Digital Signal Processing 83 (2018) 83–97

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Fast computing position of maximum of circulant convolution

Sung-Hsien Hsieh [a,b], Chun-Shien Lu [a,*], Soo-Chang Pei [b]

[a] Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, ROC
[b] Graduate Institute of Communication Engineering, National Taiwan University, Taipei 106, Taiwan, ROC

ARTICLE INFO

Article history:
Available online 24 August 2018

Keywords:
Aliasing
Circulant matrix
Convolution
Cross-correlation
Time delay estimation

ABSTRACT

Computing the position of maximum of circulant convolution has been used for many applications in image and signal processing, and it usually is time-critical. Given the signal length N and the template size K, the conventional procedure requires $O(KN)$ operations. With $K \gg \log N$, this has been speeded by Fast Fourier Transform (FFT) with computation cost $O(N \log N)$.

This paper proposes a fast but heuristic scheme, returning only the position of maximum of convolution instead of the whole sequence after convolution. The main idea is to alias both the signal and template into lower dimensional space with the same dimension M being smaller than N and K, respectively. Thus, the computation cost is reduced to $O(N + M \log M)$ operations with $M = \Omega(\sqrt{N})$, where M is the only user-defined parameter and plays the trade-off between the computation cost and successfully returning the position of the maximum. To guide how to decide M, we show that the sufficient condition of successfully returning the position of the maximum depends on the relationship between the maximum convolution and remaining convolution results based on three different cases, i.e., $K \leq M$, $K > M$ with M exactly dividing N or not exactly dividing N. We further show how the probability of success can be analyzed if both the signal and template are random. Simulations validate the proposed scheme is fast and efficient, and they support the theoretical results. A case study with synchronization in global positioning system (GPS) is taken as a case study to demonstrate the applicability of our method.

1. Introduction

Computing the position of maximum of circulant convolution is a basic problem in signal processing. Given a template $\boldsymbol{t} \in \mathbb{R}^K$ and a signal $\boldsymbol{x} \in \mathbb{R}^N$, the problem is finding the shift of $\boldsymbol{t}$ corresponding to $\boldsymbol{x}$, leading to the maximum correlation between $\boldsymbol{x}$ and $\boldsymbol{t}$, as follows:

$$m = \underset{0 \leq k < N}{\operatorname{argmax}} \, (\boldsymbol{x} \circledast \boldsymbol{t}_{-N})_k, \tag{1}$$

where $(\boldsymbol{x})_k$ is the kth entry of $\boldsymbol{x}$, $\boldsymbol{t} = [(\boldsymbol{t})_0, \ldots, (\boldsymbol{t})_{K-1}]$,

$$\boldsymbol{t}_{-N} = [(\boldsymbol{t})_0, 0, \ldots, 0, (\boldsymbol{t})_{K-1}, (\boldsymbol{t})_{K-2}, \ldots, (\boldsymbol{t})_1] \in \mathbb{R}^N,$$

and $\circledast$ denotes circulant convolution with period N. This problem is similar to calculation of the maximum cross correlation [1], but only the position is required instead of also requiring the value of the maximum.

Solving Eq. (1) plays an important role in many applications. For example, synchronization in communication [2–5] and sound source localization [6–9] require time delay estimation [10–13]. Furthermore, when signals and templates are 2-dimensional (2D), a simple way is to vectorize both the input and template as the input of Eq. (1). By doing so, Eq. (1) can be extended to 2D cases. There are many applications involving 2D signals such as template matching [14,15], phase correlation [16,17], compression [18–20], and image registration [17,21]. Video tracking [22–27] usually requires running in real-time, where many tracking techniques like [26,27] utilize maximum cross correlation to find the location of objects. In other words, it has potential to accelerate tracking by reducing the computation cost of finding the position of maximum.

Some applications run in real time and are energy-critical. For example, synchronization in global positioning system (GPS) [3] requires real-time computation because the user needs the current position instead of an outdated one. Sound source localization for traffic control [28] often demands a short response time, especially when the object is moving. In addition, if the system is built into a mobile device [3,5], energy consumption is an important issue.

Thus, one main challenge of solving Eq. (1) is avoiding high computation time. Exhaustive search (full search) is conducted by checking $(\boldsymbol{x} \circledast \boldsymbol{t}_{-N})_k$ from $k = 0$ to $k = N - 1$, which costs $O(KN)$

* Corresponding author.
E-mail address: lcs@iis.sinica.edu.tw (C.-S. Lu).

https://doi.org/10.1016/j.dsp.2018.08.009

and is unacceptably slow. Traditionally, convolution is speeded up by fast Fourier Transform (FFT) [11,14] as follows:

$$m = \underset{0\le k<N}{\operatorname{argmax}}\,(\boldsymbol{r})_k \text{ with } \boldsymbol{r} = \mathcal{F}^{-1}\left(\mathcal{F}(\boldsymbol{x}).*\mathcal{F}(\boldsymbol{t}_{-N})\right), \tag{2}$$

where ".∗" is Hadamard product and $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote FFT and inverse FFT (IFFT), respectively. The computation cost, bounded by FFT, is $O(N\log N)$. When $K<N$, zeros must be padded into $\boldsymbol{t}$ until lengths of $\boldsymbol{x}$ and $\boldsymbol{t}$ are equal, and it still costs $O(N\log N)$.

Many techniques, summarized in the following, have been developed to reduce the computation time.

- Coarse-to-fine strategy [15,29,30]: First, a coarse search is conducted by finding the downsampled template in the downsampled signal to yield a good match with reduced computational overhead. Then, a fine search is conducted in the original space starting from the neighborhood of the best match found in coarse search.
- Full search-equivalent strategy [31–33]: It employs a rejection mechanism such that the current search is terminated as soon as some criteria are satisfied.
- FFT-based fast convolution [3,14]: Eq. (1) can be done quickly by sparse FFT or FFT.

