

High-speed low-cost correlator for single molecule fluorescence correlation spectroscopy

Hsu-Yang Lee^a, Hsin-Yu Lin^b, Jonathon D. White^{b*}, Wunshain Fann^a,

^aInstitute of Atomic and Molecular Sciences, No. 1, Roosevelt Rd., Sec. 4, Taipei, Taiwan, 10617;

^bYuan Ze University, 135 Yuan-Tung Road, Chung-Li, Taiwan, 32003

ABSTRACT

Fluorescence correlation spectroscopy (FCS) has been extensively applied to study the kinetics and photophysics of molecules as well as interactions between molecules by extracting information from the fluctuation of signals. In particular, single molecule applications of FCS promise the greatest amounts of information. Ideally, one would like to carry out FCS in real-time; however, due to the time-consuming nature of the correlation process, performing the correlation in real-time is totally nontrivial. Generally an expensive hardware correlator or a TCSPC board is required for this purpose. Recently highly-efficient algorithms based on multi-tau method have been proposed to build up a software correlator. In this work, we set forth an innovative algorithm capable of realizing the real-time correlation, without turning to the multi-tau method. This algorithm takes advantage of the low count rate generally existing in the FCS experiments, directly using the time interval between each photon its adjacent photon to efficiently update the correlation function. Based on this efficiency, it is possible to build a low-cost software correlator with just an ordinary counter board. We practically demonstrate the feasibility by setting up this correlator to measure the diffusion motion of rhodamine 6G in water using FCS. The algorithm was validated by duplicating the signal from the photon detector and sending it to both the ordinary counter board with our software correlator and a commercial correlator simultaneously. The perfect coincidence of the correlation curves from these two correlators and the real-time display of the correlation function indicate the validity and practicability of our approach.

Keyword: Fluorescence Correlation Spectroscopy, Confocal Microscopy, Single Molecule Spectroscopy, Single Molecule Detection

1. INTRODUCTION

Fluorescence correlation spectroscopy (FCS), first performed by Magde et al.[1] in 1972, is an experimental technique capable of extracting information from intensity fluctuation of fluorescence signals using statistical methods. Owing to the improvement in detector sensitivity and the introduction of the confocal setup, fluorescence as weak as that from a single dye molecule turns out to be detectable, leading to a dramatic enhancement of the sensitivity of FCS[2]. Numerous applications of FCS to study the dynamics of biomolecules under single molecule condition has been reported and reviewed[3].

In a typical FCS experiment, the photo-detector generates an electric pulse whenever a photon is detected. The resulting pulse train is usually processed in one of the following three ways. The first and the simplest one, is to count the photon number in every fixed time interval (binning time), resulting in the variation of fluorescence intensity as a function of time, i.e., the fluorescence time trace, which can be subsequently processed to generate the correlation curve. However, when the binning time is short, as in the case of most FCS experiments, the high data flow rate make real-time processing impossible. An alternative, capable of processing data in real-time, is the use of a hardware correlator. This equipment has already been commercially available for many years and most commonly employed in current FCS experiments; however, besides the relative high price, it can output little information other than the correlation function: the information of each photon is essentially lost during the processing. In addition, there is the assumption that the process is stationary[4, 5]. A third method involves the introduction of the time-correlated single photon counting (TCSPC) boards, which not only

*Further author information: (Send correspondence to J.D.White)
J.D.White: E-mail: whitejd@xiaotu.com

correlate data in real-time but also record the arrival time of each photon in high temporal resolution[5, 6]. While powerful, the TCSPC boards are generally very expensive. Additionally, Recently several highly-efficient algorithm for single molecule FCS have been recently proposed[7-9] and used to build software-based correlator.

In this work, we present a highly-efficient correlation algorithm, which enables software-based real-time display of correlation curve, time trace, distribution of delay time, when the photon count rate is within the range of common for single photon counting, in particular, under the single molecule condition. Rather than binning data, the arrival time of each photon is also completely recorded for any post-processing. Since the correlation is carried out by software, only a low-cost counting board commonly used in labs is required to receive the electronic pulse generated by the detector. The performance of this correlator is validated through practically measuring the correlation curve of Rhodamine 6G (R6G) water solution.

The performance of this correlator is contributed by a number of factors. In addition to the high-efficiency of the algorithm, the use of a multi-core computer in combination with LabVIEW language, which possesses parallel computing ability, allows different tasks to be assigned to individual cores. Finally, the time-critical components are coded in C language and compiled into DLL files.

