# An Efficient Algorithm for Continuous-time Cross Correlogram of Spike Trains

Il Park * António R. C. Paiva Thomas B. DeMarse
José C. Príncipe

*University of Florida, Gainesville, FL, USA 32611*

**Abstract**

We propose an efficient algorithm to compute the smoothed correlogram for the detection of temporal relationship between two spike trains. Unlike the conventional histogram based correlogram estimations, the proposed algorithm operates on continuous time without binning the spike train nor the correlogram. Hence it can be more precise in detecting the effective delay between two recording sites. Moreover, it can take advantage of the higher temporal resolution of the spike times provided by the current recording methods. The Laplacian distribution kernel for smoothing enables efficient computation of the algorithm. We also provide the basic statistics of the estimator and a guideline for choosing the kernel size. This new technique is demonstrated by estimating the effective delays in a neuronal network from synthetic data and recordings of dissociated cortical tissue.

(`$Id: cipogram.tex 298 2007-10-22 19:04:45Z arpaiva $`)

*Key words:* cross correlation, correlogram, delay estimation, kernel intensity estimation, point process

## 1 Introduction

Precise time delay in transmission of a spike in the neural system is considered to be one of the key requirements for efficient computation in the cortex (Maass, 1997; Mainen and Sejnowski, 1995; Reinagel and Reid, 2000; VanRullen et al., 2005). One of the effective methods for estimating the delay is

* Corresponding author
*Email addresses:* `memming@cnel.ufl.edu` (Il Park), `arpaiva@cnel.ufl.edu` (António R. C. Paiva), `tdemarse@bme.ufl.edu` (Thomas B. DeMarse), `principe@cnel.ufl.edu` (José C. Príncipe).

to use a cross correlogram (Perkel et al., 1967). The cross correlogram is a basic tool to analyze the temporal structure of signals. It is widely applied in neuroscience to assess oscillation, propagation delay, effective connection strength, synchronization, and spatiotemporal structure of a network (Brown et al., 2004; König et al., 1995).

When applied to a pair of action potential trains, the cross correlogram is the superposition of the relative firing time differences. The peaks and troughs of the cross correlogram are of most interest. These features can be interpreted as a functional connectivity among neurons (Aertsen et al., 1989). The shift of the features can be interpreted as the communication delay (Nikolić, 2007), and the width of the features as temporal scale of the correlation (Brivanlou et al., 1998). Since the estimation requires averaging over a period of time, stationarity needs to be assumed during the period of analysis, and certain non-stationarities are known to induce spurious features (Brody, 1999). Also, common input from an unobserved source with delays cannot be distinguished from direct influences (Nykamp, 2005). There exist variations and extensions of cross correlation including joint peri-stimulus time histogram (JPSTH) (Aertsen et al., 1989), cross intensity function (Hahnloser, 2007), and conditional firing probability (le Feber et al., 2007).

However, estimating the cross correlation of spike trains is more intricate since spike trains are point processes. Cross correlation is well defined and intuitively understood for continuous and discrete-time random signals due the concept of amplitude, yet spike trains do not have amplitude but are characterized only by time instances at which the spikes occur. A well known algorithm for estimating the correlogram from point processes involves histogram construction with time interval bins (Dayan and Abbott, 2001). The binning process is effectively transforming the uncertainty in time to the amplitude variability. This quantization of time introduces binning error that leads to coarse time resolution. Furthermore, the correlogram does not take advantage of the higher temporal resolution of the spike times provided by current recording methods.

This can be improved using smoothing kernels to estimate the cross correlation function from a finite number of spikes. The resulting cross correlogram is continuous and provides high temporal resolution in the region where there is a peak (see Fig. 1 for a comparison between the histogram and kernel method.) As mentioned earlier, the peaks in a cross correlogram can be correctly interpreted as long as the signal is stationary. In general, spike trains are non-stationary, especially when time-varying input is presented to the neural system. Therefore, caution is required when analyzing cross correlations (Brody, 1999). Nevertheless, occasionally, stationarity can be assumed during a short period of time, requiring a windowed analysis. This means that data effective methods are needed, capable of achieving high accuracy with

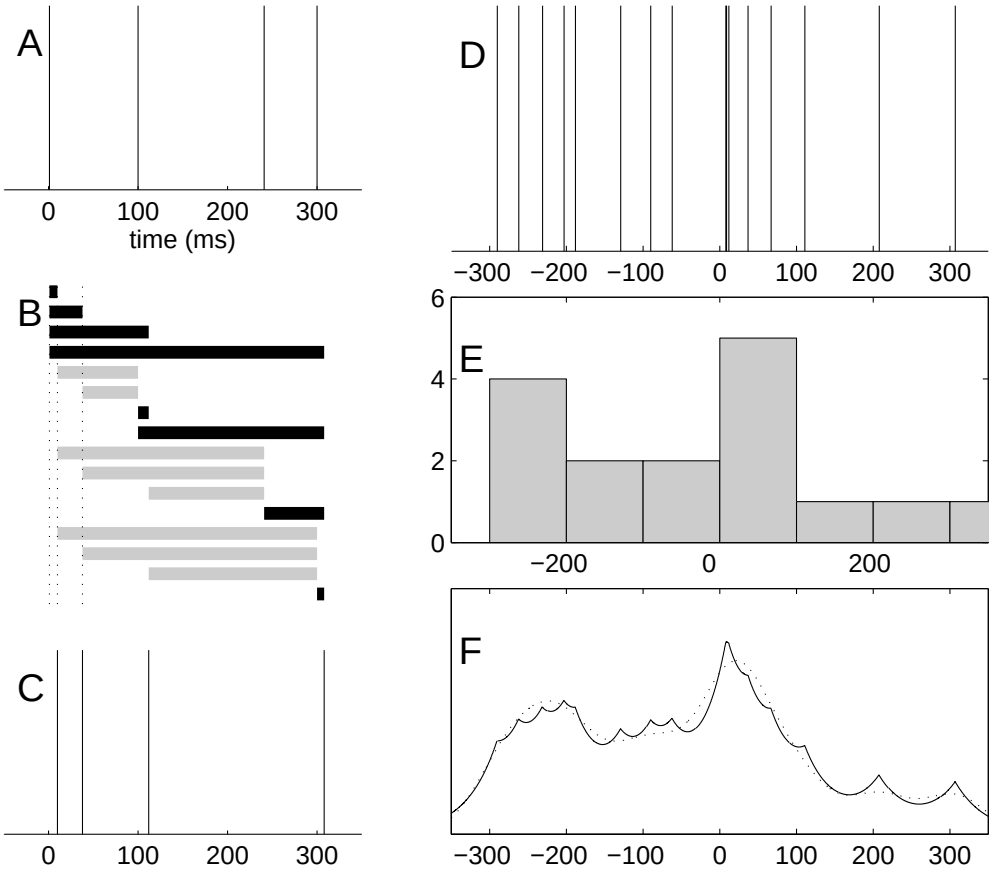



Fig. 1. Example of cross correlogram construction. A and C are two spike trains each with 4 spikes. Each spike in A invokes a spike in C with some small delay around 10 ms except for the third spike in A. B represents all the positive (black) and negative (gray) time differences between the spike trains. D shows the position of delays obtained in B. E is the histogram of D, which is the conventional cross correlogram with bin size of 100 ms. F shows the continuous cross correlogram with Laplacian kernel (solid) and Gaussian kernel (dotted) with bandwidth 40 ms. Note that the Laplacian kernel is more sensitive to the exact delay.

very small amounts of data. Given the sparse nature of the spike trains, the conventional cross correlogram is unsatisfactory in this scenario.

In this paper, we propose an efficient binless algorithm which can be used to detect interaction delay between spike trains in continuous time. We achieve this by estimating the continuous correlogram of spike trains using a smoothing kernel (section 2). However, unlike the smoothed cross correlation histogram already used, we exploit the properties of a specific type of kernel to determine immediately all possible local maxima of the correlogram and optimize the computation significantly (section 3). The performance of the proposed and other existing methods are compared in terms of precision and computation time (section 4.1). In addition, the effectiveness of the algorithm is demonstrated with a model, and *in vitro* recordings (section 4.2).

