Reduction of the Number of Complex Multiplications of Phase-Only Correlation Using Subsampled-FFT

Kenya Tanaka, Masahide Abe, Masayuki Kawamata
Department of Electoronic Engineering, Graduate School of Engineering,
Tohoku University
Sendai, Japan
Email: {kenya,masahide,kawamata}@mk.ecei.tohoku.ac.jp

***Abstract*—We propose a new computational method for reducing the number of complex multiplications of Phase-Only Correlation (POC) function. This method employs Subsampled-FFT which can compute the equally spaced frequency spectrum of an input signal in $\frac{N}{2s}\log\frac{N}{s}$ complex multiplications, where N is the signal length and $s > 0$ is an integer division parameter. Consequently, the number of complex multiplications of POC function has been reduced to $\frac{3N}{2s}\log\frac{N}{s} + \frac{N}{s}$ by using Subsampled-FFT, wheres that of the conventional POC function is $\frac{3N}{2}\log N + N$.**

Keywords*—*Phase-Only Correlation, Subsampled-FFT, Computational Cost, Image Matching, Digital Signal Processing

I. Introduction

Phase-Only Correlation (POC) function[1] is one of the popular correlation methods. POC function is highly robust for noise because every amplitude spectrum of an input signal is normalized. For this reason, POC function has been used for several areas, such as motion estimation for videos[2], 3D measurement[3], and biometric authentication[4].

The conventional algorithm for computing POC function, however, needs to compute three N-point FFT (see Fig.1), which means that the computational cost of POC function tends to be enormous when the size N of an input signal is large.

We propose a new computational method for this problem. In our method, we only use the equally spaced frequency spectrum of an input signal, which can be obtained efficiently by using Subsampled-FFT, a variant technique of FFT. The details are stated in Section IV.

Note that in this paper we discuss only the one-dimentional case, but our algorithm can be extended to higher dimensions as well.

II. Phase-Only Correlation Function

In this section, we explain the conventional process to compute POC function. The computational flow is shown in Fig.1.

Suppose we have two input signals $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^N$, and would like to compute POC function.

1) Compute the FFTs $\boldsymbol{X}$ and $\boldsymbol{Y}$ of $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively.

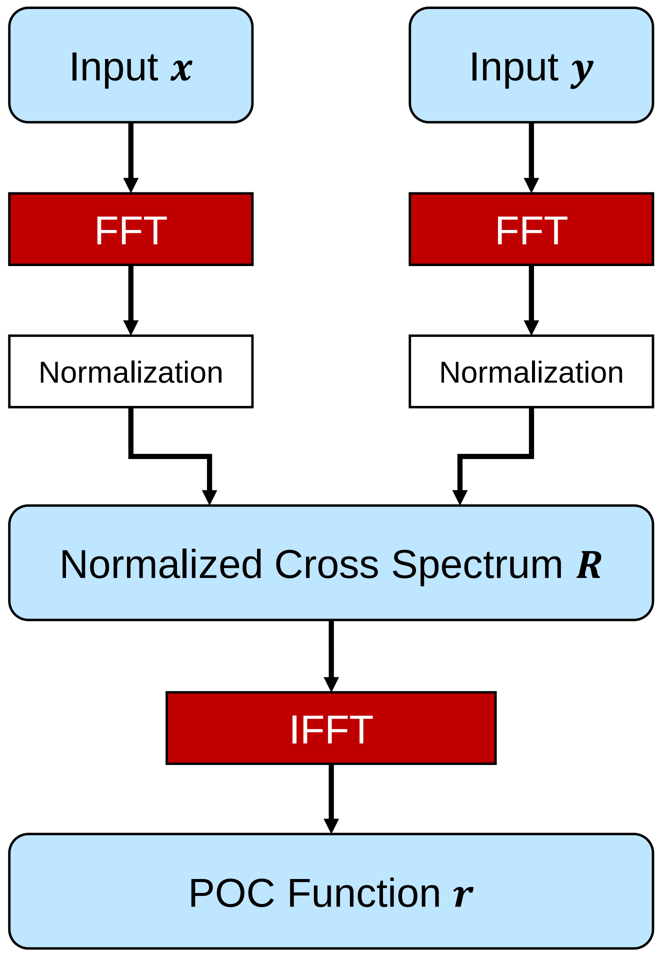

Fig. 1: Computational flow of POC function

2) Compute the normalized cross spectrum $\boldsymbol{R} \in \mathbb{C}^N$ according to the following formula:

$$R(m) = \frac{X(m)Y^*(m)}{|X(m)|\,|Y(m)|} \tag{1}$$

where $Y^*(m)$ is the complex conjugate of $Y(m)$.

