Journal of Microscopy, Vol. 234, Pt 3 2009, pp. 302–310
Received 23 September 2008; accepted 3 March 2009

Real-time data acquisition incorporating high-speed software correlator for single-molecule spectroscopy

L.-L. YANG*,§, H.-Y. LEE†, M.-K. WANG‡, X.-Y. LIN‡, K.-H. HSU‡, Y.-R. CHANG†, W. FANN*,† & J.D. WHITE‡
*Department of Physics, National Taiwan University, Taipei, Taiwan
†Institute of Atomic and Molecular Sciences, Academia Sinica, Taipei, Taiwan
‡Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan
§Nano Science and Technology Program, Taiwan International Graduate Program, Academia Sinica, Taipei, Taiwan

Key words. Data acquisition, fluorescence correlation spectroscopy, fluorescence time trace, real time, single-molecule detection, single-molecule spectroscopy.

Summary

Single-molecule spectroscopy and detection are powerful techniques for the study of single fluorescent particles and their interaction with their environment. We present a low-cost system for simultaneous real-time acquisition, storage of inter-photon arrival times and the calculation and display of the fluorescence time trace, autocorrelation function and distribution of delays histogram for single-molecule experiments. From a hardware perspective, in addition to a multi-core computer, only a standard low-cost counting board is required as processing is software-based. Software is written in a parallel programming environment with time crucial operations coded in ANSI-C. Crucial to system performance is a simple and efficient real-time autocorrelation algorithm (acf) optimized for the count rates (approximately 10^4 cps) encountered in single-molecule experiments. The algorithm's time complexity is independent of temporal resolution, which is maintained at all time delays. The system and algorithm's performance was validated by duplicating the signal from the photon detector and sending it to both the ordinary counter board and a commercial correlator simultaneously. The data acquisition system's robustness under typical single-molecule experimental conditions was tested by observing the diffusion of Rhodamine 6G molecules in deionized water.

Introduction

Single-molecule detection (SMD) and spectroscopy (SMS), that is, the detection and study of single nano-objects (e.g. molecules or nanocrystals) in the focus of an optical microscope, was initiated by the first room-temperature observations by Betzig in 1993 using a near-field optical microscope (Betzig, 1993). Following the demonstration that detection could also be preformed in the far-field and a special issue in *Science* in 1999 (*Science*, 1999), interest in the field developed rapidly. Whether employed to study a luminescent object directly or the interaction of a non-luminescent entity with other entities through the attachment of fluorescent dye molecules, the ability to observe phenomenon otherwise obscured in ensemble measurements has enabled theory to be directly correlated with experiment – in some cases confirming and in other cases overturning accepted common knowledge (Wang, 2003). Perhaps the most well-known triumph of single-molecule techniques has been in establishing the nature of the interaction between the motor protein kinesin and the cytoskeleton's microtubules, thus resolving the long-standing debate on whether motion is inch-worm-like or hand-over-hand (Yildiz, 2004).

Figure 1 presents a schematic of a typical single-molecule detection system. A laser after passing through a narrow band filter is reflected towards the sample by a dichroic mirror. A (objective) lens is used both to focus the excitation light on to the sample and to collect the photoluminescence (PL) from the sample. The PL, after passing through the dichroic mirror, followed by either a band- or long-pass filter is focussed onto a confocal aperture (in the figure here, an optical fibre doubles as an aperture). Light that passes through the aperture is detected photon by photon, and the data stream is directed via a suitable interface to a computer for storage and further processing. Studies are generally completed in either the condensed or liquid phases. In the first, the molecule of interest are diluted and immobilized in an optically inert matrix and the sample is

Correspondence to: Jonathon David White. Tel: 886-3-463-8800; fax: 886-3-435-4604; e-mail: whitejd@xiaotu.com

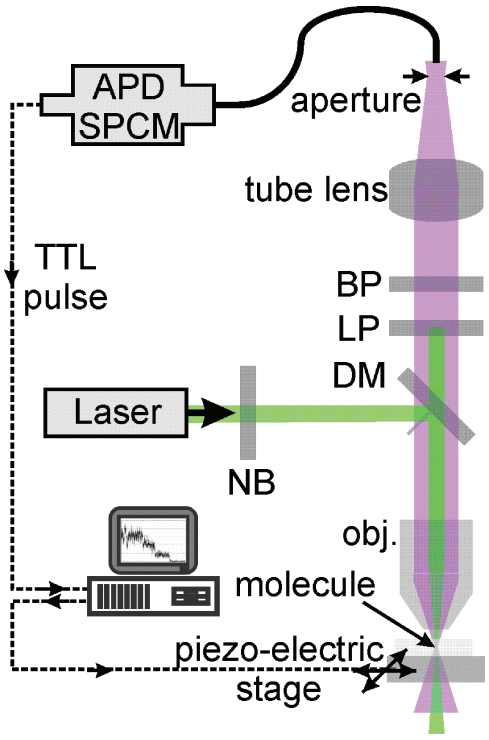

Fig. 1. Single-molecule detection system using a confocal microscope to monitor the fluorescence from single molecules. Laser light after passing through a narrow band filter (NB) is reflected off a dichroic mirror (DM) and focussed on the sample by a high numerical objective lens (obj.). Fluorescence is collected through the same lens, and after passing through the dichroic mirror (DM) and long-pass (LP) and band-pass (BP) filters, is focussed by a tube lens into an apertured optical fibre. The optical fibre directs the photon to a single photon counting APD, which generates a TTL pulse for every incident photon. This signal is directed to a counting card. The region of interest is positioned at the laser focus manually (in the case of diffusion studies) or by using a computer-controlled peizo-electric stage in the case of fixed single molecules.