## 2 Continuous Correlogram

In this section, we review the cross correlogram and explain how a smoothed cross correlogram is obtained. Two simultaneously recorded instances of point processes are represented as a sum of Dirac delta functions at the time of firing event, $s_i(t)$ and $s_j(t)$,

$$s_i(t) = \sum_{m=1}^{N_i} \delta(t - t_m^i), \tag{1}$$

where $N_i$ is the number of spikes and $t_m^i$ is the time of the $i$-th action potential. Assuming wide sense stationarity (at least piecewise) and ergodicity, the cross correlation function can be defined as,

$$Q_{ij}^{\dagger}(\Delta t) = E_t \left[ s_i(t) s_j(t + \Delta t) \right], \tag{2}$$

where $E_t[\cdot]$ denotes expected value over time $t$. The cross correlation can be interpreted as the scaled conditional probability of $j$-th neuron firing given $i$-th neuron fired $\Delta t$ seconds before (Knox, 1974). In a physiological context, there is a physical restriction of propagation delay for an action potential to have a causal influence to invoke any other action potential. Therefore, this delay would influence the cross correlogram in the form of increased value. Thus, estimating the delay involves finding the lag at which there is a maximum in the cross correlogram (inhibitory interaction, which appear as troughs rather than peaks, is not considered in this article).

Smoothing a point process is superior to the histogram method for the estimation of the intensity function (Nawrot et al., 1999), and especially the maxima (Kass et al., 2003). The cross correlation function can also be estimated better with smoothing, which is done in continuous time, so we do not lose the exact time of spikes while enabling interaction between spikes at a distance.

Instead of smoothing the histogram of time differences between two spike trains, we could first smooth the spike train to obtain a continuous signal (Gerstner, 2001). This is not done explicitly because, as we will show, this is equivalent to smoothing the time differences with a different kernel. A causal exponential decay was chosen as the smoothing kernel to achieve computational efficiency (see section 3). This result holds for any non-negative valued smoothing kernel for the spike train, but the computational advantage may disappear. The inverse relation, finding the smoothing kernel for the spike train given a histogram smoothing function, can also be found by spectral decomposition techniques, but the solution may not be unique. In the case considered here, smoothed spike trains are represented as,

$$q_i(t) = \sum_{m=1}^{N_i} \frac{1}{\tau} e^{-\frac{t - t_m^i}{\tau}} u(t - t_m^i), \tag{3}$$

where $u(t)$ is the unit step function and $\tau$ is the time constant (kernel size). The cross correlation function of the smoothed spike trains is,

$$Q^*_{ij}(\Delta t) = E_t \left[ q_i(t) q_j(t + \Delta t) \right]. \tag{4}$$

Given a finite length of observation, the expectation in (4) can be estimated from samples as,

$$\hat{Q}^*_{ij}(\Delta t) = \frac{1}{T} \int_0^\infty q_i(t) q_j(t + \Delta t) dt, \tag{5}$$

where $T$ is the length of the observation. After evaluation of the integral, the resulting estimator becomes,

$$\hat{Q}^*_{ij}(\Delta t) = \frac{1}{2\tau T} \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} e^{-\frac{|t^i_m - t^j_n - \Delta t|}{\tau}}, \tag{6}$$

which is equivalent to the kernel intensity estimation (Diggle and Marron, 1988; Parzen, 1962) from time differences using a Laplacian distribution kernel.

To determine the significance of a correlation, the mean and variance of the estimator is analyzed assuming the spike trains are realizations of two independent homogeneous Poisson processes (null hypothesis).

$$E\left[\hat{Q}^*_{ij}(\Delta t)\right] \simeq \lambda_I \lambda_J, \tag{7}$$

$$\mathrm{var}(\hat{Q}^*_{ij}(\Delta t)) \simeq \frac{\lambda_I \lambda_J}{4\tau T}, \tag{8}$$

where $\lambda_I$ and $\lambda_J$ denote the firing rate of the Poisson processes, corresponding realizations are $i$-th and $j$-th spike train, respectively (see Appendix for the derivation). Note that the variance reduces linearly as the duration of the spike train is elongated. This was an expected result because more information is available for the estimation. Based on these quantities, we standardize the measure for inter-experiment comparison by removing the mean and dividing by the standard deviation:

$$\tilde{Q}_{ij}(\Delta t) = \frac{\sqrt{4\tau T}(Q_{ij}(\Delta t) - \lambda_I \lambda_J)}{\sqrt{\lambda_I \lambda_J}}. \tag{9}$$

All the figures of cross correlograms in this paper are standardized, that is, the y-axis in Figs. 6, 7 and 8 can be interpreted as the deviation from the null hypothesis in the scale of standard deviations.

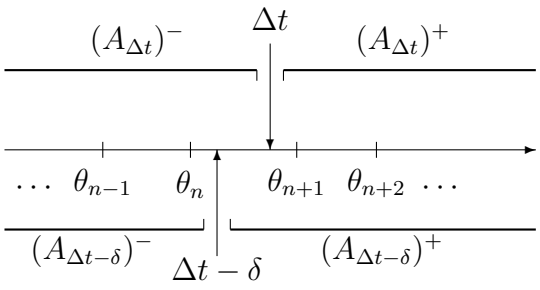



Fig. 2. Illustration of the decomposition and shift of the multiset $A$ (see (11)). $\{\theta_i\}$ are the time differences between spike timing across spike trains as in Fig. 1D.

## 3 Algorithm

In the following, an efficient algorithm to compute the continuous cross correlogram peaks is presented. It is also shown that there are only finite possible local maxima for the continuous cross correlogram, therefore the computation on continuous time can be done with finite operations. The algorithm divides the computation of the summation part of the continuous cross correlogram into disjoint regions and combines the result. By storing the intermediate computation results for all time lags, the cross correlation of each lag can be computed in constant time using its neighboring values. Therefore, the total computation time will depend on the number of lags and the sorting cost.

The essential quantity to be computed is the following double summation,

$$Q_{ij}(\Delta t) = \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} e^{-\frac{|t_m^i - t_n^j - \Delta t|}{\tau}}. \tag{10}$$

The basic idea for efficient computing is that a single multiplication results in shifting the collection of points for a summation of exponential functions evaluated at these points, that is, $\sum_i e^{x_i+\delta} = (\sum_i e^{x_i})\, e^{\delta}$. This is the fundamental reason for the choice of this particular kernel. Now, since a Laplacian kernel can be seen as two exponentials stitched together, we need only to carefully separate the computation of the two sides of the kernel.

Let us define the multiset of all time differences between two spike trains,

$$A = \{\theta \mid \theta = t_m^i - t_n^j,\ m = 1, \ldots, N_i,\ n = 1, \ldots, N_j\}. \tag{11}$$

Even though $A$ is not strictly a set, since it may contain duplicates, we will abuse the set notation for simplicity. Note that the cardinality of the multiset $A$ is $N_i N_j$. Now (10) can be rewritten as

$$Q_{ij}(\Delta t) = \sum_{\theta \in A} e^{-\frac{|\theta - \Delta t|}{\tau}}. \tag{12}$$

Now let us define a series of operations for a multiset $B \subset \mathbb{R}$ and $\delta \in \mathbb{R}$,

$$B^+ = \{x \mid x \in B \quad \text{and} \quad x \geq 0\}, \qquad \textit{(non-negative lag)} \tag{13a}$$
$$B^- = \{x \mid x \in B \quad \text{and} \quad x < 0\}, \qquad \textit{(negative lag)} \tag{13b}$$
$$B_\delta = \{x \mid y \in B \quad \text{and} \quad x = y - \delta\}. \qquad \textit{(shift)} \tag{13c}$$

Since $B$ can be decomposed into two exclusive sets $B^+$ and $B^-$, (12) can also be rewritten and decomposed as,

$$Q_{ij}(\Delta t) = \sum_{\theta \in A_{\Delta t}} e^{-\frac{|\theta|}{\tau}} = \sum_{\theta \in (A_{\Delta t})^+} e^{-\frac{|\theta|}{\tau}} + \sum_{\theta \in (A_{\Delta t})^-} e^{-\frac{|\theta|}{\tau}} \tag{14a}$$
$$= \sum_{\theta \in (A_{\Delta t})^+} e^{-\frac{\theta}{\tau}} + \sum_{\theta \in (A_{\Delta t})^-} e^{\frac{\theta}{\tau}}. \tag{14b}$$