3) Finally obtain the POC function $\boldsymbol{r} \in \mathbb{R}^N$ by computing the IFFT of $\boldsymbol{R}$.

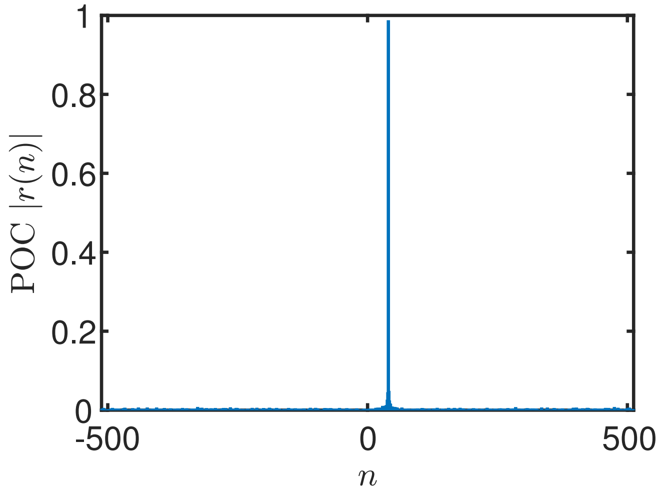

Fig. 2: Typical POC function

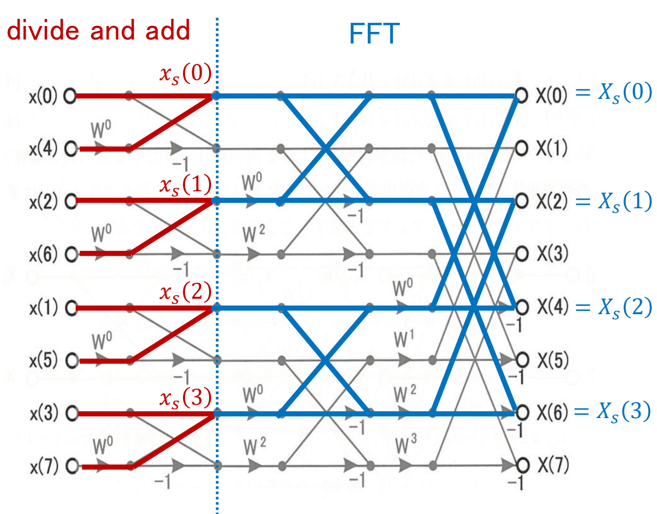

Fig. 3: Flow graph of Subsampled-FFT

A typical POC function is shown in Fig.2. The peak level indicates the strength of the correlation between input signals, and the peak index indicates the displacement value of input signals.

III. Subsampled-FFT

Subsampled-FFT allows us to compute the equally spaced frequency spectrum of an input signal efficiently. For simplicity we will use the notation $[N] = \{0, 1, \ldots, N-1\}$.

Suppose we have an input signal $\boldsymbol{x} \in \mathbb{R}^N$, and define the sub-vector $\boldsymbol{x}_s$ for $n \in \left[\frac{N}{s}\right]$ such that

$$x_s(n) = \sum_{k=0}^{s-1} x\left(n + \frac{N}{s}k\right). \tag{2}$$

Here we divide the signal into s parts and add them.

Then for $m \in \left[\frac{N}{s}\right]$, the FFTs $\boldsymbol{X}$ and $\boldsymbol{X}_s$ of $\boldsymbol{x}$ and $\boldsymbol{x}_s$ satisfy