2. HARDWARE

Fig. 1 shows the key aspects of the hardware structure used in our system. The photon detector is an avalanche photodiode single photon counting module, which generates a TTL pulse whenever a photon is detected. The stream of TTL pulses was directed into the gate channel of a National Instrument PCI-6602 counter board, of which either the 80MHz or the 20MHz clock was used as the source. Since the counter counts the number of rising edges in the source channel between every two successive rising edges in the gate channel, the output data essentially represents the time interval between two successive photons (interphoton time) with the temporal resolution uniquely determined by the clock. The data flow rate equals to the photon count rate. By contrast, in a standard implementation of obtaining the time trace, TTL pulses are used as the source, and a clock that determines both data flow rate as well as temporal resolution as the gate. Considering the low photon count rate $\sim 10^4$ Hz in a typical single molecule experiment and the requirement of temporal resolution $\sim 0.1 \mu s$ (10^7 Hz) in a typical FCS experiment, our connection always results in lower

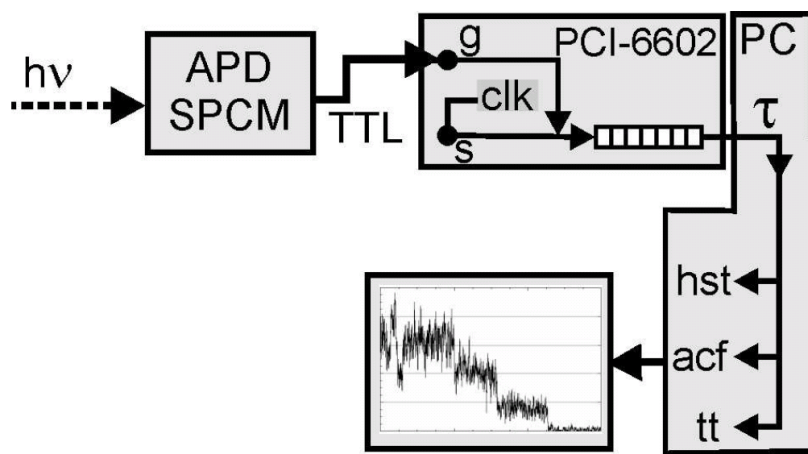


Fig. 1 Flow chart of Real Time data acquisition, processing and display system controlled by LabVIEW. For each photon, a single TTL pulse is generated by an avalanche photo diode single photon counting detector (SPCM APD) detector. The TTL pulse is used to gate the internal clock of the PCI-6602 (National Instruments) board. Labview is used to control the system and transfer the number of clock pulses between adjacent photons (t) to the computer memory (CPU core-1). As each photon arrives its arrival time relative to the previous photon is added to the inter photon arrival time histogram (hst) (CPU core-2), the autocorrelation function (acf) is updated (CPU core-3) and the arrival time data is binned to produce the classical time trace (tt) (CPU core-4). While all the above can be done on a single computer, in the implementation used in this study, an Ethernet cable was used to transfer the information to second computer which is used to display information and control experimental parameters.

data flow rate with invariant temporal resolution. At this count rate, in the standard implementations, most elements in the time trace are zeros. The counter board was installed in a computer equipped with a 4-core CPU, which allows data acquisition, storing, correlation processing and display to proceed in parallel.

3. SOFTWARE, ALGORITHM

The correlation algorithm were coded in ANSI-C language in view of the efficiency and portability, and compiled as a dynamic link library incorporated in a program created with LabVIEW (National Instruments). LabVIEW was chosen due to its current popularity in the lab and the fact that it possesses the parallel processing ability required to maximize the power of the multi-core CPU to perform multiple functions simultaneously. In addition, using the real-time extension of LabVIEW, the compiled program can then be deployed and run on a target computer with a real-time operating system while the data is transferred by an Ethernet cable back to the host computer where experimental parameters can be set and the data is displayed. In this work we have used the program both with and without real-time employment. The raw data (i.e. the interphoton arrival time) was saved into hard drive while simultaneously being processed to update the correlation curve, time trace and the histogram of the interphoton time in real-time. In real time mode, while the refresh rate of the display was limited by the bandwidth of the Ethernet communication, no data is lost during the acquisition and processing.