For convenience, we define the following summations

$$Q_{ij}^{\pm}(\Delta t) = \sum_{\theta \in (A_{\Delta t})^{\pm}} e^{\mp \frac{\theta}{\tau}}. \tag{15}$$

Let us sort the multiset $A$ in ascending order and denote the elements as $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_n \leq \theta_{n+1} \leq \ldots \theta_{N_i N_j}$. Observe that within an interval $\Delta t \in (\theta_n, \theta_{n+1}]$, the multiset $((A_{\Delta t})^{\pm})_{-\Delta t}$ is always the same (see Fig. 2). In other words, if $\Delta t = \theta_{n+1}$, for a small change $\delta \in [0, \theta_{n+1} - \theta_n)$, the multisets do not change their membership, i.e. $((A_{\Delta t})^{\pm})_\delta = (A_{(\Delta t - \delta)})^{\pm}$. Therefore, we can simplify an arbitrary shift of $Q_{ij}^{\pm}$ with a single multiplication of an exponential as,

$$Q_{ij}^{\pm}(\Delta t - \delta) = \sum_{t \in (A_{\Delta t - \delta})^{\pm}} e^{\mp \frac{t}{\tau}} \qquad = \sum_{t \in ((A_{\Delta t})^{\pm})_\delta} e^{\mp \frac{t}{\tau}} \tag{16a}$$
$$= \sum_{t \in (A_{\Delta t})^{\pm}} e^{\mp \frac{t-\delta}{\tau}} \qquad = \sum_{t \in (A_{\Delta t})^{\pm}} e^{\mp \frac{t}{\tau}} e^{\pm \frac{\delta}{\tau}} = Q_{ij}^{\pm}(\Delta t) e^{\pm \frac{\delta}{\tau}}. \tag{16b}$$

Thus, local changes of $Q_{ij}$ can be computed by a constant number of operations no matter how large the set $A$ is, so that

$$Q_{ij}(\Delta t - \delta) = Q_{ij}^{+}(\Delta t - \delta) + Q_{ij}^{-}(\Delta t - \delta) \tag{17a}$$
$$= Q_{ij}^{+}(\Delta t) e^{\frac{\delta}{\tau}} + Q_{ij}^{-}(\Delta t) e^{-\frac{\delta}{\tau}}. \tag{17b}$$

Now we can compute $Q_{ij}(\cdot)$ on a set of points efficiently, but on which points should it be computed? Can we find the local maxima and minima that we are mostly interested in? Indeed, it can be shown that $Q_{ij}(\cdot)$ has all its maxima on the points on $A$. If there is a local maximum or minimum of $Q_{ij}(\Delta t - \delta)$, it would be where $\frac{dQ_{ij}(\Delta t - \delta)}{d\delta} = 0$, which is,

$$\delta^* = \frac{\tau}{2} \left( \ln(Q_{ij}^{-}(\Delta t)) - \ln(Q_{ij}^{+}(\Delta t)) \right). \tag{18}$$

Also note that since the second derivative,

$$\frac{d^2 Q_{ij}(\Delta t - \delta)}{d\delta^2} = \frac{1}{\tau^2}\left(Q_{ij}^{+}(\Delta t)e^{\frac{\delta}{\tau}} + Q_{ij}^{-}(\Delta t)e^{-\frac{\delta}{\tau}}\right) \geq 0, \tag{19}$$

$Q_{ij}(\Delta t - \delta)$ is a convex function of $\delta$ within the range. Thus, the maximum of the function value is always on either side of its valid range, only local minimum can be in between.

In principle, we need to compute (10) for all $\Delta t \in [-T^*, T^*]$ to achieve continuous resolution, where $T^*$ is the maximum time lag of interest. However, if we only want all local minima and maxima, we just need to evaluate on all $\Delta t \in A$, and compute the minima and maxima using (17b) and (18). Therefore, if we compute the $Q_{ij}^{\pm}(\theta_n)$ for all $\theta_n \in A$, we can compute $\delta^*$ for all intervals $(\theta_n, \theta_{n+1}]$ if a local extremum exists. These can be computed using the following recursive formulae,

$$Q_{ij}^{-}(\theta_{n+1}) = Q_{ij}^{-}(\theta_n)e^{-\frac{\theta_{n+1}-\theta_n}{\tau}} + 1, \tag{20a}$$

$$Q_{ij}^{+}(\theta_{n+1}) = Q_{ij}^{+}(\theta_n)e^{\frac{\theta_{n+1}-\theta_n}{\tau}} - 1. \tag{20b}$$

In practice, due to accumulation of numerical error, the following form is preferable for $Q_{ij}^{+}$,

$$Q_{ij}^{+}(\theta_n) = (Q_{ij}^{+}(\theta_{n+1}) + 1)e^{-\frac{\theta_{n+1}-\theta_n}{\tau}}. \tag{21}$$

Initial conditions for the recursions are $Q_{ij}^{-}(\theta_1) = 1$ and $Q_{ij}^{+}(\theta_N) = 0$. The resulting pseudocode is listed as in Algorithm 1. [1]

The bottleneck for time complexity is the sorting of the multiset $A$, thus the overall time complexity is $\mathcal{O}(N_i N_j \log(N_i N_j))$, where $\mathcal{O}$ is the big-O notation for the asymptotic upper bound. [2] Note that the time complexity of direct evaluation of (10) is $\mathcal{O}(N_i N_j)$ for *each time lag* $\Delta t$, which makes it $\mathcal{O}((N_i N_j)^2)$ for *all time lags*. Assuming homogeneous Poisson process for individual spike trains, the average time complexity becomes $\mathcal{O}(N^* \log N^*)$ with $N^* = \lambda_I \lambda_J T$, where $T$ is the length of spike train and $\lambda_I$ represents the average firing rate for the Poisson process. The conventional cross correlogram algorithm (Dayan and Abbott, 2001) has the time complexity of $\mathcal{O}(N^*)$ on average, which is slightly better than the proposed algorithm.

A naïve way to compute the smoothed histogram is to directly apply kernels to the data points and discretizing (as in Fig. 1 D to F). If evaluated in

[1] The MATLAB (The MathWorks, Inc.) implementation of the algorithm is available at `http://www.cnel.ufl.edu/~memming/research/ccc.html`.

[2] It is possible to reduce the sorting to $\mathcal{O}(N_i N_j \log(\min(N_i, N_j)))$ using merge sorting partially sorted lists. However, it is only a minor improvement in general.

**Algorithm 1** Calculate $Q_{ij}$

**Require:** $\tau > 0, A \neq \emptyset, N = |A|$

**Ensure:** $Q_{ij}(\Delta t) = \sum_{t \in A} e^{-\frac{|t-\Delta t|}{\tau}}, \forall \Delta t \in A$

$\mathbf{A} \Leftarrow \text{sort}(A)$ $\{O(N \log N)\}$
$Q^-(1) \Leftarrow 1$
$Q^+(N) \Leftarrow 0$
**for** $k = 1$ to $N - 1$ **do**
    $ed(k) \Leftarrow e^{-\frac{\mathbf{A}(k+1)-\mathbf{A}(k)}{\tau}}$
**end for**
**for** $k = 1$ to $N - 1$ **do**
    $Q^-(k+1) \Leftarrow 1 + Q^-(k) \cdot ed(k)$
    $Q^+(N-k) \Leftarrow (Q^+(N-k+1)+1) \cdot ed(N-k)$
**end for**
**for** $k = 1$ to $N$ **do**
    $Q_{ij}(\mathbf{A}(k)) \Leftarrow Q^+(k) + Q^-(k)$
**end for**

| Method | Time complexity | Space complexity |
|---|---|---|
| CCH | $\mathcal{O}(N^*)$ | $\mathcal{O}(\frac{T^*}{\gamma})$ |
| CCC | $\mathcal{O}(N^* \log N^*)$ | $\mathcal{O}(N^*)$ |
| CCSH | $\mathcal{O}(\frac{N^* T^*}{\gamma})$ | $\mathcal{O}(\frac{T^*}{\gamma})$ |

Table 1
Average time and space complexity of different methods. $N^*$ is the expected number time differences in the window of $[-T^*, T^*]$, $\gamma$ is the bin size for discretizing the time difference. Notice that because the CCC does not force a discretization on the time domain it does not depend on the bin size $\gamma$. If $N^* > T^*/\gamma$, the space complexity (memory requirement) of CCC is worse than the other two algorithms, but the time complexity is still better than the cross correlation smoothed histogram (CCSH) and always worse than the cross correlation histogram (CCH).

continuous time, this is equivalent to (6). However, discrete point evaluation is required for representation, therefore the result is quantized in the form of histogram we call cross correlation smoothed histogram (CCSH). The advantage of CCSH is that any type of kernel, such as Gaussian or polynomial kernel, can be used. The time complexity between CCSH, the proposed algorithm (denoted CCC, for continuous cross correlogram estimator), and the histogram method (denoted CCH, for cross correlation histogram) is given in table 1. Computer simulation results will be shown in the next section where these three algorithms, CCC, CCH and CCSH, are compared for computation time and precision in table 2.