raster-scanned until only a single molecule is within the focal volume. In the second, the laser is focussed into the sample and one waits until the molecule of interest enters the focal spot. To facilitate data interpretation, the molecule of interest is diluted into an optically inert fluid such that on average 0.1–1 molecule is within the focal region at one time. In the case that the molecule is being used as a probe for another system, care must be taken in selecting a probe whose absorption and emission lie in a region in which the system being probed is optically inert.

The resulting data stream (a series of TTL pulses) is generally processed and displayed in one of three ways. The simplest, cheapest and most common is to record fluorescence as a function of time (fluorescence time trace) using a low-cost counting board. Often in this type of experiment, a temporal bin size of approximately 10 ms is chosen in order to optimize the signal-to-noise ratio on the display allowing changes in fluorescence with time to be observed (White, 2002). A second method in common use is that of autocorrelation. In this case, the data stream is directed towards a hardware autocorrelator before display. While autocorrelation has the advantages of allowing information to be extracted at relatively low signal-to-noise ratios, real-time autocorrelation traditionally has required dedicated, relatively expensive hardware. In addition, there is the assumption that the process is stationary (Krichevsky, 2002; Lippitz, 2005). A third method involves plotting the distribution of delays (Lippitz, 2005), alternatively denoted as the photo arrival time histogram (PART) (Fore, 2005), obtained from time-correlated single-photon counting (TCSPC). While powerful, this technique requires expensive hardware designed explicitly for TCSPC. For many experiments, it is desirable to work with more than one of these techniques simultaneously in order to gain maximum insight into the phenomenon under investigation (i.e. an autocorrelation function can tell you the characteristic time for a change in state, while the time trace indicates the actual points in time at which the changes in state occur). One way of doing this is by using a router to split the signal between a counter board and a hardware autocorrelator. Alternatively, TCSPC boards offer a dedicated hardware/software solution albeit at a high price.

In this work, we present a system allowing for simultaneous real-time acquisition and display of the fluorescence time trace, autocorrelation function and distribution of delays histogram. Rather than binning data, the arrival time of each photon is saved to allow for post-processing with minimum loss of information. To do all of this in real time, we have written a simple and highly efficient software correlation algorithm optimized for the count rates encountered in single-molecule experiments. Its high efficiency allows real-time software correlation using only the low-cost counter board commonly used in labs to record the fluorescence time trace.

The key factors that make this possible are the low signal rate in single-molecule experiments (approximately 10^4 cps), the existence of multiple cores in modern PCs allowing for parallel processing, coding of time critical components in ANSI-C and the use of National Instrument's LabVIEW language that allows tasks to be assigned to individual cores. In the following sections, we will discuss first the hardware configuration and then the software configuration, including the algorithm for the software correlator. Finally, we will present experimental results for both deterministic signals and for the diffusion of Rhodamine 6G in deionized water. All software is available in the on-line supporting materials.

Hardware

Figure 2 illustrates the key aspects of the hardware. The TTL pulses are directed to a National Instruments PCI-6602 counting card that contains an 80 MHz clock (12.5 ns time resolution) and a 20 MHz clock (50 ns time resolution).

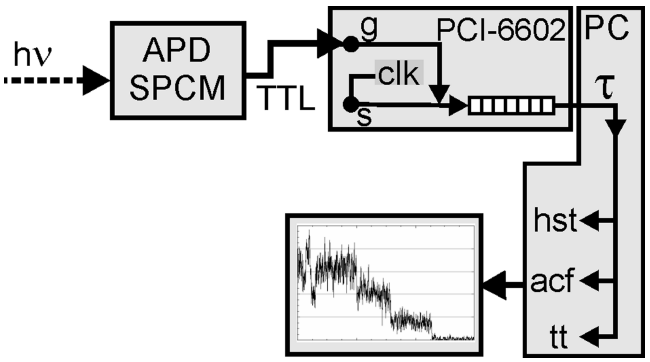

Fig. 2. Flow chart of real-time data acquisition, processing and display system controlled by LabVIEW. For each photon, a single TTL pulse is generated by an avalanche photo diode single-photon counting detector (SPCM APD) detector. The TTL pulse is used to gate the internal clock of the PCI-6602 (National Instruments, Austin, TX, USA) board. LabVIEW (National Instruments) is used to control the system. The main program encompasses four loops that run in parallel. One loop transfers the number of clock (τ) to the computer memory. Another loop is responsible as each photon arrives; its arrival time relative to the previous photon is added to the inter-photon arrival time histogram (hst). A third loop updates the autocorrelation function (acf) and a fourth loop bins the arrival time data to produce the classical time trace (tt). 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 the second computer, which is used to display information and control experimental parameters.

