# An FFT-Based Full-Search Block-Matching Algorithm With SSD Criterion

Zhen Li*, Atsushi Uemura* and Hitoshi Kiya*
* Faculty of System Design,Tokyo Metropolitan University,6-6 Asahigaoka Hino-shi 191-0065 Tokyo Japan
E-mail: li-zhen@sd.tmu.ac.jp,uemura-atsushi@sd.tmu.ac.jp,kiya@eei.metro-u.ac.jp Tel: +81-42-585-8454

***Abstract*—An FFT-based full-search block-matching algorithm is described that uses the sum of squared difference(SSD) criterion. The proposed method focuses on the relationship between the circular cross-correlation and the SSD criterion. Because FFT is used to calculate the cross-correlation between signals of different sizes, the processing speed of block matching is greatly increased. If the macroblock is composed of real signals, two blocks can be matched at the same time. In a simulation of motion estimation, the proposed method achieved the same performance as a direct SSD full search, but with a processing speed 8 to 700 times higher.**

## I. Introduction

Block matching is widely used in many fields, including pattern recognition, object tracking, motion detection, and motion estimation. Because of its efficiency and simplicity, it has been widely adopted in many video coding standards. 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 full search window. For example three-step search[1] and diamond search[2] are based on this approach. While the computational load is greatly decreased, the accuracy is less than that of a full-search BMA, and the initial value greatly affects the results.

The second type has the same performance as a full-search BMA in terms of accuracy, but imposes a lighter computational load so processing speed is higher. The successive elimination algorithm(SEA)[3][4] is one representative 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[5] or cross-correlation[6][7]. And in all these full-search BMAs, SSDcorr[7] has the highest processing speed.

Here we describe an FFT-based full-search BMA that is of the third type. The proposed method focuses on the relationship between the circular cross-correlation and the SSD criterion. With this method, we do not have to extend real signals into complex signals, a big difference compared to other methods using FFT-based cross-correlation[6][7]. Whatever the input signal, an FFT approach involves complex arithmetics. Therefore, if two macroblocks share the same search window and all signals are real, the proposed method can match them at the same time. And the greater the number of macroblocks, the higher the processing speed.

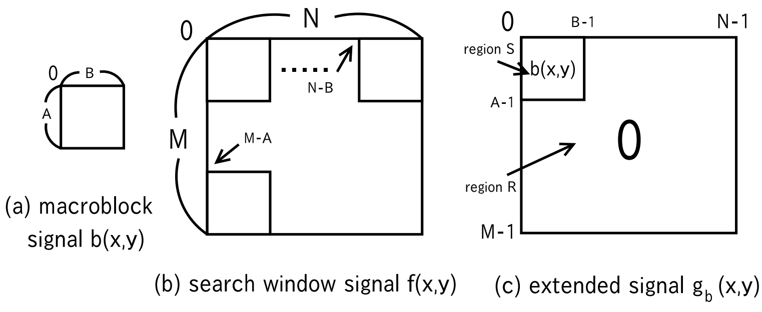



Fig. 1. macroblock and Search window.

## II. Preparation

We begin by defining block matching and discussing the relationship between the cross-correlation and SSD of two circular signals of the same size. Let $\mathbf{Z}$ denote the set of integer numbers.

### A. Block matching

First, a definition for block matching is needed. As shown in Fig.1(a) and (b), let 2-D signal $b(x,y)$ be a macroblock, and let 2-D signal $f(x,y)$ be the search window. Suppose that the search window is bigger than the macroblock. That is,

$$b(x,y), \quad x=0,1,\ldots,A-1, \quad y=0,1,\ldots,B-1, \quad (1)$$
$$f(x,y), \quad x=0,1,\ldots,M-1, \quad y=0,1,\ldots,N-1. \quad (2)$$
$$A<M, B<N, \quad x,y,A,B \in \mathbf{Z}.$$

Inside the search window there are $(N-B)\times(M-A)$ different blocks, which have the same size as the macroblock in terms of integral pixels. All these different blocks are compared with the macroblock to find the most similar one(the one with the minimum matching error). This procedure can be defined as "full-search block matching". One of the matching criteria is the sum of squared differences(SSD) as