## 4 Results

In this section, we analyze the statistical properties and demonstrate the usefulness of the continuous cross correlogram estimator (CCC) compared to the cross correlation histogram (CCH). The CCC is defined by the linear interpolation of (9) between the possible maxima (but not the minima). CCH discretizes the time difference between spikes rather then the time of action potentials, which is superior on error performance. In order to compare with CCC, CCH is standardized in a similar way to (9) with estimators for mean and variance according to Palm et al. (1988).

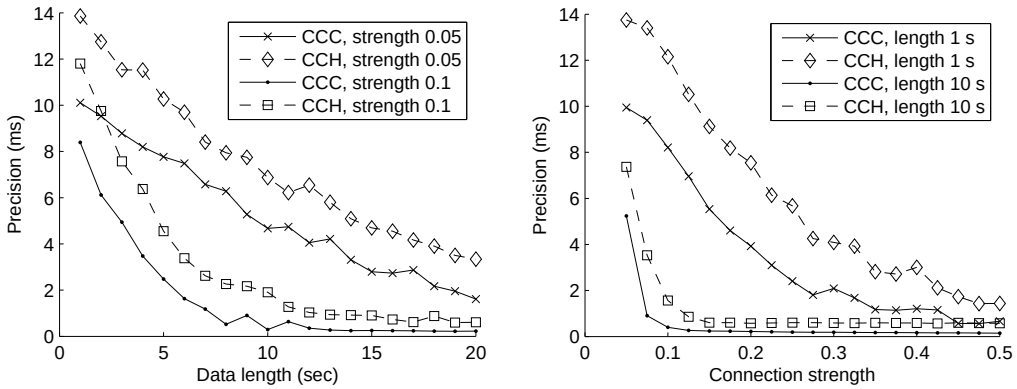
Since CCH is essentially equivalent to using a uniform distribution kernel (or a boxcar kernel) and sampling at equally spaced intervals (of the time differences) as opposed to the Laplacian distribution kernel used in CCC, in order to make a fair comparison we choose the kernel size (bin size) of both distributions to have the same standard deviation. To be specific, if the time bin size of CCH is $h$, then we compare the result to CCC with kernel size of $\tau = \frac{h}{2\sqrt{6}}$.

Moreover, since the histogram method is highly sensitive to bin size, we used the procedure of optimal bin size selection of Poisson processes suggested by Shimazaki and Shinomoto (2006). The method is designed for the estimation of firing rate or PSTH from a measurement assuming a Poisson process. However, since the time difference between two Poisson processes of finite length can be considered as a realization of a Poisson process, it is possible to directly apply it to the CCH.

### *4.1 Analysis*

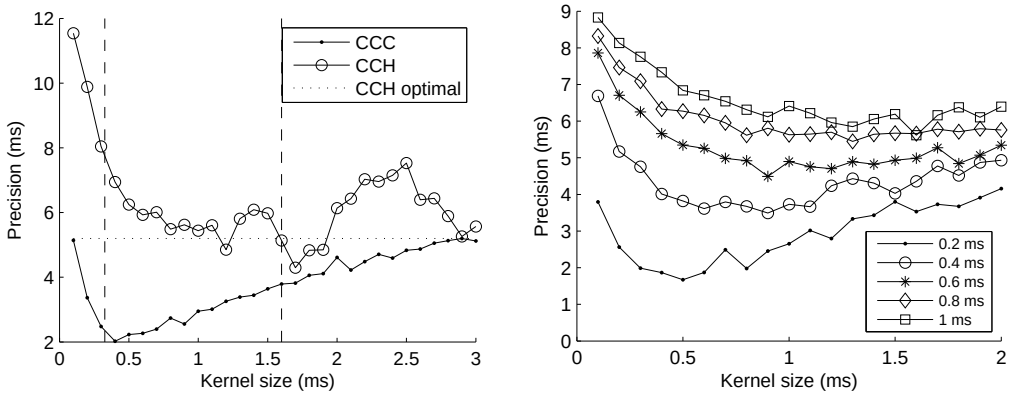
For a pair of directly synapsing neurons, the delay from the generation of an action potential of the presynaptic neuron to the generation of an action potential of the post synaptic neuron is not always precise. Various sources of noise such as variability in axon conduction delay, presynaptic waveform, probability of presynaptic vesicle release, and threshold mechanism (Sabatini and Regehr, 1999) effect the location, significance and width of the cross correlogram peak. Furthermore, if the neurons are in a network, multiple paths, common input sources, recurrent feedback and local field potential fluctuation can influence the cross correlogram.

In this section, we model the timing jitter with a Gaussian distribution and analyze the statistical properties of CCC and CCH on time delay estimation. A pair of Poisson spike trains of firing rate 10 spikes/s were correlated by copying a portion of the spikes from one to another and then shifting by the




(a) Effect of spike train length (b) Effect of connection strength

Fig. 3. Effect of the length of spike train and strength of connectivity on precision of delay estimation. The precision is estimated by the standard deviation in 1000 Monte Carlo runs with kernel size $\tau = 0.4$ ms (or bin size $h = 1.96$ ms). (Smaller precision indicates higher temporal resolution.)




(a) CCC vs CCH (b) Optimal $\tau$ for CCC

Fig. 4. Effect of kernel size (bin size) of CCC (CCH) to the performance. The connection strength was 5% and the spike trains are 10 seconds long, i.e. 5 spikes are correlated on average. (a) Sensitivity of CCC and CCH on kernel size for noise standard deviation 0.25 ms. The horizontal dotted line indicates the performance when optimal bin size is chosen for each set of simulated spike time differences. The median of the optimal bin size chosen (right) and corresponding kernel size for CCC (left) are plotted as vertical dashed lines. Note that CCC is more robust to the kernel size selection and always performs better than CCH. (b) For different standard deviations of jitter noises, the precision is plotted versus the kernel size $\tau$. Note that the optimal kernel size increases as the jitter variance increases. For each point, 3000 Monte Carlo runs are used, and the actual delay is uniformly distributed from 3 ms to 4 ms to reduce the bias of CCH.

delay with the Gaussian jitter noise. The fraction of spikes copied represents the effective synaptic connectivity.

The total number of correlated spikes depend on two factors: the length of

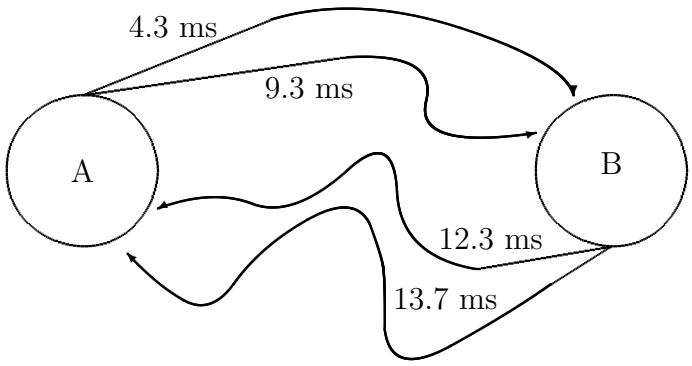



Fig. 5. Schematic diagram for the configuration of neurons.

spike train, and the synaptic connectivity. In figure 3, the precision of CCC and CCH are compared according to these factors. The precision is defined to be the standard deviation of the error in estimating the exact delay. Precision of both CCC and CCH improves as the number correlated spikes increases in a similar trend. CCC converges to a precision lower than half the jitter noise standard deviation (500 $\mu$s).