In the standard implementation of this card, the incoming TTL pulses are used as the source and the clock is used as a gate. The TTL pulses are then binned with a bin-size that is an integral multiple of the number of clock cycles outputting the counts/clock cycle to the host computer. In our implementation, the leads are switched so that the clock acts as the source and the TTL pulses act as the gate. The result is that the output of the card is the time interval between successive photons in the unit of clock cycles [i.e. inter-photon time (τ)]. This allows for a significantly reduced data transfer rate between the card and the host computer as can be seen using a simple example. Consider a typical single-molecule count rate of 10^4 cps. The conventional set-up for 100 ns time resolution requires a data transfer rate of 10 MHz from card to computer. However, because of the low count rate, only 0.1% of the bins are non-zero. In our implementation, the data transfer rate is equal to the photon count rate, that is, 10 kHz. (Note that while increasing resolution to approximately 10 ns would require a data transfer rate of 100 MHz in the conventional set-up, no increase is required in the inter-photon time implementation as the data transfer rate is proportional to count rate not resolution.) The second key aspect of the hardware is the use of a four-core computer. This allows one core to be involved in storing the incoming data stream and displaying data, while the other three cores read from the data stream and calculate the fluorescence time trace, autocorrelation function and delay time histogram.

Software

Overview

The algorithms were coded in ANSI-C language (for speed and portability) and compiled in the form of dynamic link library (DLL). National Instrument's LabVIEW was used as the user interface, to call the DLLs and to display the real-time experimental results (Fig. 3). LabVIEW was chosen not only because it includes drivers to support the counting card but more importantly the language supports parallel processing. In addition, using the real-time extension, the compiled program can then be deployed and run on a target computer with a real-time operating system, while the data are transferred by an ethernet cable back to the host computer where experimental parameters are set and the data are displayed. (In this work, we have used the program both with and without real-time employment. The original ANSI-C code and the LabVIEW VIs are available in the online supplementary materials.). In the program, the obtained inter-photon time (τ) is simultaneously saved to a file and processed by the DLL function running our correlation algorithm. In parallel, both the time trace of the photon count rate and the histogram of the inter-photon time are also updated based on τ. The correlation function, histogram and time trace are then put into memory, from which the data are transmitted through the ethernet cable to the host computer for display in the case of real-time deployment or displayed in the case of a single computer. In real-time deployment, while no data are lost, the refresh rate of the display is limited by the bandwidth and the speed of the host computer (especially when the array size is large).

Software-based high-speed correlation function

The key time-critical component of the processing is the calculation of the autocorrelation function. The basic autocorrelation formula for autocorrelation at time lag k for a discrete series of n samples, $X = \{X_0, X_1, \ldots X_i, \ldots X_n\}$ is

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

where

–σ is the standard deviation,

–μ is the mean and

–k is the current time lag.

Ignoring normalization and assuming that the process is stationary (i.e. no permanent photobleaching), this summation can be simplified to

$$\mathrm{acf}\,(k) \propto \sum_{t=1}^{n-k} X_t\, X_{t+k}.$$

As the time complexity is $O(n^2)$, the correlation process is time-consuming even for post-processing, let alone real-time

