_{k=0}^{M_1-1} \sum_{l=0}^{N_1-1} \{\overline{G'_b(k, l)} F_i(k, l)\} W_{M_1}^{-uk} W_{N_1}^{-vl} \\
&\quad + j \frac{1}{M_1 N_1} \sum_{k=0}^{M_1-1} \sum_{l=0}^{N_1-1} \{\overline{G'_b(k, l)} F_{i+1}(k, l)\} W_{M_1}^{-uk} W_{N_1}^{-vl} \\
&= \mathrm{cor}_{g'_b, f_i}(u, v) + j\mathrm{cor}_{g'_b, f_{i+1}}(u, v).
\end{aligned} \tag{14}$$

By separating Eq. (14) into the real and imaginary parts, $\mathrm{cor}_{g'_b, f_i}(u,v)$ and $\mathrm{cor}_{g'_b, f_{i+1}}(u,v)$ are obtained at the same time.

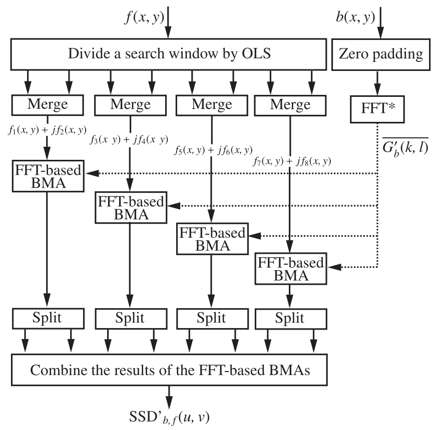


Fig. 6. The block diagram of the proposed method with concurrent calculation of the cross-correlations. (L = 8).

This property leads to a reduction in computational cost of the proposed method. Fig. 6 shows the block diagram of the proposed method with concurrent calculation of the cross-correlations.

4. EXPERIMENTAL RESULTS

To evaluate the processing time of the proposed method, the method was implemented, and the experiment was carried out.

4.1. Conditions

The experiment used the image of 1024 × 1024 pixels, and the search window sizes were 1024 × 1024 and 512 × 512 pixels. Three images whose sizes were 16 × 16, 32 × 32, and 64×64 pixels were used as the macroblock. The program was implemented using Microsoft Visual C++ 2010 and OpenCV 2.4.2. The experimental platform was intel Core i7 3930K CPU at 3.2GHz (6 cores, Hyper-Threading was disabled) with 16GB RAM.

4.2. Results

Fig. 7 shows the processing time of the proposed method and the conventional FFT-based full-search BMA [16] when the search window size was 1024×1024. The horizontal axis r of the graph indicates the ratio of the sub search window size to the macroblock size ($r = M_1N_1/AB$). This figure shows that the optimal r, which minimizes the processing time exists. On the other hand, the maximum r, the case of non division and non parallel processing, is equal to the conventional FFT-based full-search BMA [16]. Tab. 1 shows the processing time of the conventional method and the proposed method (at the optimal r). In each case, the proposed method was

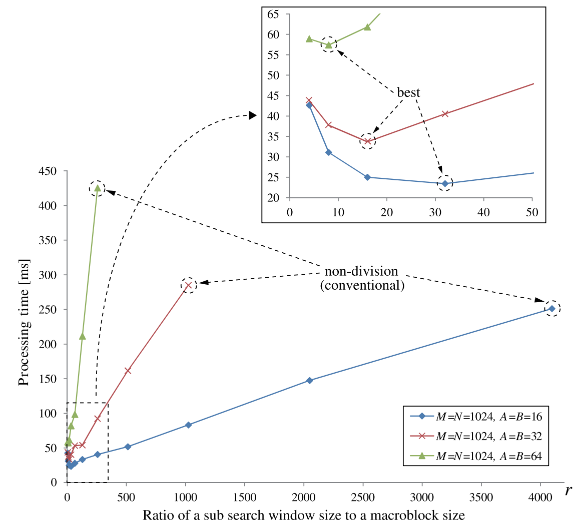


Fig. 7. The processing time against the ratio of the sub search window size to the macroblock size ($r = M_1N_1/AB$).

Table 1. The processing time of the conventional method and the proposed method (at the optimal r).