The optimal kernel size (or bin size) which gives the best precision depends on the noise jitter level. In figure 4(a), CCC and CCH is compared across different kernel sizes. In general, CCC performs better than optimal bin size and most of the bin sizes CCH. As mentioned above, CCH is sensitive to bin size, but CCC is robust to the kernel size for precision performance. Also note that the optimal kernel size for CCC corresponds to equal median value of the variance optimal bin size selected (vertical dash lines). Increasing the jitter level worsens the best precision and increases the optimal kernel size for CCC as shown in Fig. 4(b).

The computational time is also compared by using the same task. Table 2 shows the results for the computer simulation. The same Laplacian distribution kernel is used to compute the CCSH. We can observe that CCC is faster than CCSH almost always, and always better in precision. This is obvious from the fact that CCSH is a discretization of CCC. In fact, even in the most favorable situation for CCSH, CCC shows better precision with $8 \sim 200$ times less computation time. As expected, CCH is always faster than CCC but always less precise. This is most notorious for small amounts of data where the precision with CCH clearly depends on the bin size.

### *4.2 Examples*

In this section, we demonstrate the power of CCC using two examples: the first example uses synthetic spike trains from a simple spiking neuronal network model, and for the second we use recordings from a cortical culture on a

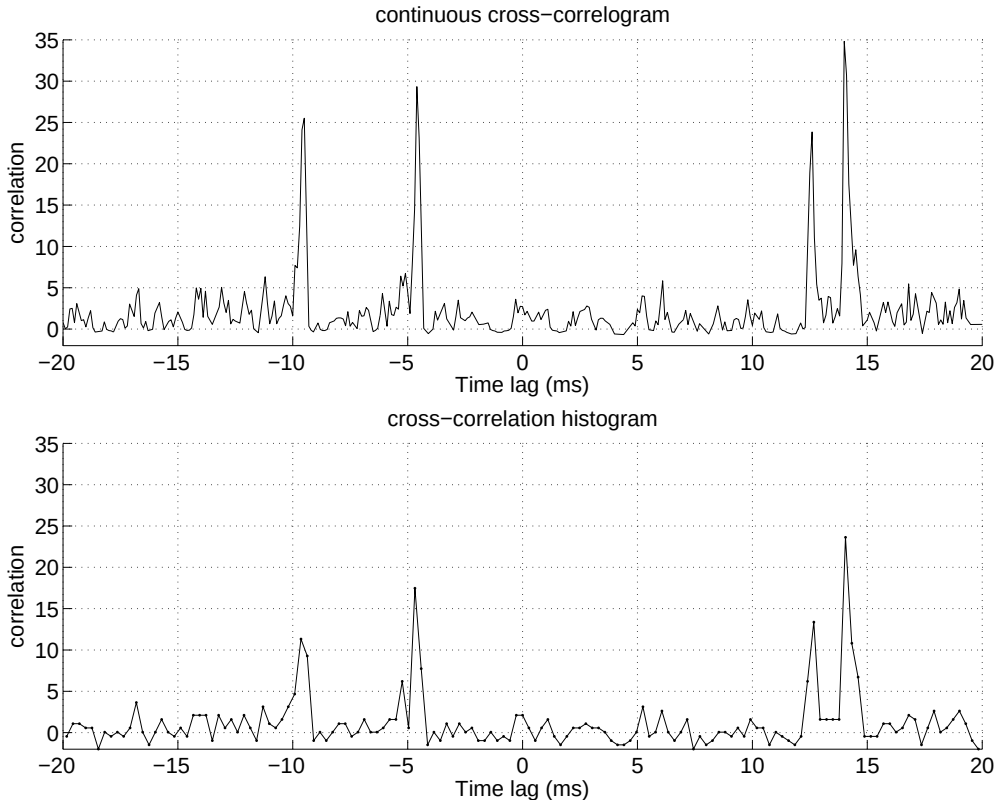



Fig. 6. Comparison between CCC and CCH on synthesized data.

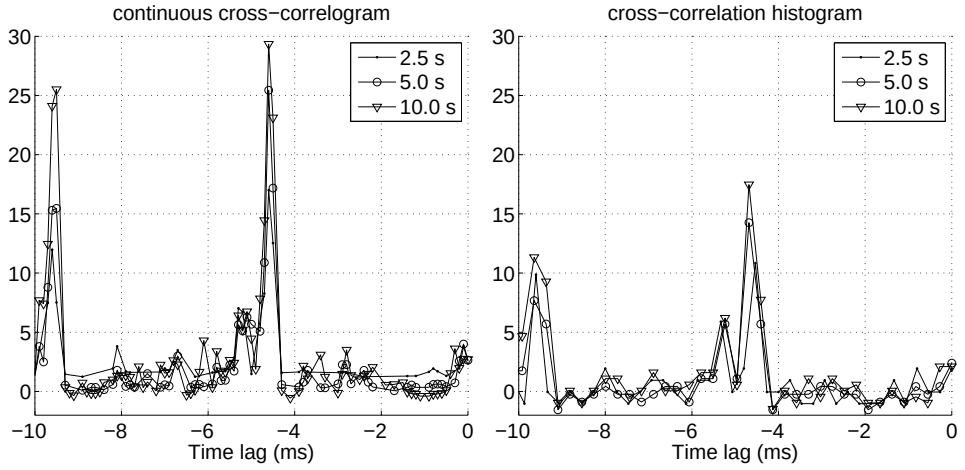



Fig. 7. Effect of length of spike trains. Comparison of continuous cross correlogram (left) and cross correlation histogram (right) with different length of spike trains (2.5, 5, 10 seconds). Note that when there are less time differences CCC is sparse and flat (e.g., 2.5 s), whereas CCH is uniformly dense and the values are fluctuating. Estimated optimal bin size is 0.267 ms.

| Length (sec) | Method | Bin size | | | |
|---|---|---|---|---|---|
| | | 10 ms | 1 ms | 0.1 ms | 0.02 ms |
| 1 | CCC | 0.19 / 0.12 | | | |
| | CCSH | 0.18 / 14.73 | 0.29 / 0.47 | 1.63 / **0.11** | 7.72 / 0.12 |
| | CCH | 0.12 / 5.78 | <u>0.11</u> / 2.00 | 0.12 / 7.89 | <u>0.11</u> / 12.61 |
| 10 | CCC | 0.47 / **0.05** | | | |
| | CCSH | 0.79 / 14.95 | 2.04 / 0.29 | 14.95 / **0.05** | 72.10 / **0.05** |
| | CCH | <u>0.16</u> / 3.98 | <u>0.16</u> / 0.29 | 0.17 / 0.09 | <u>0.16</u> / 0.11 |
| 100 | CCC | 3.43 / **0.02** | | | |
| | CCSH | 6.93 / 15.67 | 19.21 / 0.29 | 145.54 / 0.03 | 709.83 / **0.02** |
| | CCH | 0.58 / 3.84 | 0.58 / 0.29 | 0.58 / 0.06 | <u>0.57</u> / 0.06 |
| 1000 | CCC | 33.51 / **0.001** | | | |
| | CCSH | 67.71 / 14.18 | 190.04 / 0.28 | 1457.27 / 0.03 | 7104.35 / 0.01 |
| | CCH | 4.81 / 3.97 | <u>4.77</u> / 0.28 | 4.82 / 0.03 | 4.86 / 0.04 |

Table 2
Execution time / delay detection precision comparison of three methods. All units are milliseconds. The fastest runs are underlined and the most precise runs are highlighted in bold. Continuous cross correlogram (CCC) is not dependent on bin size. Cross correlation smoothed histogram (CCSH) and cross correlation histogram (CCH) depends on the number of time bins to use which is usually limited by the sampling rate. The task is same as Fig. 3 and 4 with 25 Hz firing rate, 0.2 ms jitter, 20% correlated spikes, $T^* = 20$ ms, and $\tau = 0.4$ ms. Precision and run time are average values from 100 Monte Carlo runs using MATLAB version 7.2 on an Intel machine with a 2.6GHz Pentium 4 processor and 1G RAM.

microelectrode array (MEA).

Two standard leaky-integrate-and-fire neurons are configured with 4 synapses, two from neuron A to neuron B, and two for the other direction as illustrated in figure 5. Individual synapses are static (no short/long-term plasticity), with equal weights and generate EPSP with a time constant of 1 ms. Each neuron is injected with positively biased Gaussian white noise current, so that they would fire with mean firing rate of 35 spikes/s. The simulation step size is 0.1 ms.