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

The $u$ and $v$ are shift amounts. The purpose of block matching is to find the particular $(u,v)$ that yields the minimum of the matching criterion:

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

### *B. Circular cross-correlation and SSD*

Next, let us take a look at the relationship between the circular cross-correlation and SSD of two circular signals, $g(x,y)$ and $f(x,y)$, that have the same period ($M \times N$). Given shift amounts $u$ and $v$, the circular cross-correlation of $g(x,y)$ and $f(x,y)$ can be defined as

$$\widehat{cor}_{g,f}(u,v) = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} g(x,y)f(x+u,y+v), \quad (5)$$
$$(u \in [0, M-1], v \in [0, N-1]).$$

The SSD of two signals can be written as

$$\begin{aligned} SSD_{g,f}(u,v) &= \sum_{y=0}^{N-1}\sum_{x=0}^{M-1}\{g(x,y) - f(x+u,y+v)\}^2 \\ &= C_g - 2\widehat{cor}_{g,f}(u,v) + C_f. \end{aligned} \quad (6)$$

where

$$C_g = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1}\{g(x,y)\}^2, \quad (7)$$
$$C_f = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1}\{f(x+u,y+v)\}^2 = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1}\{f(x,y)\}^2. \quad (8)$$

Because $C_g$ and $C_f$ are independent of shift amounts $u$ and $v$,

$$(u_0, v_0)_{SSD} = \min_{u,v}\{SSD_{g,f}(u,v)\} = \max_{u,v}\{\widehat{cor}_{g,f}(u,v)\}. \quad (9)$$

The DFT of the cross-correlation, the "cross-spectrum" can be calculated using the FFT approach if both signals are circular and have the same period. If these conditions hold, the $(u_0, v_0)_{SSD}$ in Eq.(9) can be easily found.

## III. PROPOSED METHOD

### *A. Non-circular cross-correlation*

In the block matching, the macroblock $b(x,y)$ and search window $f(x,y)$ are non-circular signals of different sizes. Therefore, we can not use the property discussed in the foregoing section directly. To solve this problem, we need to extend macroblock signal $b(x,y)$ into a new signal, $g_b(x,y)$, by padding it with zeros, as shown in Fig.1(c). Namely,

$$\begin{cases} \text{region R} = \{(x,y) | A-1 < x \le M-1 \text{ or } B-1 < y \le N-1\}, \\ \text{region S} = \{(x,y) | 0 \le x \le A-1 \text{ and } 0 \le y \le B-1\}, \\ \text{entire region Q} = R \cup S. \end{cases} \quad (10)$$

$$\begin{cases} g_b(x,y) = 0, \quad (x,y) \in R, \\ g_b(x,y) = b(x,y), \quad (x,y) \in S. \end{cases} \quad (11)$$

Then it is assumed that the extended signal, $g_b(x,y)$, and search window signal, $f(x,y)$, are both circular. Therefore, the circular cross-correlation of the two signals can be written as

$$\widehat{cor}_{g_b,f}(u,v) = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} g_b(x,y)f(x+u,y+v), \quad (12)$$
$$(u \in [0, M-1], v \in [0, N-1]).$$

Let the range of shift amounts $u$ and $v$ be appropriate to the case of block matching in Eq.3 and define the cross-correlation as

$$cor_{g_b,f}(u,v) = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} g_b(x,y)f(x+u,y+v), \quad (13)$$
$$(u \in [0, M-A], v \in [0, N-B]).$$

Then we can get the relation

$$cor_{g_b,f}(u,v) = \widehat{cor}_{g_b,f}(u,v), \quad (14)$$
$$(u \in [0, M-A], v \in [0, N-B]).$$

This means that the non-circular cross-correlation, $cor_{g_b,f}(u,v)$, can be calculated by using the FFT.