Size (pixels) ($M \times N, A \times B$)	Processing time (ms)		Speeding up (times)
	Proposed	Conventional	
(1024 × 1024, 16 × 16)	23.45	251.17	10.71
(1024 × 1024, 32 × 32)	33.68	285.04	8.46
(1024 × 1024, 64 × 64)	57.40	425.37	7.41

above 6 times faster than the conventional method on 6 cores CPU. Especially, the proposed method was 10.71 times faster than the conventional method when the macroblock size was 16 × 16. On the other hand, the proposed method was 8.08 times faster than the conventional method when the search window size was 512×512 and the macroblock size was 16× 16. The experiment showed that the proposed method became effective as the search window size became larger.

5. CONCLUSIONS

A parallel implementation method of FFT-based BMAs was proposed in this paper. By dividing the search window, the proposed method can not only process in parallel, but also select the efficient FFT size. Moreover, the method can also calculate two cross-correlations at the same time. By these features, the method on 6 cores CPU can achieve the processing time that above 6 times faster than the conventional FFT-based BMA [16].

6. REFERENCES

[1] A. Uemura, Z. Li, and H. Kiya, “Fast Sub-Pixel Motion Estimation Using FFT-based Full Search Block Matching,” in Proc. APSIPA ASC, vol.Student Symposium, p.29, Biopolis, Singapore, 15th December, 2010.

[2] A. Koschan, V. Rodehorst, and K. Spiller, “Color Stereo Vision Using Hierarchical Block Matching and Active Color Illumination,” in Proc. 13th Int. Conf. on Pattern Recognition, vol.1, pp. 835-839, Vienna, Austria, Aug. 1996.

[3] Y. Wexler, E. shechtman, and M. Irani, “Space-Time Completion of Video,” IEEE Trans. PAMI, vol.29, pp.463–476, Mar. 2007.

[4] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in Proc. IEEE CVPR, vol.2, 2005, pp.60–65.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising with block-matching and 3D filtering,” Proc. of SPIE-IS&T Electronic Imaging, SPIE vol.6064, pp.606414(1)-606414(12), 2006.

[6] A. Takagi, S. Muramatsu, and H. Kiya, “Motion Estimation with Power Scalability and Its VHDL Model,” Proc. IEEE International Conference on Image Processing, no.WA04.06, Vancouver, Canada, 1st September, 2000.

[7] T. Koga, K. Iinuma, A. Hirano, and Y. Iijima, “Motion compensated interframe coding for video conferencing,” Proc. IEEE NTC Conf., pp.G5.3.1–G5.3.5, New Orleans, Dec. 1981.

[8] L. Po and W. Ma, “A novel four-step search algorithm for fast block motion estimation,” IEEE Trans. Circuits Syst. Video Technol., vol.6, no.3, pp.313–317, Jun. 1996.

[9] S. Zhu and K. Ma, “A new diamond search algorithm for fast block motion estimation,” IEEE Trans. Image Process., vol.9, no.2, pp.287–290, Feb. 2000.

[10] W. Li and E. Salari, “Successive elimination algorithm for motion estimation,” IEEE Trans. Image Process., vol.4, no.1, pp.105-107, Jan. 1995.

[11] Y. Noguchi, J. Furukawa, and H. Kiya, “A fast full search block matching algorithm for MPEG-4 video,” IEEE Int. Conf. Image Process., vol.1, pp.61–65, Kobe, Japan, Oct. 1999.

[12] M. Li, M. Biswas, S. Kumar, and T. Nguyen, “DCT-based phase correlation motion estimation,” IEEE Int. Conf. Image Process., vol.1, pp.445–448, Singapore, Oct. 2004.

[13] A. Fitch, A. Kadyrov, W. christmas, and J. Kittler, “Fast robust correlation,” IEEE Trans. Image Process., vol.14, no.8, pp.1063–1073, Aug. 2005.

[14] 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.

[15] Z. Li and H. Kiya,“Double-Search-Window Block Matching Using the Fast Fourier Transform,” in Proc. IEEE ICASSP, no.IVMSP–P11.8, pp.1418–1421, Dallas, TX, the U.S., 19th March, 2010.

[16] Z. Li, A. Uemura, and H. Kiya,“An FFT-Based Full-Search Block Matching Algorithm with Sum of Squared Difference Criterion,” IEICE Trans. Fundamentals, vol.E93–A, no.10, pp.1748–1754, October 2010.

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

[18] S. Muramatsu, and H. Kiya, “Extended Overlap-Add and -Save Methods for Multirate Signal Processing,” IEEE Trans. Signal Processing, vol.45, no.9, pp.2376–2380, September 1997.

[19] 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.

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