It should be noted that most of the techniques focus on template matching instead of Eq. (1) directly. Nevertheless, template matching is equivalent to solving Eq. (1) by adopting cross-correlation as the similarity measure.

Recently, based on the assumption that $\boldsymbol{r}$ is approximately 1-sparse,[1] Hassanieh et al. [3] proposed replacing $\mathcal{F}^{-1}$ by sparse FFT (sFFT), which is faster than FFT for the sparse Fourier transform (SFT) problem. Instead of solving $\boldsymbol{r}$ directly, sFFT solves aliased $\boldsymbol{r}$ by downsampling $\mathcal{F}(\boldsymbol{x}).*\mathcal{F}(t)$ as input, where aliasing a signal is defined as follows.

Definition 1. (Aliasing a signal as N is divisible by M) [3] Suppose a signal $\boldsymbol{x}\in\mathbb{R}^N$ and $M\in\mathbb{N}$, which exactly divides N. The aliased signal $\boldsymbol{x}_M\in\mathbb{R}^M$ via an aliasing factor M in the time domain is defined as:

$$(\boldsymbol{x}_M)_k = \sum_{i=0}^{\frac{N}{M}-1}(\boldsymbol{x})_{k+iM}.$$

By the fact that aliasing a signal in the time domain is equivalent to downsampling it in the frequency domain and *vice-versa*, we know that, when the input of sFFT, i.e., $\mathcal{F}(\boldsymbol{x}).*\mathcal{F}(t)$, is downsampled, its output is aliased in the time domain accordingly. Thus, the position of maximum is determined by checking all possible shifts and picking the shift that gives the maximum correlation. Hassanieh et al.'s method requires that N be divisible by the size of the aliased signal. In addition, the computation cost requires $O(N\sqrt{\log N})$ or $O(N)$ operations, depending on the variance of noise.

1.1. Our contributions

Traditionally, solving Eq. (1) requires one to compute all N values of $(\boldsymbol{x}\circledast\boldsymbol{t}_{-N})_k$'s before finding the true position m of the maximum. Instead of wasting time to compute all values, in this paper, we propose a fast but somewhat heuristic scheme to find the candidate positions, including m, before calculating the value of $(\boldsymbol{x}\circledast\boldsymbol{t}_{-N})_m$.

Our method employs a circulant matrix to alias the signal first, and subsequent computations are conducted in the lower dimension space. Thus, when finding the candidate positions, no computation in the higher dimension space is required. This results in the computation overhead being $O(N+M\log M)$ with $M=\Omega(\sqrt{N})$. When $M<\frac{N}{\log N}$, the complexity is bounded by $O(N)$. All we need to do is decide M, which plays the trade-off between the successful match and computation cost (see Theorem 6). In fact, our method can be considered to be a generalization of [3] with the main results and differences summarized as follows.

1. We show how to find the candidate positions, including the ground truth m, via aliased signal and aliased template even when the length of the original signal is not divisible by that of the aliased signal. Specifically, we derive the sufficient conditions of successfully finding the position of maximum under different parameter settings: (i) $K\le M$ with M not exactly dividing N, (ii) $K>M$ with M not exactly dividing N, and (iii) any K with M exactly dividing N. Our results characterize the relationship among N, M, the maximum convolution $(\boldsymbol{r})_m$, and remaining convolution results $(\boldsymbol{r})_i$'s for all $i\ne m$. Compared with [3], their method is based on the assumption that M must exactly divide N.
2. [3] requires checking all possible $O(\frac{N}{M})$ shifts (candidate positions), including the position of the maximum. The authors show that $O(M)$ multiplications and additions are needed to check each shift only if both the signal and template are binary random. Thus, the lowest computation cost is dominated by $O(N)$ multiplications. In contrast, our algorithm exploits the commutative property between circulant matrices and employs two aliased $\boldsymbol{r}$'s with different aliasing patterns based on Chinese Remainder Theorem (CRT) to directly identify the position of maximum without calculating all values of candidate positions such that the number of candidate positions can be reduced further to $O(1)$. We show that only $O(M)$ multiplications are required in checking all candidate positions; thus, our scheme is dominated by $O(N)$ additions, which come from the aliased signal. Furthermore, checking each shift, in general, requires $O(N)$ operations without any assumptions about the signal and template. Under this situation, [3] and our method are dominated by $O(\frac{N^2}{M})$ and $O(N)$ multiplications, respectively.
3. We take GPS as a critical application to compare with [3] under noisy environments. We prove that our method is effective for find the position of maximum with probability of success being larger than $1-o(1)$ and computational cost being $O(N)$ additions when the variance of noise is less than $\frac{c(N)M}{\ln N}$ with $c(N)=o(1)$ and $M\le\frac{N}{\log N}$, implying that our scheme has comparable performance with [3] under the condition that the probability of success converges to 1 with the same noise variance.

We further clarify the difference between the proposed method and previous SFT works based on CRT [34–37]. First, the SFT problem begins to have $\mathcal{F}(\boldsymbol{x}).*\mathcal{F}(t)$ as inputs instead of $\boldsymbol{x}$ and $\boldsymbol{t}$ in our study work. In other words, the SFT problem originally assumes that the required samples for solving $\boldsymbol{r}$ are known. In this paper, however, the required samples are unknown at the beginning. Thus, the computation cost for generating samples, which is ignored in CRT-based SFT methods, needs to be considered. Second, the SFT methods based on CRT suppose that there is greater than or equal to one divisor of N, where the number of divisors is algorithm-dependent. Nevertheless, our method can work even

[1] A vector contains only an entry with magnitude larger than other entries that are close to zero.