### *B. Non-circular cross-correlation and SSD*

According to Eq.(6), $SSD_{g_b,f}(u,v)$ can be written as

$$SSD_{g_b,f}(u,v) = C_{g_b} - 2cor_{g_b,f}(u,v) + \sum_{(x,y)\in Q}\{f(x+u,y+v)\}^2, \quad (15)$$

where

$$C_{g_b} = \sum_{y=0}^{B-1}\sum_{x=0}^{A-1}\{b(x,y)\}^2. \quad (16)$$

Since Eq.(3) and $g_b(x,y)$ are made up of two regions, $SSD_{g_b,f}(u,v)$ can also be written as

$$\begin{aligned} &SSD_{g_b,f}(u,v) \\ &= \sum_{(x,y)\in R}\{g_b(x,y) - f(x+u,y+v)\}^2 + \\ &\sum_{(x,y)\in S}\{g_b(x,y) - f(x+u,y+v)\}^2, \\ &= \sum_{(x,y)\in R}\{f(x+u,y+v)\}^2 + SSD_{b,f}(u,v). \end{aligned} \quad (17)$$

Therefore, combining Eqs.(15) and (17) we obtain

$$SSD_{b,f}(u,v) = C_{g_b} - 2cor_{g_b,f}(u,v) + S_{f^2}(u,v), \tag{18}$$

where

$$S_{f^2}(u,v) = \sum_{(x,y)\in S} \{f(x+u,y+v)\}^2, \tag{19}$$

$$= \sum_{(x,y)\in S} f^2(x+u,y+v). \tag{20}$$

Thus,

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

$$= \max_{u,v}\{2cor_{g_b,f}(u,v) - S_{f^2}(u,v)\}. \tag{21}$$

The $cor_{g_b,f}(u,v)$ can be easily calculated by using the FFT. Since $S_{f^2}(u,v)$ can be considered as the convolution or the correlation, there are some different ways to calculate it.

For example, we may use a recursive running sum (RRS) filter[8] and the transfer function for the RRS filter can be written as

$$H(z_1,z_2) = (1 + z_1^{-1} + \cdots + z_1^{-(B-1)})$$

$$(1 + z_2^{-1} + \cdots + z_2^{-(A-1)}), \tag{22}$$

$$= \frac{1-z_1^{-B}}{1-z_1^{-1}}\frac{1-z_2^{-A}}{1-z_2^{-1}}. \tag{23}$$

When $S_{f^2}(u,v)$ is calculated as the convolution, only 4 additions is needed for one coordinate point.

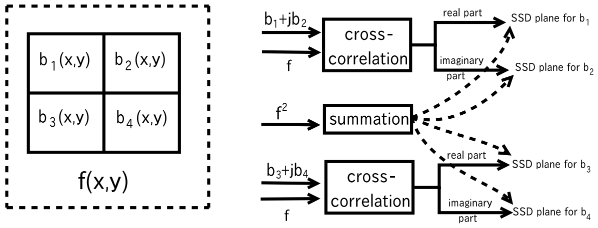
We may also calculate $S_{f^2}(u,v)$ using FFT-based cross-correlation which has higher processing speed and is easier to be applied, if the process is performed by software such as Matlab.

In addition, from a point view of computational load, the heaviest part in Eq.(21) is the calculation of $cor_{g_b,f}(u,v)$.

### C. BMA for real signals

In most cases, macroblock and search window in the block matching are real signals. In contrast, the FFT approach is designed for complex signals. Unlike other FFT-based methods that need to use the imaginary part of real signals, the proposed method does not extend a real signal into a complex one. Therefore, if we combine two real signals to form a complex signal before the FFT approach, the proposed method can match two macroblocks sharing the same search window at the same time.

For a simple explanation, let us assume three 1-D real signals of the same length, $N$: $g_1(n)$, $g_2(n)$, and $f(n)$. $G_1(k)$, $G_2(k)$, and $F(k)$ are the DFTs of the signals, respectively. We combine $g_1(n)$ and $g_2(n)$ to create a new complex signal, $h(n) = g_1(n) + jg_2(n)$. The DFT of $h(n)$ can be written as $H(k) = G_1(k) + jG_2(k)$. Therefore, the cross-correlation between $h(n)$ and $f(n)$ can be written as




(a) Four blocks sharing the same search window

(b) Flow chart

Fig. 2. Four macroblocks sharing the same search window.