The most important component which makes software-based correlation possible is the highly-efficient algorithm. The basic autocorrelation formula for autocorrelation at time lag k for a discrete series of n samples, $X=\{X_0, X_1, \dots, X_i \dots X_n\}$ is:

$$acf(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} (X_t - \mu)(X_{t+k} - \mu), \quad (1)$$

where σ is the standard deviation, μ the mean, k the current time lag. Ignoring normalization and assuming that the process is stationary (i.e. no permanent photobleaching), this summation can be simplified to:

$$acf(k) = \sum_{t=1}^{n-k} X_t X_{t+k} \quad (2)$$

As a consequence, the processing time increases quadratically with the number of samples. When the number of samples becomes enormous, as in the case of high temporal resolution time trace, the autocorrelation turns time-consuming even for post-processing, let alone displaying in real-time. The bulky size of the time trace in low count rate ($\sim 10^4$ Hz) situation largely comes from the high percentage of zeros among the elements. As stated formerly, when the interphoton time is introduced to record the same information, the data size is dramatically reduced. Therefore, the key factor in our algorithm is to update the correlation function directly using the interphoton time.

This algorithm can be established by first converting the stream of interphoton time back into its corresponding time trace. In this kind of time trace, only dispersed unity and plenty of zeros exist. If we try to examine how each “unity” contributes to the autocorrelation result, we can find that the autocorrelation function increases one whenever the delay time equals to the time interval between the “unity” and any other “unity”, which is exactly the accumulation of the interphoton time. In practice, two arrays are created to accomplish the operation. The first one, “delay array”, is used to record the time interval between the last photon and each of its previous photon, and is updated by adding the last interphoton time into each element when a new datum is available. The second one is just the autocorrelation array (“acf array”), which is updated according to the look up table “delay array”. Expressed in ANSI-C code, the calculation is done in three simple steps at each inter photon arrival time (tau):

```
m++;
for (k=m; k>0; k--) delay[k]=delay[k-1] + tau;
while (delay[m]>mtau) m--;
for (k=m; k>0; k--) if (delay[k]>0) acf[delay[k]]++;
```

where τ is the time interval between the arrival of the present and previous photon, In this algorithm, m is the last element of the array to be updated. $m < n$ where n is the number of expected photons in the maximum time interval for which

The efficiency of this algorithm can be seen through a simple comparison with a brute force approach. The following assumptions are employed: addition takes 4 clock cycles, multiplication 4 clock cycles, and the shift and comparison operations both take 1 clock cycle of the CPU, and the ratio between the photon count rate and the inverse of binning time is d . Accordingly, to update the autocorrelation function by the time trace for each new coming time binning, it takes $\sim m\tau + (m\tau * d)$. By direct update from interphoton time data stream for each new coming element, it takes $(m\tau * d) * 4 + (m\tau * d) * 1$. Considering in a certain time interval, the number of time binning is $1/d$ fold more than the elements of interphoton time, the ratio of required time is $5d * d * m\tau / (m\tau + (m\tau * d))$. Namely, the time trace method is more time-consuming than the interphoton time one by $O(d^2)$.

A similar but more tedious algorithm can be applied to cross-correlation. In this algorithm, the correlation function is updated by the time interval between the new coming photon in one array and each of its previous photon in the other array. To synchronize the two arrays, an additional variable which records the time difference between their last components is created. As more comparisons in the update process are unavoidable the efficiency is worse than that for the autocorrelation case but still a considerable improvement over the brute force binning method.

4. EXPERIMENTAL VALIDATION

The performance of the correlator was practically tested by measuring the autocorrelation function of organic dye molecules diffusing in water solution. Two different concentrations of R6G (Molecular Probe) water solution, 1nM and 10nM, were used to fill the cavity on a caved glass slide. The slide was then covered with a 22x22mm cover slip and sealed with transparent silicon. Excitation light source was provided by a frequency doubled diode-pumped solid state laser (JL-LD532-GTE; Jetlaser) operated at 532 nm. Residual light at other frequencies was eliminated using a narrow band excitation filter. After introduction into an modified upright microscope (E600; Nikon), the laser beam was reflected by a dichroic beamsplitter (540 DCLP; Chroma), and then focused by an objective (Plan Fluor, N.A. 1.3 oil; Nikon) onto the sample. Fluorescence was collected in epifluorescence mode through the same objective. The light after passing through an emission filter (HQ575/50; Chroma), was focused into a 50 μ m pinhole before entering a multimode fiber leading to a avalanche photodiode single photon counting module.

To validate the algorithm as well as the programming, the TTL signals from the APD were split by a pulse divider into two identical parts following two different paths. Then they were recovered to the original voltage level by fast electronics and delivered into our system and a commercial hardware correlator, respectively, for real-time comparison. Since the temporal resolution of the hardware correlator is 125ns, we chose the 20MHz clock on the counter board, rather than the 80MHz one.