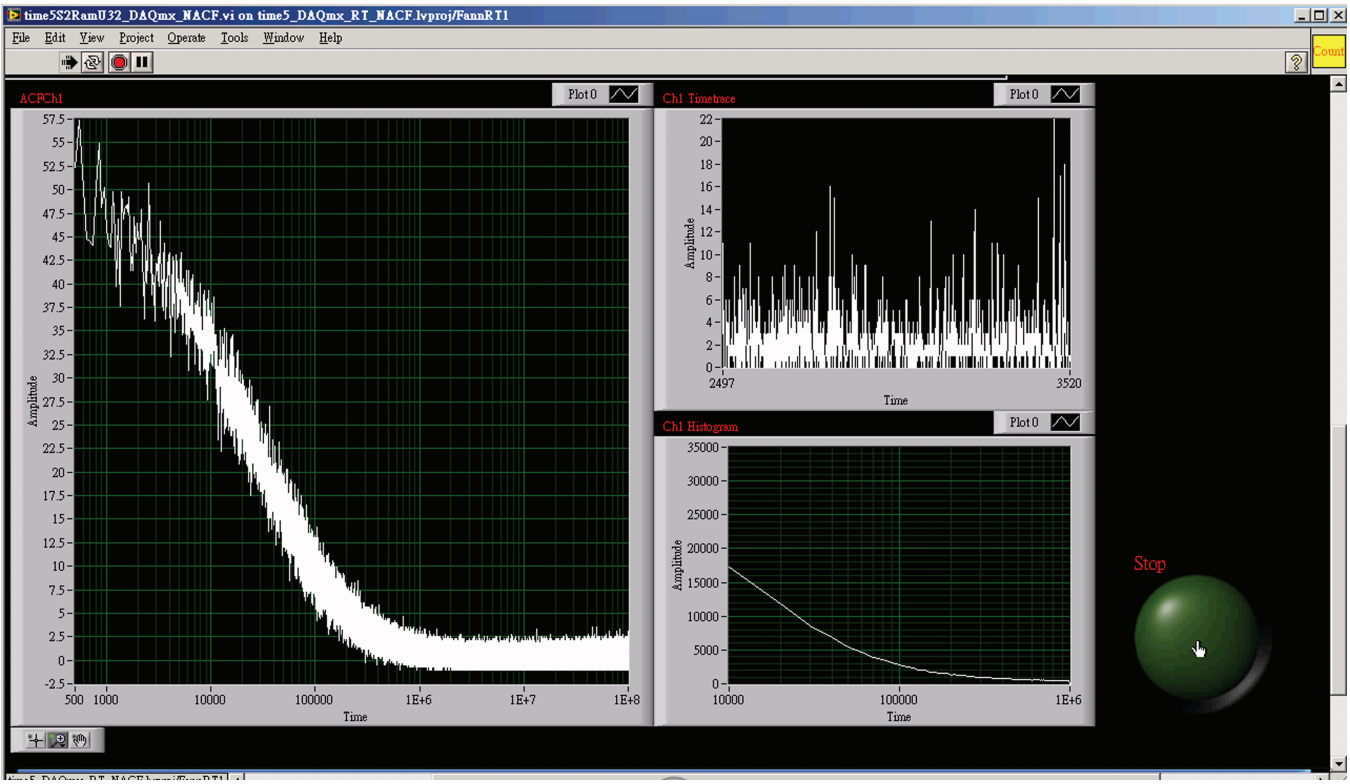

Fig. 3. User interface of the real-time data acquisition system (DAQ). Three different analyses, including the auto-correlation (left panel), time trace (right upper, in the unit of counts/10 ms) and the photon arrival time interval distribution (right lower, in the unit of ns) are calculated and displayed simultaneously in real time. (The LabVIEW VIs and ANSI-C source code for each module are included in the on-line supplementary materials.)

display during signal acquisition. Numerous techniques have been proposed to tackle this problem. The direct approach to calculating the acf makes use of the number of pulses arriving in a given time interval. Fast electronics and hardware correlators are employed to improve the computing efficiency, while a series of algorithms, named multi-tau algorithms, vary time resolution in order to reduce the array size of the autocorrelation function. Unfortunately, these can lead to aliasing. In addition, while having the advantage of being able to work with high count rates, the direct approach is not efficient for SMS as the low count rates involved lead to many time bins having zero counts when high time resolution is required. Recently, several authors have proposed algorithms that rely on the time sequences of the photon arrival times either alone (Laurence, 2006) or in combination with the traditional clock-based techniques (Magatti, 2003). However, both of these algorithms are quite complicated to implement.