$$\widehat{cor}_{h,f}(n) = \frac{1}{N}\sum_{k=0}^{N-1}\overline{H(k)}F(k)W_N^{-nk}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}(\overline{G_1(k)} - j\overline{G_2(k)})F(k)W_N^{-nk}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\overline{G_1(k)}F(k)W_N^{-nk} - j\frac{1}{N}\sum_{k=0}^{N-1}\overline{G_2(k)}F(k)W_N^{-nk}, \tag{24}$$

where $j$ is the square root of $-1$, $\overline{H(k)}$ means the complex conjugate of $H(k)$, and $W_N^{-nk} = e^{-j2\pi/N}$.

Note that the real part of $\widehat{cor}_{h,f}(n)$ is the cross-correlation between $g_1(n)$ and $f(n)$ and that the imaginary part of $cor_{h,f}(n)$ is the cross-correlation between $g_2(n)$ and $f(n)$.

### D. BMA for multiply macroblocks

Let us consider the case shown in Fig. 2(a). Four macroblocks, $b_1(x,y)$, $b_2(x,y)$, $b_3(x,y)$, and $b_4(x,y)$, share the same search window, $f(x,y)$. They are all real signals.

Using the property discussed before, the proposed method can match two macroblocks at the same time. And please note that $S_{f^2}(u,v)$ is separated from the macroblocks and the search window. Therefore we calculate this part only one time. These two advantages make the proposed method better than SSDcorr[7] at matching multiply macroblocks in terms of the computation efficiency. Fig.3 shows times for FFT and IFFT, which are the heaviest part in the computation load, against number of macroblock sharing the same search window.

Motion estimation may involve such a case of block matching many times, so using the proposed method can increase the processing speed significantly. Fig.4 shows the flow chart for the proposed method when only one macroblock needs to be matched. Fig.2(b) shows the flow chart for matching four macroblocks sharing the same search window.

## IV. SIMULATION

We simulated motion estimation by using the 0-9 frames of the video sequence "caltrain". We used one frame before the current one to generate a predicted frame. The macroblock size

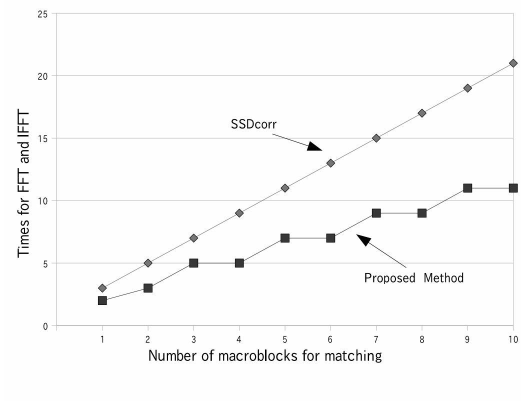



Fig. 3. Times for FFT and IFFT against number of macroblock sharing the same search window.

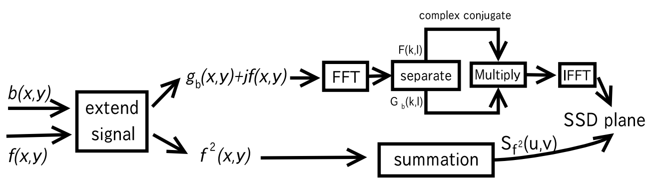



Fig. 4. Flow chart for proposed method. $j$ is the square root of $-1$. $F(k,l)$ and $G_b(k,l)$ are DFTs of $f(x,y)$ and $g_b(x,y)$.

was fixed at $16 \times 16$. There were two search window sizes: $32 \times 32$ ($\pm 8$ pixels around the macroblock) and the entire frame ($400 \times 512$). We compared four different methods in terms of both the PSNR (between the predicted frame and the current one) and the processing speed. The methods were direct SSD full search, SSD diamond search[2], SSDcorr[7], and the proposed method.

Fig.5 plots the PSNR between the predicted frame and the current frame for different methods. The proposed method and SSDcorr are exactly the same as the direct SSD full search from the view point of the PSNR, so in Fig.5 they share the same curve. Diamond search depends greatly on the start point. So if we do not select the start point appropriately, both the PSNR and the processing speed became worse.