As shown in figure 6, both CCH and CCC identifies the delays imposed by the conduction delay, synaptic delay, and the delay for the generation of action potential by noisy fluctuation of membrane potential. However, the time lag identified by CCC is more accurate than that of CCH, since the temporal precision provided by CCH is limited by the bin size and the jitter noise on

delay, but for CCC, it is only limited by the jitter. In other words, if there is no jitter, or a sufficient amount of spike timings has the exact delay, then CCC is capable of quantifying the delay with infinite resolution (although, in practice, the resolution is always limited by the sampling period).

In figure 7, we illustrate the difference in performance of the methods according to the length of the spike trains. When the spike trains are only of length 2.5 seconds, the CCC has significantly lower time resolution where no spikes had that time difference, yet maintaining the high resolution in highly correlated peaks. In contrast, the CCH is uniformly sampled regardless of the amount of data. The non-uniform sampling gives significant advantage to CCC when only a short segment of data is available.

Spike trains recorded *in vitro* were used to further test the method. We recorded electrical activity from dissociated E-18 rat cortex cultured on a 60 channel microelectrode array from MultiChannel Systems with sampling rate 25 kHz (Potter and DeMarse, 2001). For a particular pair of electrodes, specific delays were observed as shown in Fig. 8. Those delays are rarely observed (3 to 5 times through 5 to 10 minutes of recording), however the dispersion of the delay is less than 2 ms which makes it significant in CCC. The delays persisted at least 2 days, and many more interaction delays were observable as the culture matured. As observable in the CCH analysis, it is almost impossible to detect the delays and their consistency.

Note that the delays are much longer ($> 40$ ms) than the expected conduction time which is estimated to be in the order of 2 ms for conduction speed of 100 $\mu$m/ms (Kawaguchi and Fukunishi, 1998). One possible mechanism would be a rarely activated chain of synaptic pathway from a common source neuron with different delays.

The time scale of our analysis is smaller compared to a recent study by le Feber et al. (2007) where a single delay between two channels is estimated with a single approximated Gaussian shape with bandwidth much larger than 10 ms. It would result in estimating the firing rate response from the network dynamics rather than propagation of a single action potential.

## 5 Discussion

We proposed an estimator of cross correlogram from an observation of a point process, and provide a efficient algorithm to compute it. The method utilizes the fact that there are more samples where the correlation is stronger (cf. Fig. 7). Thus, computing the continuous correlogram at the lags of samples provides an advantage for the delay estimation by this non-uniform sampling effect.

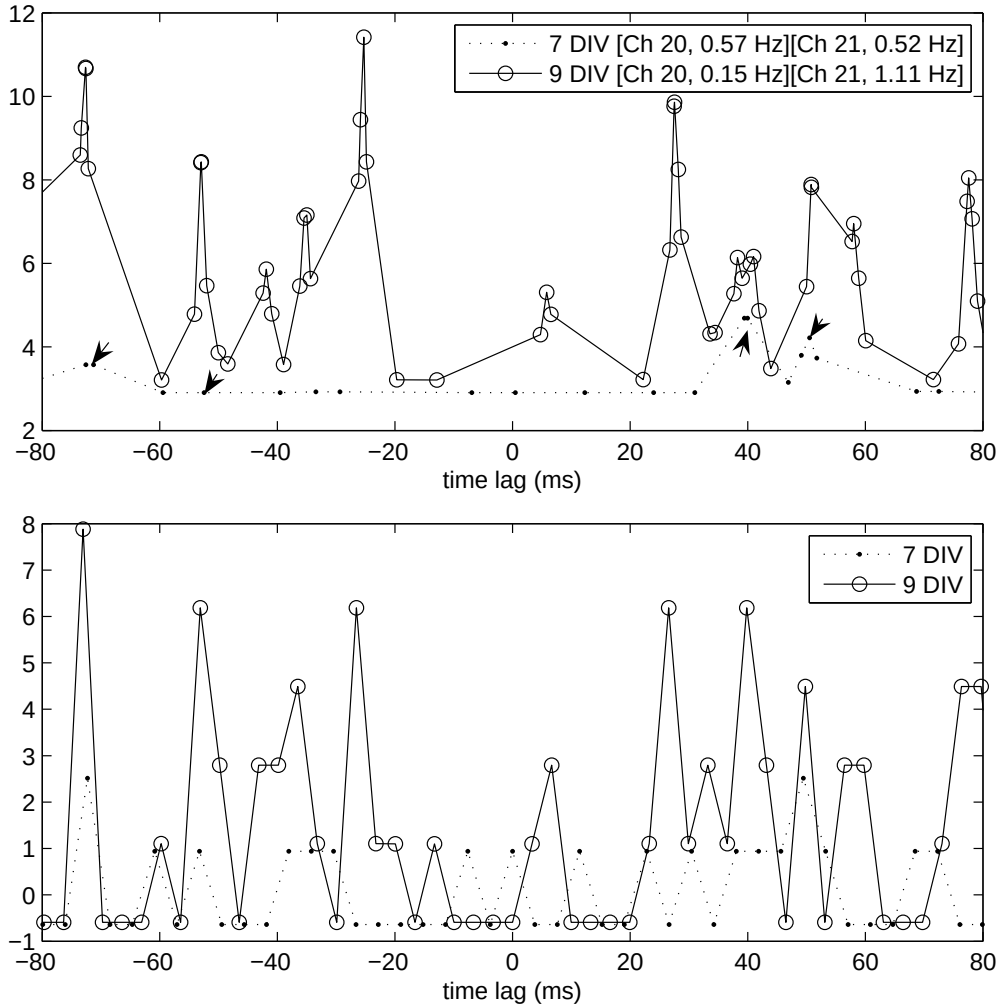



Fig. 8. CCC (top) and CCH (bottom) of 7 DIV (days *in vitro*) and 9 DIV cortical culture recordings. Spike trains from two adjacent electrodes are analyzed. On 7 DIV, CCC shows two significant peaks (larger than 4 standard deviations, see Eq. (9)) and they are also observable on 9 DIV, and some non-significant spike time differences corresponds to peaks on 9 DIV (marked with arrows). In contrast, this structure is difficult to note from CCH. The optimal bin size is 3.8 ms for 7 DIV and 3.3 ms for 9 DIV data. The total recording time is 350 seconds for 7 DIV and 625 seconds for 9 DIV. The average firing rates were 0.57 Hz and 0.53 Hz on 7 DIV and 0.15 Hz and 1.11 Hz on 9 DIV.

Unfortunately, this non-uniform sampling is disadvantageous for inhibitory relations, therefore only positively related delays can be accurately estimated. To achieve computational efficiency, as presented here, the algorithm is limited to the use of Laplacian distribution as the kernel. However, it has been shown that the bandwidth (kernel size) is more important than the shape of the kernel for the performance of intensity estimation (Nawrot et al., 1999).

The only free parameter is the kernel size which determines the amount of smoothing. Unlike the conventionally used histogram method, the proposed

method is much more robust to the choice kernel size, however, the optimal kernel size depends on the noise level of the delay. In a biological neuronal network, the noise level may depend on which path the signal was transmitted. Therefore each peak of the correlogram may have different amount of noise. We suggest using the optimal bin size for histogram as a guideline for the kernel size selection. Although it is possible to empirically determine the optimal kernel size given a model (as in Fig. 4(a)), the general problem of optimal kernel size for delay estimation remains as an open problem.

The proposed algorithm is not limited to cross correlations, it can be used to smooth any type of point process histograms, and also to find the similarity between two spike trains over continuous time lags. However, due to accumulation of numerical error, the algorithm has to be non-causal (see (21)). This prevents the algorithm to be used as an online filter to detect certain spike train patterns, while offline analysis can still be done.

## Appendix

To assess the significance of the correlation, it is necessary to know the probability distribution of the estimator given the null hypothesis (independent Poisson spike trains). However, instead of calculating the complicated closed form of the distribution for $\hat{Q^*}_{ij}(\Delta t)$, we derive of mean and variance of the estimator $\hat{Q}^*_{ij}(\Delta t)$, and assume Gaussianity. For the time binning with sufficiently small bin size case, Palm and coworkers have derived statistics for the histogram of a Poisson process (Palm et al., 1988).