By contrast, the acf algorithm presented here is simple and straightforward. It makes use of the inter-photon times provided by the data stream to update the acf in real time. Two arrays are used, one to hold the autocorrelation function (acf) and one (delay) that serves as a look up table indicating which elements of the acf need to be updated after the arrival of a new photon. 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 = 1; k < m; k++)  acf[delay[k]]++;
```

In this algorithm, m is the last element of the array to be updated. Furthermore, $m < n$, where n is the number of expected photons in the maximum time interval for which the autocorrelation is to be performed (e.g. if the count rate is 10^4 cps and one desires to calculate the autocorrelation function out to 100 ms, $n = 10^3$), independent of the desired temporal precision or clock frequency. *mtau* is the maximum interval at which to perform the autocorrelation. The key advantage of this algorithm is that n is independent of the required time precision. Figure 4 illustrates the operation of the algorithm for a series of inter-photon times $\tau = \{3,2,3,9,5\}$ corresponding to $X = \{1,0,0,\ 1,0,1,\ 0,0,1,\ 0,0,0,\ 0,0,0,\ 0,0,1,0,0,0,0,1\}$ if processed using the clock-based techniques. In this example, $mtau = 15$ clock cycles. The first photon that is detected starts the process. Three clock cycles (of the counter board) later the second photon arrives. As there

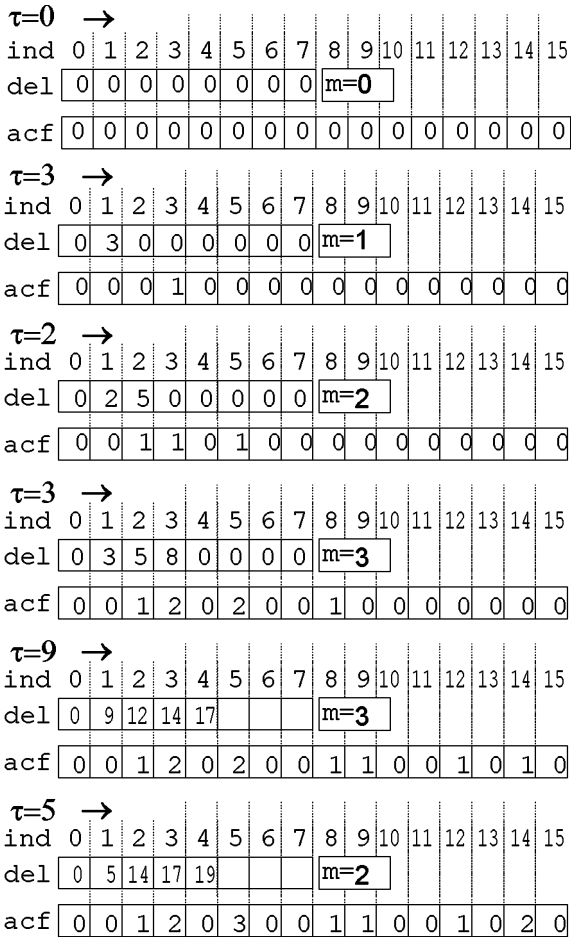

Fig. 4. Sample calculation of the raw autocorrelation function for the arrival to a series of six photons under the assumption that the maximum delay time of interest is equal to 15 clock cycles.

is nothing in the array, the first loop does nothing, the arrival time is simply stored in position 1 of the delay time array, and element 3 of the acf is incremented. Two counter board clock cycles later, the third photon arrives. The first loop updates position 2 of the delay time array by adding the current τ with the previous delay times and stores the arrival time in position 1 of the delay time array. In the third loop, elements 2 and 5 of the acf are incremented. The purpose of the second loop, which on average will only have two cycles, is to remove points for which the delay time is greater than the maximum desired delay time. The process is continued as each photon arrives as shown in Fig. 4.

The efficiency of this algorithm can be seen through a simple comparison with a brute force approach. The following assumptions are employed: addition takes four, multiplication four and the shift and comparison operations both take one clock cycle (of the host computer) on a 1 GHz computer, a temporal resolution of 100 ns is required with a maximum time window of 100 ms, and data arrive at a typical single-molecule detection count rate of 10^4 cps. Under these assumptions, the autocorrelation function requires a 10^6 element array. In a brute force application of the correlation algorithm, the n used for time complexity is equal to the size of the autocorrelation array. As this calculation requires $4n$ additions and $4n$ multiplications for each data point – that is, approximately 10^7 clock cycles, or about 10 ms for each data point. Considering that data points arrive every 100 ns, real-time operation is clearly not possible without using dedicated hardware and/or advanced algorithms. By contrast, for the proposed algorithm, $n = 10^3$ – the number of photons arriving in the maximum time interval for which the correlation is to be preformed. Thus for each data point additions, comparisons and shift operations require a total of 6×10^3 clock cycles or approximately 6 μs to update the autocorrelation function. With an average count rate of 10^4 cps, each 1 data point arrives every 100 μs – thus there is adequate time to update the correlation function in real time. Note that the above calculation is independent of the time resolution, while for the brute force method, the time requirements increase quadratically. In summary, if we define the duty cycle (d) as the ratio of the average count rate and the clock frequency (time resolution), when d is much less than unity, as is generally the case in single-molecule experiments, efficiency will be greatly improved.

Experimental validation

The single-molecule data acquisition system and autocorrelation algorithm was first validated using a 1-MHz deterministic count rate source – a count rate two orders of magnitude higher than that expected for single-molecule experiments. The system was able to keep updating the acf function, histogram, time trace on the screen, and all the inter-photon time data were stored on the computer's hard drive without any data loss due to time taken by the correlation process. The system was next used to investigate the emission properties of 100-nm-diameter fluorescent nano-diamonds (FND) containing numerous fluorescent $(N\text{-}V)^-$ centres. For this experiment, the FNDs were immobilized. The lack of correlation between emitters for this sample suggests that there is little communication between individual $(N\text{-}V)^-$.

