

Fast calculation of fluorescence correlation data with asynchronous time-correlated single-photon counting

Michael Wahl¹, Ingo Gregor², Matthias Patting¹, Jörg Enderlein²

¹ PicoQuant GmbH, Rudower Chaussee 29 (IGZ), D-12489 Berlin, Germany

²Institute for Biological Information Processing I Forschungszentrum Jülich 52425 Jülich, Germany
j.enderlein@fz-juelich.de

Abstract: Fluorescence correlation spectroscopy (FCS) is a powerful spectroscopic technique for studying samples at dilute fluorophore concentrations down to single molecules. The standard way of data acquisition, at such low concentrations, is an asynchronous photon counting mode that generates data only when a photon is detected. A significant problem is how to efficiently convert such asynchronously recorded photon count data into a FCS curve. This problem becomes even more challenging for more complex correlation analysis such as the recently introduced combination of FCS and time-correlated single-photon counting (TCSPC). Here, we present, analyze, and apply an algorithm that is highly efficient and can easily be adapted to arbitrarily complex correlation analysis.

© 2003 Optical Society of America

OCIS code: (170.6280) Spectroscopy, fluorescence and luminescence; (180.1790) Confocal microscopy; (300.2530) Fluorescence, laser-induced

References

1. C. Zander, J. Enderlein, R.A. Keller (Eds.) *Single-Molecule Detection in Solution - Methods and Applications* (VCH-Wiley, Berlin/New York, 2002).
2. W. Becker, H. Hickl, C. Zander, K.H. Drexhage, M. Sauer, S. Siebert, J. Wolfrum "Time-resolved detection and identification of single analyte molecules in microcapillaries by time-correlated single-photon counting (TCSPC)," *Rev. Sci. Instrum.* **70**, 1835-41 (1999).
3. M. Böhmer, F. Pampaloni, M. Wahl, H.J. Rahn, R. Erdmann, J. Enderlein "Advanced Time-Resolved Confocal Scanning Device For Ultrasensitive Fluorescence Detection," *Rev. Sci. Instrum.* **72**, 4145-52 (2001).
4. C. Eggeling, S. Berger, L. Brand, J.R. Fries, J. Schaffer, A. Volkmer, C.A.M. Seidel "Data registration and selective single-molecule analysis using multi-parameter fluorescence detection," *J. Biotechnol.* **86**, 163-80 (2001).
5. M. Böhmer, J. Enderlein "Single molecule detection on surfaces with the confocal laser scanning microscope," in ref.[1], pp.145-83.
6. R. Rigler, E. Elson (Eds.) *Fluorescence Correlation Spectroscopy* (Springer, New York/Berlin, 2001).
7. D.C. Lamb, A. Schenk, C. Röcker, C. Scalfi-Happ, G.U. Nienhaus "Sensitivity Enhancement in Fluorescence Correlation Spectroscopy of Multiple Species Using Time-Gated Detection," *Biophys. J.* **79**, 1129-38 (2000).
8. M. Böhmer, M. Wahl, H.J. Rahn, R. Erdmann, J. Enderlein "Time-resolved fluorescence correlation spectroscopy," *Chem. Phys. Lett.* **353**, 439-45 (2002).
9. J.S. Eid, J.D. Müller, E. Gratton "Data acquisition card for fluctuation correlation spectroscopy allowing full access to the detected photon sequence," *Rev. Sci. Instrum.* **71**, 361-8 (2000).
10. D. Magatti, F. Ferri "25 ns software correlator for photon and fluorescence correlation spectroscopy," *Rev. Sci. Instrum.* **74**, 1135-44 (2003).
11. K. Schätzel "Correlation techniques in dynamic light scattering," *Appl. Phys. B.* **42**, 193-213 (1987).
12. K. Schätzel, M. Drewel, S. Stimac "Photon correlation measurements at large lag times: Improving statistical accuracy," *J. Mod. Opt.* **35**, 711-8 (1988).