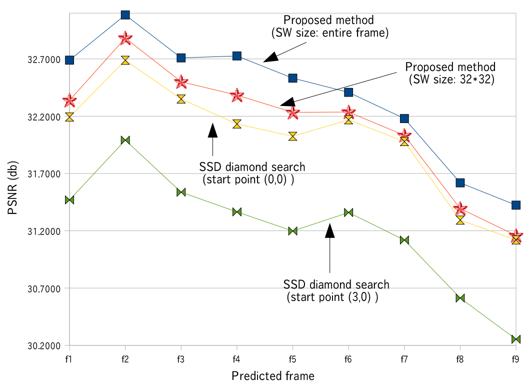



Fig. 5. PSNR between predicted frame and current frame. Diamond search depends greatly on the start point. The larger the search window, the better the PSNR of Proposed method.

TABLE I
PROCESSING SPEED OF GENERATING ONE PREDICTED FRAME(1).

| Method | Time(s) | Ratio |
|---|---|---|
| Proposed (SW size: $32 \times 32$) | 0.36 | 1.0 |
| SSD diamond search (start point (0,0)) | 0.78 | 2.2 |
| SSD diamond search (start point (3,0)) | 1.02 | 2.8 |
| Direct SSD full search (SW size: $32 \times 32$) | 3.05 | 8.5 |

TABLE II
PROCESSING SPEED OF GENERATING ONE PREDICTED FRAME(2).

| Method | Time(s) | Ratio |
|---|---|---|
| Proposed (SW size: entire frame) | 26.6 | 1.0 |
| SSDcorr (SW size: entire frame) | 44.1 | 1.7 |
| Direct SSD full search (SW size: entire frame) | $19.6 \times 10^3$ | 737 |

Table I and Table II show the processing speed of generating one predicted frame using different methods. We can see that the proposed method is faster than the others. (The experiments were performed using Matlab 6.0 on a computer with an Intel Core2 2.4-GHz CPU and 2-GB memory. $S_{f^2}(u,v)$ were calculated using FFT-based cross-correlation.)

## V. CONCLUSIONS

In this paper, we proposed a full-search block-matching algorithm that uses the sum of squared difference (SSD) criterion. Because we define the SSD criterion on the basis of the cross-correlation, which can be calculated using FFT approach, the spatial domain problem is shifted into the frequency domain. If the block-matching signals are all real and more macroblocks share the same search window, the processing speed can be largely increased.

## REFERENCES


[1] T. Koga, K. Iinuma, A. Hirano, and Y. Iijima,"Motion compensated interframe coding for video conferencing," in *Proc. IEEE NTC Conf.*, pp. G5.3.1-G5.3.5, 1981.
[2] 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, 2000.
[3] W. Li and E. Salari,"Successive Elimination Algorithm for Motion Estimation," *IEEE Trans. Image Proc.*, vol. 4, no. 1, pp. 105-107, Jan. 1995.
[4] Y. Noguchi, J. Furukawa, and H. Kiya,"A fast full search block matching algorithm for MPEG-4 video," *IEEE Int. Conf. Image Proc.*, vol. 1, pp. 61-65, 1999.
[5] M. Li, M. Biswas, S. Kumar, and T. Nguyen,"DCT-based phase correlation motion estimation," *IEEE Int. Conf. Image Proc.*, vol. 1, pp. 445-448, 2004.
[6] A. Fitch, A. Kadyrov, W. christmas, and J. Kittler,"Fast robust correlation," *IEEE Trans. Image Process.*, vol. 14, no. 8, pp. 1063-1073, 2005.
[7] F. Essannouni, R. Oulad Haj Thami, D. Aboutajdine, and A. Salam,"Simple noncircular correlation method for exhaustive sum square difference matching," *Optical Engineering,* vol. 46, no. 10, 107004, Oct. 2007.
[8] J. W. Adams and A. N. Willson,"Some Efficient Digital Prefilter Structure," *IEEE Trans. Circuits and Systems,* vol. 31, pp. 260-265, May 1984.