Additional experimental confirmation of the above system was provided by the classical dye molecule diffusion experiment in which a dilute mixture of dye molecules is observed using the apparatus shown in Fig. 1. By fitting the resulting acf, the concentration, diffusion time and average number of molecules in the focal region can be determined. Rhodamine 6G (R6G) organic dye molecules (Molecular probes, Invitrogen, Carlsbad, CA, USA) were dissolved in and diluted with deionized water. Two different concentrations, (nominally approximately 1 nM and approximately 10 nM) were used to fill the cavity on a caved glass slide, covered with a 22 × 22-mm cover slip and sealed with transparent silicon. Optical excitation

was provided by a continuous-wave frequency-doubled diode-pumped solid-state laser ($\lambda = 532$ nm). After passing through a narrow-pass filter and being reflected by a dichroic mirror (540 DCLP), the light was focussed into the solution approximately 5 μm from the interface between the cover slip and the dye solution. The excitation power for the low- and high-concentration sample is 37.4 and 3.18 μW, respectively, measured without the objective. The fluorescence generated by the sample was collected with the same objective, separated from the Rayleigh and Raman scattering with a band-pass filter ($\lambda_0 = 575$, $\Delta\lambda = 50$, critical for suppressing the Raman of the water), and then refocussed by the tube lens through a 50-μm pinhole and into a 200-μm core fibre leading to the photon detector (SPCM-APD, PerkinElmer, Waltham, MA, USA). To validate the algorithm, the TTL pulse generated by the photon detector after passing through a digital filter to eliminate pulses shorter than 25 ns was split into two pulses with half voltage level in two different cables by a pulse divider. The pulses were recovered to TTL pulses by fast electronics and delivered to our system and a commercial correlator (ALV-5000) for real-time comparison. As the typical diffusion time for R6G in dilute water is tens to hundreds of microseconds depending on the focal spot size and the fact that the maximum temporal resolution of our ALV correlator is only 125 ns, we chose to use the 20-MHz clock (50 ns time resolution) rather than the 80-MHz clock on the counting board. Data in this experiment were continuously recorded over 30 min.

Figure 5 presents typical experimental results for the autocorrelation after 30 min of data acquisition. For both the low- and high-concentration samples, the correlation curve, as shown in the figure, coincides with that from the commercial correlator, except for the noisier presentation at long time lags. This 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 50 ns time resolution over the complete range, that is, the curve will become as smooth if logarithmic binning, similar to that used in the commercial system is applied.

To determine the diffusion time, focal volume and average number of molecules within the focal region, the autocorrelation curves were fit to

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

where a_0, a_1 and a_2 are adjustable parameters. Under the assumption of a Gaussian beam profile, a knowledge of these parameters allows the diffusion time (τ_{diff}) the focal volume, average number of molecules in the focal region (N), and hence concentration to be obtained through comparison with the equation (Eiger, 1994):

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

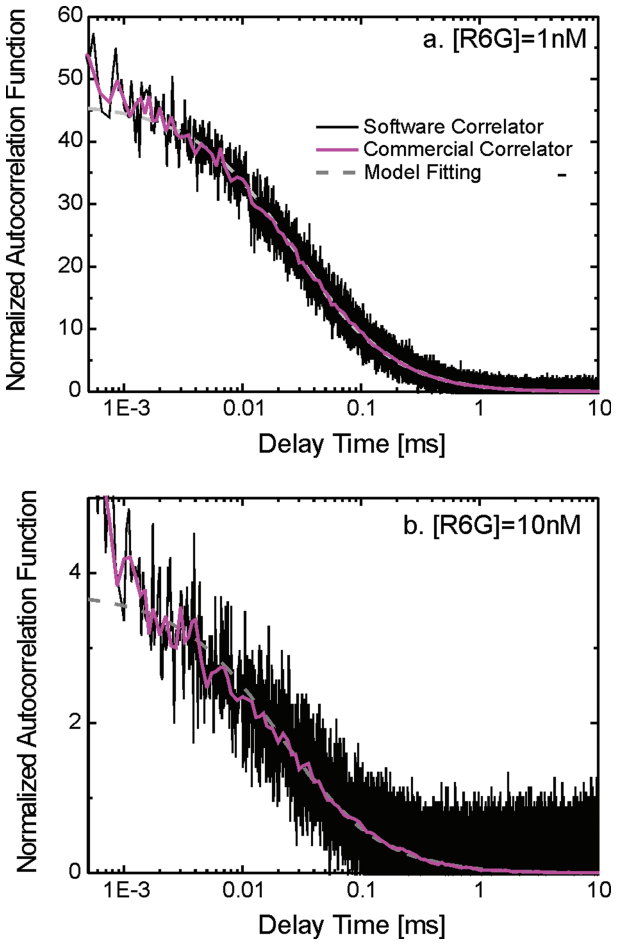

Fig. 5. Comparison of the calculated auto-correlation function of the fluorescence time trace of (a) low (approximately 1 nM) and (b) high (approximately 10 nM) concentration of R6G molecules diffusing in water compared with that generated by a commercial correlator. The photon stream leading to the two correlators is identical, and is acquired during the same period. The dashed curve represents a fitting to the experimental data (see text for details).

The results for two different dye concentrations are summarized in Table 1. The results are both self-consistent and consistent with the nominal concentrations used in this experiment. In addition, the diffusion times fall in the range of values commonly measured for small molecules.

Discussion

Currently, the maximum count rate is limited not by the algorithms used here, or the speed of data processing, but rather by the small size of the first-in first-out (FIFO) buffer available on the counter board (16 × 32-bit). As has been discussed by other authors (Magatti, 2003), at high count rates it is increasingly likely that a bunch of photons will surge in and saturate this buffer resulting in a buffer overwrite error. The simplest way of dealing with this problem (and the

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	