13. P. Schwille, F.J. Meyer-Almes, R. Rigler "Dual-color fluorescence cross-correlation spectroscopy for multicomponent diffusional analysis in solution," *Biophys. J.* **72**, 1878-86 (1997).
 14. M. Höbel, J. Ricka "Dead-time and afterpulsing correction in multiphoton timing with nonideal detectors," *Rev. Sci. Instrum.* **65**, 2326-36 (1994).
-

1. Introduction

An important issue when detecting fluorescence signals at very low light intensities as e.g. in single molecule spectroscopy (SMS) [1] is the occurrence of long time intervals with few or no photon detection events. Usually photon detection intensities are continuously monitored within consecutive macroscopic time bins of equal width (e.g., 100 ns to 1 ms). This sampling mode becomes increasingly problematic at short bin width (i.e., high time resolution) under low light intensities such as applied in SMS. In this case, most bins are accounted by zero photon intensity and, thus, of no worth. For example, sampling photon detection intensities with a temporal resolution of 100 ns requires recording of 10^7 intensity values per second, although most of these values will be zero for signal intensities usually encountered in SMS experiments (typical count rates are between 10^3 and 10^5 counts per second). The situation becomes even more dramatic when desiring to measure not only fluorescence intensities but also fluorescence lifetime by using pulsed laser excitation and correlating the photon detection times on a nanosecond timescale with the pulse times of the laser, i.e. direct determination of the delay time between excitation and detection (time-correlated single-photon counting or TCSPC). The conventional approach [2] requires of building up complete TCSPC histograms for every time bin. In case of TCSPC histograms containing $2^8 = 256$ delay time channels and a moderate counting depth of only 4 bit per channel, one ends up with a data flow rate of $4 \times 256 \times 10^7$ bit/s = 1.28 GByte/s, which is beyond any reasonable limit. A much more advantageous mode of data acquisition than the synchronous acquisition mode is to record data only in case of an actual photon detection event (asynchronous data acquisition). This is the philosophy behind the so-called time-tagged time-resolved (TTTR) data acquisition mode that was recently introduced in low-intensity fluorescence measurements [3-5]. In that mode, every photon detection event generates two numbers: the time-correlated single-photon correlation (TCSPC) time (photon arrival time with respect to most recent laser pulse), and the macroscopic arrival time on a continuous time axis with ca. 100 ns time resolution (time tag). Only these two numbers are transferred to and stored in the PC memory. Additionally, each photon can be associated with a third tag containing information about, e.g., which photo-diode in a multi-channel detection scheme has detected the actual photon. Thus, the data flow is directly proportional to the photon detection count rate, generating no overhead of redundant information as in a synchronous data acquisition mode. Although this asynchronous TTTR data acquisition mode is now widely used in time-resolved single-molecule fluorescence spectroscopy, an important question is how to efficiently convert the TTTR data into a fluorescence correlation spectroscopy (FCS) curve. FCS is a powerful tool for evaluating SMS measurements and has found innumerable applications, see, e.g., ref.[6]. In many applications, FCS curves are generated from raw fluorescence intensity signals by dedicated correlator hardware. However, in that case any information about fluorescence lifetime or fluorescence intensity characteristics that are not reflected in the FCS curve is completely lost. Recently, combinations of TCSPC and FCS have been proposed, either for improving signal-to-noise ratio of an FCS measurement by time-gating photons on the nanosecond timescale [4,7], a virtual multi-channel FCS measurement scheme (i.e., creating several FCS data for different fluorophores present in the sample) by using the fluorescence lifetime characteristics of different fluorophores as the distinguishing parameter (TCSPC-FCS) [8]. In these cases, hardware correlators can only be

used if they are preceded by another piece of data processing hardware. Furthermore, hardware correlators must, by design, immediately process incoming data and store the results. This prohibits any sectioning of the data and elimination of unwanted sections. This may be a severe practical limitation, since measurements that are accidentally spoiled by e.g. diffusing dust particles must be completely discarded.