During these test experiments, the system was able to keep updating the auto-correlation curve, time trace, the histogram of the interphoton time, in real-time, while saving the raw data into hard drive without any data loss for longer than 30 minutes with no sign of system failure. Fig. 3 shows a typical experiment result, in which the auto-correlation curves from our system and the commercial hardware correlator well coincide with each other, except that the former is significantly noisier in longer delay time. The is due to the fact that the commercial correlator applies a multi-tau algorithm resulting in decreased time resolution at longer lag times while the proposed algorithm maintains the 50ns time resolution over the complete range. The curve will become as smooth if logarithmic binning, similar to that used in the commercial system is applied.

The auto-correlation curve can be fitted by the curve

$$G(\tau) = a_0 \frac{1}{\left(1 + \frac{\tau}{a_1}\right)} \frac{1}{\sqrt{1 + \frac{\tau}{a_2}}}, \quad (3)$$

where a_0 , a_1 , and a_2 are adjustable parameters. Under the assumption of a Gaussian beam profile, the diffusion time τ_{diff} , the focal volume, average number of molecules in the focal region (N), and hence concentration to be obtained through

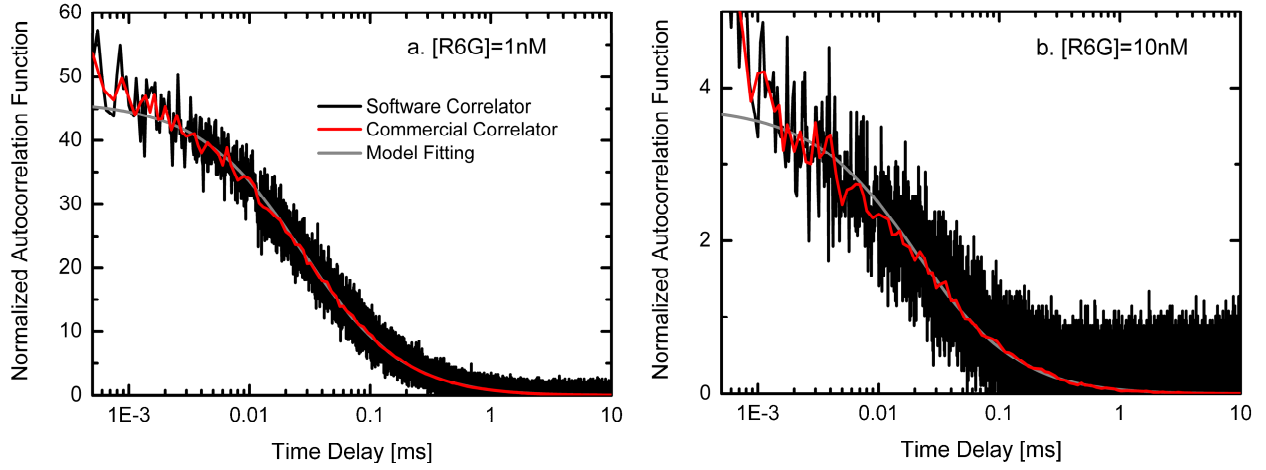


Fig. 3 Comparison of the calculated auto-correlation function of the fluorescence time trace of (left) low ($\sim 1\text{nM}$) and (right) high ($\sim 10\text{nM}$) concentration of R6G molecules diffusing in water compared with that generated by a commercial correlator. The photon stream leading to the two correlators is identical, and is acquired during the same period. The gray curve represents a fitting to the experimental data (see text for details).

comparison with the equation[2],

$$G(\tau) = \frac{1}{N} \frac{1}{\left(1 + \frac{\tau}{\tau_{diff}}\right)} \frac{1}{\sqrt{1 + \frac{(r/l)^2 \tau}{\tau_{diff}}}}. \quad (4)$$

The results listed in Table 1 are both self-consistent and consistent with the nominal concentrations used in this experiment. The diffusion times also fall in the range of values commonly measured for small molecules.

The cross-correlation algorithm was validated by connecting the split signal into the two channels of the cross-correlator, and running the auto-correlator for each channel as well as the cross-correlator simultaneously. If the cross-correlation algorithm works correctly, a correlation curve resembling the autocorrelation ones with a time shift due to

Table 1: Fitting Parameters and physical information about R6G molecules in DI water based on the autocorrelation function (see text for details).