	~1 nM	~10 nM
a_0	46	3.7
a_1	27178	20305
a_2	774255	1120000
Diffusion time, μs (τ_{diff})	27	20
Molecules in focal region (N)	0.02	0.26
Focal volume, fl	0.16	0.14
Concentration, nM	0.24	3.1

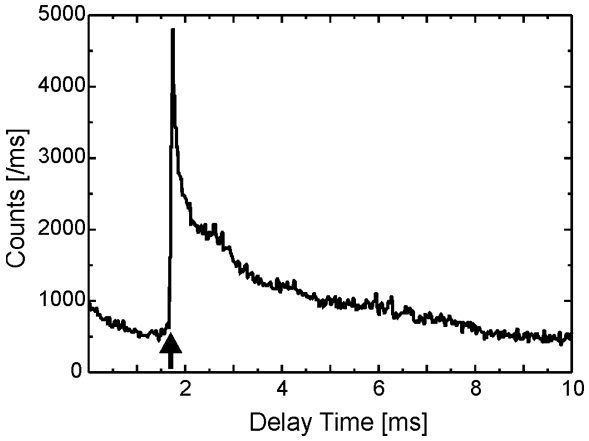

Fig. 6. Recorded time trace of photoluminescence of an MEH-PPV thin film. The amplitude scale is counts per 10 ms time window. The vertical arrow indicates the point when the excitation position was abruptly shifted to a new unphotobleached area of the film. The high count rate (>2 MHz) resulted in the data acquisition system freezing for about 10 s. Data acquisition and display resumed when photobleaching of the sample resulted in the count rate dropping below 4500 counts/10 ms.

way used in this program) is to simply clear the FIFO buffer and restart counting, resulting in the loss of photon arrival times immediately before and after the error. In terms of the correlation function calculations, m is simply reset to zero and correlation is resumed. This works well provided that the count rate is low enough so that the average period between buffer overwrite errors is longer than any expected correlation signal as well as the time required to reset the buffer on the PCI-6602 card.

With that in mind, we decided to investigate the maximum count rates that could be sustained by the data acquisition system. Using a deterministic pulse, count rates of 1 MHz for more than 1 min without error and 2.5 MHz over a 5-s window were sustained without error. However, when dealing with actual experimental data, the achievable count rate depends on the number missed photons that one is able to accept. For example, in the Rhodamine diffusion experiment reported earlier, where the average count rate was 500 cps, there were no data acquisition errors caused by photon bunching. To test the maximum average count rate that could be sustained under experimental conditions, we excited a thin film of MEH-PPV (not a single-molecule sample) and attempted to record data as the sample photobleached. The results are shown in Fig. 6. At the initial photon flux rate of >2 MHz, the data acquisition software was not able to acquire data owing to numerous FIFO errors coming from the PCI-6602 card. It was only when the count rate dropped to 500 kHz was it possible to start recording data (see figure) – albeit with a large number of missed photons. Stable performance (i.e. few FIFO errors) was obtained once the count rate dropped below approximately 200 kHz. At the latter count rate, an average of between four and six FIFO overwrite errors per minute due to photon bunching were recorded. At lower count rates, data acquisition was error-free.

One way of removing this limitation is to replace the PCI-6602 card with hardware having a larger FIFO buffer. This can be achieved, along with further reduction in cost, by replacing PCI-6602 card with a simple two-part circuit. The input part implements a counter circuit using a single IC and a single (60 MHz) clock chip that writes the inter-photon time information into a 100-MHz FIFO memory buffer. The output component reads the data from the FIFO buffer and sends the data via RS232 protocol to the host computer. The average count rate is limited by the 115000 baud rate of the RS232 protocol. Assuming that the maximum time interval between successive photons is 1 s and the time resolution is 50 ns, 24 bits are required to time stamp each photon translating to a maximum average count rate of 5 kHz. While this is not as high as that obtained using the PCI-6602 board, it is sufficient for single-molecule experiments. The circuit does however offer improved performance in terms of the maximum instantaneous count rate that can be handled under experimental conditions. Using a FIFO chip (CY7C4251) with an 8-kB buffer, bursts of up to approximately 2800 photons can be handled by the circuit [i.e. if one photon arrives every 100 ns (10 MHz), data can be recorded for 280 μs before a buffer overwrite error occurs], clearly eliminating the buffer overflow problems caused by photon bunching. The total cost of the circuit components is less than US$40, with the expenses being dominated by the high-speed FIFO buffer (US$30) – a significant cost-down for the implementation.

The algorithm presented here calculates full precision data for the whole range of ACF delay times, which is not strictly necessary (i.e. 100 ns time resolution may not be required at 10 ms time delays). This results in the correlation function that appears to be noisier than it really is (cf. Fig. 5). For real-time display, it may be preferable to bin data at long-time delay so

as to smoothen the display while simultaneously improving the performance of the program by reducing the amount of data being sent to the computer's video display memory. The following algorithm does this simply and quickly making use of the fact that dividing by 2 is a simple one-cycle bit-shift operation (>>) in ANSI-C.