$$X_s(m) = X(sm) \tag{3}$$

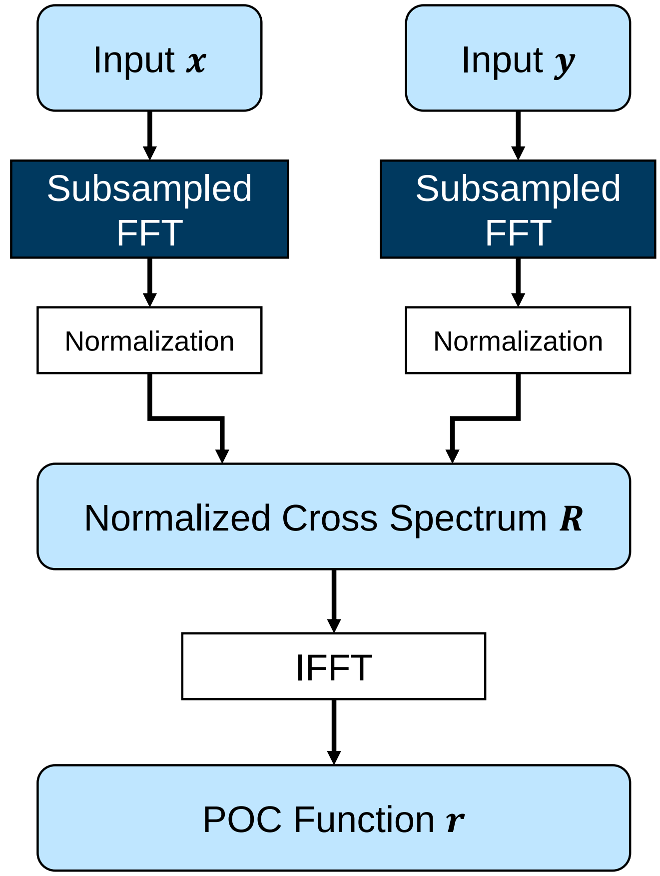

Fig. 4: Computational flow of proposed method

Eq.(3) can be easely understood from the flow graph of FFT (see Fig.3). In (2) we compute the part of the first $\log_2(s)$ stages of FFT, which corresponds to the red part of Fig.3, and then computing the FFT of $\boldsymbol{x}_s$ we can obtain the equally spaced frequency spectrum $\boldsymbol{X}_s$ of the input signal as a result.

Thus, the equally spaced $\frac{N}{s}$-point frequency spectrum of an N-point input signal can be obtained by computing $\frac{N}{s}$-point FFT.

IV. POC Function Using Subsampled-FFT

In this section, we explain our proposed method for computing POC function. The computational flow of our algorithm is shown in Fig.4.

Suppose we have two input signals $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^N$, and would like to compute POC function using Subsampled-FFT.

1) Make sub-vectors $\boldsymbol{x}_s$ and $\boldsymbol{y}_s \in \mathbb{R}^{\frac{N}{s}}$ from $\boldsymbol{x}$ and $\boldsymbol{y}$ using (2).
2) Compute the FFTs $\boldsymbol{X}_s$ and $\boldsymbol{Y}_s$ of $\boldsymbol{x}_s$ and $\boldsymbol{y}_s$, which satisfy (3).
3) Compute the normalized cross spectrum $\boldsymbol{R}_s \in \mathbb{C}^{\frac{N}{s}}$, which also satisfies

$$R_s(m) = R(sm)$$

for $m \in \left[\frac{N}{s}\right]$, where $\boldsymbol{R} \in \mathbb{C}^N$ is the normalized cross spectrum computed in the conventional method.

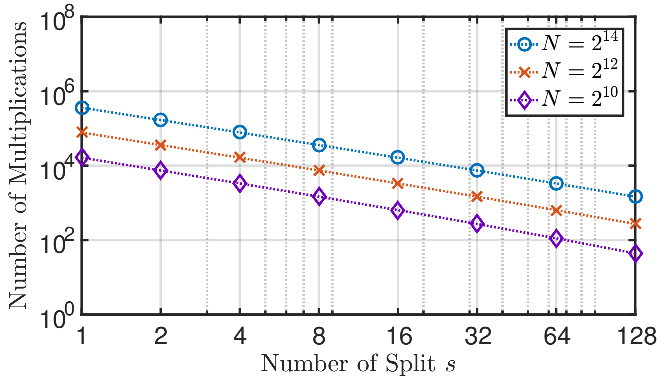

Fig. 5: Number of complex multiplication

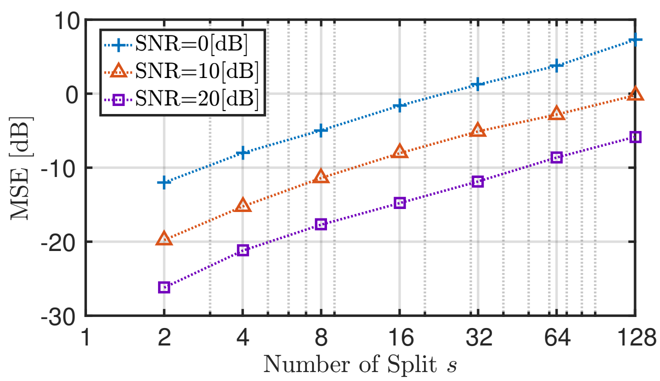

Fig. 6: MSE of $r(0)$ with $r_s(0)$

4) Finally obtain the POC function $\boldsymbol{r}_s \in \mathbb{R}^{\frac{N}{s}}$ by computing the IFFT of $\boldsymbol{R}_s$, which satisfies