Item	Nominal R6G Concentration	
	$\sim 1\text{nM}$	$\sim 10\text{nM}$
a_0	46	3.7
a_1	27,178	20,305
a_2	7244,55	1,120,000
Diffusion Time [μs]	27	20
Molecules in Focal Region (N)	0.02	0.26
Focal Volume [fl]	0.16	0.14
Concentration [nM]	0.24	3.1

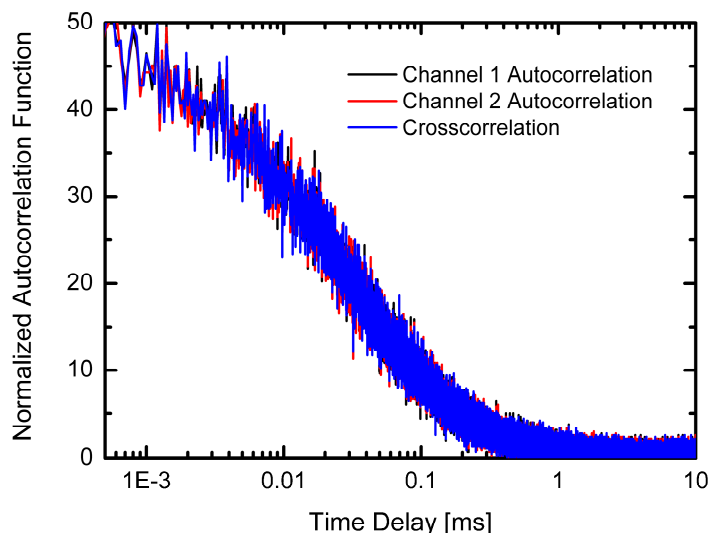


Fig. 4 Comparison of the auto-correlation functions of the fluorescence time trace of 1nM R6G obtained from channel 1 and channel 2 as well as the cross-correlation function of these two channels. All the three curves well coincide with one another, verifying the validity of the cross-correlation algorithm.

the different circuit length is expected. Fig. 4 shows the results of this test, in which three curves again well coincide with one another, just as expected. It should be noted that even in the situation that three correlators run at the same time, no data was lost for more than 30 mins.

5. CONCLUSION

In conclusion, we present correlation algorithms that by processing the interphoton time, rather than the time trace, obtain high efficiency. This allows us to build a software-based correlator capable of displaying correlation curve, time trace, histogram of interphoton time in real-time while preserving the most complete information of each photon. Both the auto-correlator and the cross-correlator were built and practically validated through FCS experiments. Compared with previous work, the autocorrelation algorithm has the advantages of 1) being simpler and more understandable, 2) having a time complexity independent of the temporal resolution, 3) being more affordable and flexible. The performance of the correlation can be further upgraded by reading the data from the counter board more frequently, and making use of better acquisition devices. In addition, the algorithm holds the potential to be incorporated into other already-established software correlator for efficiency improvement.

REFERENCES

- [1] Magde D., Webb W. W. and Elson E., "Thermodynamic Fluctuations in a Reacting System - Measurement by Fluorescence Correlation Spectroscopy," *Physical Review Letters*, 29(11), 705-& (1972).
- [2] Eigen M. and Rigler R., "Sorting Single Molecules - Application to Diagnostics and Evolutionary Biotechnology," *Proceedings of the National Academy of Sciences of the United States of America*, 91(13), 5740-5747 (1994).
- [3] Hess S. T., Huang S. H., Heikal A. A. *et al.*, "Biological and chemical applications of fluorescence correlation spectroscopy: A review," *Biochemistry*, 41(3), 697-705 (2002).
- [4] Krichevsky O. and Bonnet G., "Fluorescence correlation spectroscopy: the technique and its applications," *Reports on Progress in Physics*, 65(2), 251-297 (2002).
- [5] Lippitz M., Kulzer F. and Orrit M., "Statistical evaluation of single nano-object fluorescence," *Chemphyschem*, 6(5),

770-789 (2005).

- [6] Fore S., Laurence T. A., Yeh Y. *et al.*, "Distribution analysis of the photon correlation spectroscopy of discrete numbers of dye molecules conjugated to DNA," IEEE Journal of Selected Topics in Quantum Electronics, 11(4), 873-880 (2005).
- [7] Laurence T. A., Fore S. and Huser T., "Fast, flexible algorithm for calculating photon correlations," Optics Letters, 31(6), 829-831 (2006).
- [8] Magatti D. and Ferri F., "25 ns software correlator for photon and fluorescence correlation spectroscopy," Review of Scientific Instruments, 74(2), 1135-1144 (2003).
- [9] Wahl M., Gregor I., Patting M. *et al.*, "Fast calculation of fluorescence correlation data with asynchronous time-correlated single-photon counting," Optics Express, 11(26), 3583-3591 (2003).