Let $\Omega(\lambda, T)$ be the class of all possible homogeneous Poisson spike trains of rate $\lambda$ and length $T$. A Poisson process can be decomposed into a pair of independent random variables – a discrete random variable for the number of spikes, and a continuous random variable for the distribution of the spikes (Snyder and Miller, 1991). The probability density of having a realization $\Omega_i = \{t^i_1, t^i_2, \ldots, t^i_{N_i}\}$ of $\Omega(\lambda_I, T)$ is,

$$f_\Omega(\Omega_i|\lambda_I, T) = \Pr[N(T) = N_i] f_{\Omega|N}(\mathbf{t^i_1} = t^i_1, \mathbf{t^i_2} = t^i_2, \ldots, \mathbf{t^i_{N_i}} = t^i_{N_i}|N_i) \quad (22)$$

$$= \frac{(\lambda_I T)^{N_i}}{N_i!} e^{-\lambda_I T} \prod_{m=1}^{N_i} f_t[\mathbf{t} = t^i_m] \quad (23)$$

$$= \frac{(\lambda_I T)^{N_i}}{N_i!} e^{-\lambda_I T} \prod_{m=1}^{N_i} \frac{1}{T} = \frac{\lambda_I^{N_i}}{N_i!} e^{-\lambda_I T}, \quad (24)$$

where $f_t$ is the pdf for a uniform distribution. The expected value of the

estimator for all possible pairs of independent spike trains is,

$$E_{ij}\left[\hat{Q}^*_{ij}(\Delta t)\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_\Omega(\Omega_i|\lambda_I,T) f_\Omega(\Omega_j|\lambda_J,T) Q^*_{ij}(\Delta t) d\Omega_i d\Omega_j \tag{25a}$$

$$= \sum_{N_i=0}^{\infty}\sum_{N_j=0}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \frac{\lambda_I^{N_i}}{N_i!}e^{-\lambda_I T}\frac{\lambda_J^{N_j}}{N_j!}e^{-\lambda_J T}$$

$$\frac{1}{2\tau T}\sum_{m=1}^{N_i}\sum_{n=1}^{N_j} e^{-\frac{|t^i_m - t^j_n - \Delta t|}{\tau}} dt^i_1 dt^i_2\cdots dt^i_{N_i} dt^j_1 dt^j_2\cdots dt^j_{N_j} \tag{25b}$$

$$= \frac{1}{2\tau T}e^{-(\lambda_I+\lambda_J)T}\sum_{N_i=0}^{\infty}\sum_{N_j=0}^{\infty}\frac{(\lambda_I T)^{N_i}}{N_i!}\frac{(\lambda_J T)^{N_j}}{N_j!}\frac{N_i N_j}{T^2}$$

$$\int_0^T\int_0^T e^{-\frac{|t^i_m - t^j_n - \Delta t|}{\tau}} dt^i_m dt^j_n \tag{25c}$$

Let us evaluate the integral first, from the symmetry of $t^i_m$ and $t^j_n$ we can assume $\Delta t \geq 0$ without loss of generality.

$$\int_0^T\int_0^T e^{-\frac{|t^i_m - t^j_n - \Delta t|}{\tau}} dt^i_m dt^j_n \tag{26a}$$

$$= \int_0^{\Delta t}\int_0^T e^{\frac{t^i_m - t^j_n - \Delta t}{\tau}} dt^i_m dt^j_n$$

$$+ \int_{\Delta t}^T\int_0^{t^i_m - \Delta t} e^{-\frac{t^i_m - t^j_n - \Delta t}{\tau}} dt^i_m dt^j_n$$

$$+ \int_{\Delta t}^T\int_{t^i_m - \Delta t}^T e^{\frac{t^i_m - t^j_n - \Delta t}{\tau}} dt^i_m dt^j_n \tag{26b}$$

$$= -\tau\int_0^{\Delta t}\left(e^{\frac{t^i_m - T - \Delta t}{\tau}} - e^{\frac{t^i_m - \Delta t}{\tau}}\right) dt^i_m$$

$$+ \tau\int_{\Delta t}^T\left(1 - e^{-\frac{t^i_m - \Delta t}{\tau}}\right) dt^i_m - \tau\int_{\Delta t}^T\left(e^{\frac{t^i_m - T - \Delta t}{\tau}} - 1\right) dt^i_m \tag{26c}$$

$$= -\tau^2\left(e^{\frac{\Delta t - T - \Delta t}{\tau}} - e^{\frac{0 - T - \Delta t}{\tau}} - e^{\frac{\Delta t - \Delta t}{\tau}} + e^{\frac{0 - \Delta t}{\tau}}\right)$$

$$+ \tau(T - \Delta t) - \tau^2\left(-e^{-\frac{T - \Delta t}{\tau}} + e^{-\frac{\Delta t - \Delta t}{\tau}}\right)$$

$$- \tau^2\left(e^{\frac{T - T - \Delta t}{\tau}} - e^{\frac{\Delta t - T - \Delta t}{\tau}}\right) + \tau(T - \Delta t) \tag{26d}$$

$$= 2\tau(T - \Delta t) + \tau^2(e^{\frac{-T + \Delta t}{\tau}} + e^{\frac{-T - \Delta t}{\tau}} - 2e^{-\frac{\Delta t}{\tau}}) \tag{26e}$$

$$= 2\tau(T - \Delta t) + O(\tau^2). \tag{26f}$$

Approximating $\lambda_I = \frac{N_i}{T}$, and substituting the integral to (25c) gives,

$$E_{ij}\left[\hat{Q}^*_{ij}(\Delta t)\right] \simeq \lambda_I\lambda_J\frac{2\tau(T - \Delta t) + O(\tau^2)}{2\tau T}, \tag{27}$$

where $O(\tau^2)$ are the terms with order of $\tau^2$ or higher. Assuming $\tau \ll 1$ and $\Delta t \ll T$, (27) can be approximated by $\lambda_I\lambda_J$ which is the desired value.

Now let us evaluate the second-moment of the estimator.

$$E_{ij}\left[\hat{Q}^*_{ij}(\Delta t)^2\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_\Omega(\Omega_i|\lambda_I,T) f_\Omega(\Omega_j|\lambda_J,T)\; Q^*_{ij}(\Delta t)^2 d\Omega_i d\Omega_j \tag{28a}$$

$$= \sum_{N_i=0}^{\infty}\sum_{N_j=0}^{\infty} \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \frac{\lambda_I^{N_i}}{N_i!}e^{-\lambda_I T}\frac{\lambda_J^{N_j}}{N_j!}e^{-\lambda_J T}$$
$$\frac{1}{4\tau^2T^2}\sum_{p=1}^{N_i}\sum_{q=1}^{N_j}\sum_{r=1}^{N_i}\sum_{s=1}^{N_j} e^{-\frac{|t^i_p - t^j_q - \Delta t|}{\tau}} e^{-\frac{|t^i_r - t^j_s - \Delta t|}{\tau}}$$
$$dt^i_1 dt^i_2\cdots dt^i_{N_i} dt^j_1 dt^j_2\cdots dt^j_{N_j} \tag{28b}$$

$$= \sum_{N_i=0}^{\infty}\sum_{N_j=0}^{\infty} \frac{\lambda_I^{N_i}}{N_i!}e^{-\lambda_I T}\frac{\lambda_J^{N_j}}{N_j!}e^{-\lambda_J T}\frac{1}{4\tau^2T^2}T^{N_i}T^{N_j}\frac{1}{T^4}$$
$$\int_0^T\int_0^T\int_0^T\int_0^T \sum_{p=1}^{N_i}\sum_{q=1}^{N_j}\sum_{r=1}^{N_i}\sum_{s=1}^{N_j} e^{-\frac{|t^i_p - t^j_q - \Delta t|}{\tau}} e^{-\frac{|t^i_r - t^j_s - \Delta t|}{\tau}}$$
$$dt^i_p dt^j_q dt^i_r dt^j_s \tag{28c}$$

Let us consider the integral part first.