$$r_s(n) = \sum_{k=0}^{s-1} r\left(n + \frac{N}{s}k\right) \tag{4}$$

for $n \in \left[\frac{N}{s}\right]$, where $\boldsymbol{r} \in \mathbb{R}^N$ is the POC function obtained by the conventional method.

When two input signals have strong correlation, much of the energy of POC function would be concentrated on the peak index (around $n = 0$ and $n = N-1$ if the displacement value of input signals is small enough). Hence we can approximate $r(n) \approx 0$ except around $n = 0$ and $n = N-1$, and then from (4) we obtain

$$r_s(n) \approx r(n) \tag{5}$$

for $n = \left[\frac{N}{s}\right] - \frac{N}{2s}$. Here $\boldsymbol{r}$ and $\boldsymbol{r}_s$ are periodic from the property of FFT.

Therefore the computational cost of POC function can be reduced by making the input signal $\frac{1}{s}$ times shorter.

V. Computational Cost And Effect of Aliasing

The number of complex multiplications of one-dimentional POC function can be reduced to $\frac{3N}{2s}\log Ns + \frac{N}{s}$ (see Fig.5). The case of $s = 1$ is just as that of conventional POC function. It can be seen that the number of complex multiplications become smaller as s become larger.

We evaluate the effect of aliasing on the peak of POC function that Subsampled-FFT causes. In this experiment we use two one-dimentional signals $\boldsymbol{x}$ and $\boldsymbol{x}_{\mathrm{WGN}} \in \mathbb{R}^{1024}$. Vector $\boldsymbol{x}$ is a row vector extracted from an image signal, and $\boldsymbol{x}_{\mathrm{WGN}}$ is defined by

$$x_{\mathrm{WGN}}(n) = x(n) + v_{\mathrm{WNG}}(n) \tag{6}$$

where $v_{\mathrm{WGN}}(n)$ is Gaussian white noise with zero mean. Then we compute POC function for $\boldsymbol{x}$ and $\boldsymbol{x}_{\mathrm{WGN}}$ with proposed method and evaluate the effect of aliasing according to the following formula:

$$\mathrm{RMS} = 10\log_{10} E\left[\left(\frac{r_s(0) - r(0)}{r(0)}\right)^2\right] \tag{7}$$

where $r(0)$ and $r_s(0)$ are the peaks of the POC functions because there is no displacement between $\boldsymbol{x}$ and $\boldsymbol{x}_{\mathrm{WGN}}$. The result of this experiment is shown in Fig.6. It can be seen that the number of RMS become worse as s become larger.

These results mean that there is an trade-off between the computational cost and the effect of aliasing.

VI. Conclusion

In this paper we have proposed a new computational method for reducing the computational cost of POC function. Our approach is based on reducing the number of frequency spectrum used for computing POC function. Although there is still an trade-off between the computational cost and the effect of aliasing on the peak, our method can reduce the computational cost efficiently without loss of accuracy if the division parameter s is chosen appropriately.

References

[1] H. Hagiwara and M. Kawamata, "Detection of Subpixel Displacement for Images Using Phase-Only Correlation," IEICE Technical Report. DSP, Vol. 101, No. 145, pp. 79-86, 2001.

[2] M. Hagiwara, M. Abe and M. Kawamata, "Estimation Method of Frame Displacement for Old Films Using Phase-Only Correlation," Journal of Signal Processing, Vol. 8, No. 5, pp. 421-429, 2004.

[3] A. Mouquit, T. Shibahara and T. Aoki, "A High-Accuracy Passive 3D Measurement System Using Phase-Based Image Matching," IEICE Trans. Fundamentals, Vol. E89-A, No.3, 2006.

[4] K. Ito, H. Nakajima, K. Kobayashi, T. Aoki and T. Higuchi, "A Fingerprint Matching Algorithm Using Phase-Only Correlation," IEICE Trans. Fundamentals, Vol. E87-A, No.3, 2004.