```
bitshift = 0;
offset = 0;
for (k = 1;  k <= m;  k++){
     d = delay [i] >> bitshift;
     while (d > 255){
       bitshift += 8;
       offset += 255;
       d = d >> 8;
     }
     acf [offset + d]++;
}
for (k = 0;    k < 256;  k++) display [k] = acf [k];
for (k = 256;  k < 512;  k++) display [k] = acf [k] >> 8;
for (k = 512;  k < 768;  k++) display [k] = acf [k] >> 16;
```

where offset is the interval for which a constant time period is used and bitshift is the number of bits that time delay should be shifted and display is the autocorrelation function data sent to the display unit. In the above algorithm, the time resolution for the first 256 bins of the acf function is equal to the clock period (i.e. 100 ns for time delays less than 25.6 μs). For the next 256 bins, the resolution is reduced 256 times (i.e. to 25.6 μs for time delays greater than 25.6 μs and less than 6.5 ms). For the next 256 bins, the resolution is further reduced by a factor of 256 (i.e. to 6.5 ms for delay times greater than 6.5 ms and less than 1.6 s). Note that the while loop will run, at most twice as the values in the acf array are monotonically increasing. For display purposes, the total counts in the acf function bins are scaled back down using the same bit shift operations that were used in binning. The result of these calculations significantly reduces the workload required by the graphics drivers (i.e. 768 data points are sent rather than a million data points). However, it should be noted that using any multi-tau algorithm may result in aliasing. Thus, it is important to save the raw inter-photon data for later review using different values for binning to check that the observed effects are real.

The generalization of the above software autocorrelation algorithm to cross-correlation is relatively straightforward. Unfortunately, if one seeks to use a single PCI-6602 card for data acquisition, the average experimental count rate is limited to approximately 500 Hz, as the card uses a common buffer and buffer overflow errors are more likely to occur owing to photon bunching. To overcome this problem, it is necessary either use two PCI-6602 cards in parallel, design one's own counting circuit, or maintain the average count rates to be below 500 Hz.

Conclusions

In conclusion, we have presented a system for simultaneous real-time acquisition, storage of inter-photon arrival time data and display of the fluorescence time trace, autocorrelation function and distribution of delays histogram applicable for single-molecule spectroscopy. This is made possible by assigning different processing tasks to different processors in a multi-core computer making use of the LabVIEW programming language as well as the development of a high-speed algorithm for performing autocorrelation. Compared with previous work, this autocorrelation algorithm has the advantages of (1) being simpler and more understandable, (2) maintaining time resolution at all time-scales throughout the correlation curve (i.e. up to 1 s at the count rates common in single-molecule experiments) and (3) having a time complexity independent of the temporal resolution. Comparing to other solutions, this acquisition system is more flexible and cost-effective relative to one based on commercially available hardware correlators or TCSPC cards. We note that the autocorrelation algorithm developed here is also effective for post-processing of experimental data.

Acknowledgements

This research is supported by grants NSC no. 96-2752-E-007-007-PAE, NSC no. 96-2752-E-007-008-PAE and NSC no. 95-2112-M-155-001 from the National Science Council of the Republic of China.

Memorial note

We regret the passing away of Prof. Wunshain Fann after a decade long personal battle against cancer at the age of 47. During his career, he published more than 100 original papers in the areas of free electron lasers, near-field optical microscopy, single-molecule spectroscopy and charge transport within conjugated polymer systems. His positive attitude, coupled with his concern for the well-being of his graduate students, post-docs and visiting researchers, along with his love of research remains a model for those of us who had the privilege to know him and work under his leadership. We extend our sympathy to his wife Katherine and two precious young daughters.

References

Betzig, E. & Chichester, R. (1993) Single molecules observed by near-field scanning optical microscopy. *Science* **262**, 1422.

Eiger, M. & Rudolf Rigler, R. (1994) Sorting single molecules: application to diagnostics and evolutionary biotechnology. *Proc. Natl. Acad. Sci. U.S.A.* **91**, 5740–5747.

Fore, S., Laurence, T.A., Yeh, Y., Balhorn, R., Hollars, C.W., Cosman, M. & Huser, T. (2005) Distribution analysis of the photon correlation spectroscopy of discrete numbers of dye molecules conjugated to DNA, Selected Topics in Quantum Electronics. *IEEE J. Select. Topics Quantum Electron.* **11**, 73–880.

Krichevsky, O. & Gregoire Bonnet, G. (2002) Fluorescence correlation spectroscopy: the technique and its applications. *Rep. Prog. Phys.* **65**, 251–297.

Laurence, T.A., Fore, S. & Huser, T. (2006) Fast, flexible algorithm for calculating photon correlations. *Opt. Lett.* **31**, 829–831.

Lippitz, M., Kulzer, F. & Michel Orrit, M. (2005) Statistical evaluation of single nano-object fluorescence. *Chemphyschem.* **6**, 770–789.

Magatti, D. & Ferria, F. (2003) 25 ns software correlator for photon and fluorescence correlation spectroscopy. *Rev. Sci. Instrum.* **74**, 1135–1143.

Special Issue on Single Molecules. (1999) *Science.* **283**, 1593–1804.

Wang, C.F., White, J.D., Lim, T.L., *et al.* (2003) Illumination of exciton migration in rod-like luminescent conjugated polymers by single molecule spectroscopy. *Phys. Rev. B.* **67**, 035202-1–035202-8.

White, J.D., Hsu, J.H., Wang, C.F., *et al.* (2002) Single molecule fluorescence spectroscopy. *J. Chinese Chem. Soc. (Taipei)*. **49**, 669–676.

Yildiz , Y. & Tomishige, M. (2004) Ronald D. Vale, and Paul R. Selvin, kinesin walks hand-over-hand. *Science.* **303**, 676–678.

Supporting Information

Additional Supporting Information may be found in the online version of this article:

Appendix S1. Complete LabVIEW VI for Real-time Data Acquisition System. Includes program explanation (File: labVIEW-VI-Explanation.pdf), DLLs. The main program is ACF. vi. It has been tested in LabVIEW 8.5.

Appendix S2. ANSI-C source code for elements of the autocorrelation function and algorithm for calculation of the autocorrelation function, photon arrival time histogram, and time trace along with sample data file..

Please note: Wiley-Blackwell are not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.