$$\frac{1}{T^4}\int_0^T\int_0^T\int_0^T\int_0^T \sum_{p=1}^{N_i}\sum_{q=1}^{N_j}\sum_{r=1}^{N_i}\sum_{s=1}^{N_j} e^{-\frac{|t^i_p - t^j_q - \Delta t|}{\tau}} e^{-\frac{|t^i_r - t^j_s - \Delta t|}{\tau}} dt^i_p dt^j_q dt^i_r dt^j_s \tag{29a}$$

$$= \frac{1}{T^4}\sum_{p=1}^{N_i}\sum_{q=1}^{N_j}\sum_{r\neq p}\sum_{s\neq q}\int_0^T\int_0^T e^{-\frac{|t^i_p - t^j_q - \Delta t|}{\tau}} dt^i_p dt^j_q \int_0^T\int_0^T e^{-\frac{|t^i_r - t^j_s - \Delta t|}{\tau}} dt^i_r dt^j_s$$
$$+ \frac{1}{T^2}\sum_{p=1}^{N_i}\sum_{q=1}^{N_j}\int_0^T\int_0^T e^{-\frac{2|t^i_p - t^j_q - \Delta t|}{\tau}} dt^i_p dt^j_q$$
$$+ \frac{1}{T^3}\sum_{p=1}^{N_i}\sum_{q=1}^{N_j}\sum_{r\neq p}\int_0^T\int_0^T\int_0^T e^{-\frac{|t^i_p - t^j_q - \Delta t|}{\tau}} e^{-\frac{|t^i_r - t^j_s - \Delta t|}{\tau}} dt^i_p dt^j_q dt^i_r$$
$$+ \frac{1}{T^3}\sum_{p=1}^{N_i}\sum_{q=1}^{N_j}\sum_{s\neq q}\int_0^T\int_0^T\int_0^T e^{-\frac{|t^i_p - t^j_q - \Delta t|}{\tau}} e^{-\frac{|t^i_r - t^j_s - \Delta t|}{\tau}} dt^i_p dt^j_q dt^j_s \tag{29b}$$

$$= \frac{N_iN_j(N_i-1)(N_j-1)}{T^4}(2\tau(T-\Delta t)+O(\tau^2))^2$$
$$+ \frac{N_iN_j}{T^2}(\tau(T-\Delta t)+O(\tau^2))$$
$$+ \frac{N_iN_j(N_i-1)}{T^3}(\tau^2(2T(2+e^{-\frac{T}{\tau}}))+O(\tau^3)$$
$$+ \frac{N_iN_j(N_j-1)}{T^3}(\tau^2(2T(2+e^{-\frac{T}{\tau}}))+O(\tau^3) \tag{29c}$$

By assuming $\tau \ll 1 \ll T$, we can approximate $e^{-\frac{T}{\tau}} \simeq 0$, $O(\tau^3) \simeq 0$. And we

further approximate $\lambda_I = \frac{N_i}{T}$, and $N_i - 1 \simeq N_i$. These approximations lead to,

$$\begin{aligned} E_{ij}\left[\hat{Q}^*_{ij}(\Delta t)^2\right] &\simeq \frac{1}{4\tau^2 T^2}\{(\lambda_I\lambda_J)^2(2\tau(T-\Delta t))^2 \\ &\quad + (\lambda_I\lambda_J)(\tau(T-\Delta t)) + (\lambda_I\lambda_J)(\lambda_I+\lambda_J)(4\tau^2 T)\} \\ &= (\lambda_I\lambda_J)^2\left(\frac{T-\Delta t}{T}\right)^2 + \frac{\lambda_I\lambda_J}{4\tau T}\frac{T-\Delta t}{T} \\ &\quad + (\lambda_I\lambda_J)(\lambda_I+\lambda_J)\frac{1}{T}. \end{aligned} \tag{30}$$

Finally, the variance of the estimator is given by

$$E_{ij}\left[(\hat{Q}^*_{ij}(\Delta t) - \lambda_I\lambda_J)^2\right] \simeq \frac{\lambda_I\lambda_J(T-\Delta t)}{4\tau T^2}. \tag{31}$$

## Acknowledgments

I. P. thanks Karl Dockendorf for insightful discussion, and Alexander Singh-Alverado and Alex Cadotte for carefully reading the manuscript. This work was partially supported by NSF grant ECS-0422718 and CISE-0541241. A. R. C. Paiva was supported by Fundação para a Ciência e a Tecnologia under grant SFRH/BD/18217/2004.

## References

Aertsen, A. M., Gerstein, G. L., Habib, M. K., Palm, G., 1989. Dynamics of neuronal firing correlation: modulation of "effective connectivity". Journal of Neurophysiology 61 (5), 900–917.

Brivanlou, I. H., Warland, D. K., Meister, M., 1998. Mechanisms of concerted firing among retinal ganglion cells. Neuron 20, 527–539.

Brody, C. D., 1999. Correlations without synchrony. Neural Comput 11, 1537–1551.

Brown, E. N., Kass, R. E., Mitra, P. P., may 2004. Multiple neural spike train data analysis: state-of-the-art and future challenges. Nature neuroscience 7 (5), 456–61.

Dayan, P., Abbott, L. F., 2001. Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT Press, Cambridge, MA, USA.

Diggle, P., Marron, J. S., Sep. 1988. Equivalence of smoothing parameter selectors in density and intensity estimation. Journal of the American Statistical Association 83 (403), 793–800.

Gerstner, W., 2001. Coding properties of spiking neurons: reverse and cross-correlations. Neural Networks 14, 599–610.

Hahnloser, R. H. R., May 2007. Cross-intensity functions and the estimate of spike-time jitter. Biol. Cybern. 96 (5), 497–506.

Kass, R. E., Ventura, V., Cai, C., 2003. Statistical smoothing of neuronal data. Network: Comput. Neural Syst. 14, 5–15.

Kawaguchi, H., Fukunishi, K., 1998. Dendrite classification in rat hippocampal neurons according to signal propagation properties. Experiments in Brain Research 122, 378 – 392.

Knox, C. K., 1974. Cross-correlation functions for a neuronal model. Biophysical Journal 14 (8), 567–582.

König, P., Engel, A., Roelfsema, P., Singer, W., May 1995. How precise is neuronal synchronization? Neural Comp. 7 (3), 469–485.

le Feber, J., Rutten, W. L. C., Stegenga, J., Wolters, P. S., Ramakers, G. J. A., van Pelt, J., 2007. Conditional firing probabilities in cultured neuronal networks: a stable underlying structure in widely varying spontaneous activity patterns. J. Neural Eng. 4, 54–67.

Maass, W., 1997. Networks of spiking neurons: the third generation of neural network models. Neural Networks 10, 1659–1671.

Mainen, Z. F., Sejnowski, T. J., 1995. Reliability of spike timing in neocortical neurons. Science 268 (5216), 1503–1506.

Nawrot, M., Aertsen, A., Rotter, S., 1999. Single-trial estimation of neuronal firing rates: From single-neuron spike trains to population activity. Journal of Neuroscience Methods 94, 81–92.

Nikolić, D., 2007. Non-parametric detection of temporal order across pairwise measurements of time delays. J. Comput. Neurosci. 22, 5–19.

Nykamp, D. Q., 2005. Revealing pairwise coupling in linear-nonlinear networks. SIAM Journal on Applied Mathematics 65 (6), 2005–2032.

Palm, G., Aertsen, A. M. H. J., Gerstein, G. L., 1988. On the significance of correlations among neuronal spike trains. Biological Cybernetics 59, 1–11.

Parzen, E., Sep. 1962. On the estimation of a probability density function and the mode. The Annals of Mathematical Statistics 33 (2), 1065–1076.

Perkel, D. H., Gerstein, G. L., Moore, G. P., Jul. 1967. Neuronal spike trains and stochastic point processes: II. simultaneous spike trains. Biophys J 7 (4), 419–440.

Potter, S. M., DeMarse, T. B., 2001. A new approach to neural cell culture for long-term studies. Journal of Neuroscience Methods 110, 17–24.

Reinagel, P., Reid, R. C., 2000. Temporal coding of visual information in the thalamus. Journal of Neuroscience 20 (14), 5392–5400.

Sabatini, B. L., Regehr, W. G., 1999. Timing of synaptic transmission. Annual Review of Physiology 61, 521–542.

Shimazaki, H., Shinomoto, S., 2006. A recipe for optimizing a time-histogram. In: Neural Information Processing Systems.

Snyder, D. L., Miller, M. I., 1991. Random Point Processes in Time and Space. Springer-Verlag.

VanRullen, R., Guyonneau, R., Thorpe, S. J., Jan 2005. Spike times make sense. Trends in Neurosciences 28 (1), 1–4.