An alternative approach is to calculate the FCS curve by software on the basis of the directly recorded asynchronous photon count data. The straightforward approach would be to generate a continuous file of fluorescence intensity values in consecutive time bins of fixed bin width, and to use this file for a conventional autocorrelation calculation. But then again, one encounters the same unreasonably large amount of data as when directly recording fluorescence intensity values in a synchronous data acquisition mode. Recently, Eid *et al.* [9] and Magatti and Ferri [10] described new algorithms for converting asynchronously acquired photon counting data into correlograms. Here, we expand on their ideas and present an algorithm especially designed for the evaluation of TTTR photon count data, emphasizing its generalization to more complex correlation analysis techniques such as combining the autocorrelation analysis with the analysis of the fast fluorescence decay on the nanosecond time-scale (TCSCP-FCS).

2. Theory

2.1 Asynchronous single-photon counting data

Asynchronously measured time-tagged single-photon counting data consist of a linear file of detection times $\{t_1, t_2, \dots, t_N\}$ of the detected photons, where N is the total number of detected photons during a given measurement. As special feature of these detection times is that they are integer multiples of some minimal time δt , determined by the temporal resolution of the detection electronics. Without restriction of generality, it will be assumed that all times are measured in units of δt , so that all the numbers t_j take integer values. The value $g(\tau)$ of the autocorrelation function for a given lag time τ is defined as

$$g(\tau) = \langle I(t)I(t+\tau) \rangle \quad (1)$$

where $I(t)$ is the photon detection intensity at a given time t , and the brackets denote averaging over all possible values of time t . For a photon detection measurement with temporal resolution δt , the intensity values $I(t)$ within consecutive time intervals can only take the values $1/\delta t$ or 0, depending on whether there was a photon detection event during a time interval of width δt or not. The average in Eq. (1) is then calculated as the sum over all consecutive time intervals of width δt , divided by the total number of summed intervals.

2.2 Autocorrelation with logarithmic time-scale

In practice, one does not compute the autocorrelation function for all possible values of lag time τ , but at increasingly spaced lag time values. If the temporal resolution of the photon detection is, e.g., 100 ns, and one desires to follow correlation processes up to a minute, possible values of lag time τ are any value between 100 ns and 60 s in intervals of 100 ns, resulting in 6×10^8 possible lag time value. Calculation of $g(\tau)$ for all of these values would be an enormous time-consuming numerical effort. Instead, the autocorrelation is calculated for only few, approximately logarithmically spaced values of τ , which is sufficient for most FCS applications. This calculation at selected, approximately logarithmically spaced lag times was first used for hardware correlators, due to restrictions in practical complexity [11], and remains applicable for software correlators. However, the latter permit a much wider choice

of lag time spacing and can easily be reconfigured for the given experiment. A convenient choice for the values of τ is

$$\tau_j = \begin{cases} 1 & \text{if } j = 1 \\ \tau_{j-1} + 2^{\lfloor (j-1)/B \rfloor} & \text{if } j > 1 \end{cases} \quad (2)$$

with j taking integer values starting with one and running up to some maximum number $j_{\max} = n_{\text{casc}}B$; where B is some integer base number; the bracket $\lfloor \cdot \rfloor$ gives the integer part of the enclosed expression. The resulting lag times are grouped into n_{casc} cascades with equal spacing of $2^{\lfloor j/B \rfloor}$. The advantage of such a choice of lag times is that all τ_j have integer values so that fast integer arithmetic can be used in subsequent computations. For example, when using a base number value of $B = 10$ and $n_{\text{casc}} = 3$, one obtains the lag time sequence $\{\tau_j\} = \{1, 2, \dots, 9, 10, 12, \dots, 28, 30, 34, \dots, 70\}$.

2.3 Synchronous photon intensity data: bin-and-correlate algorithm

As mentioned above, a straightforward way of calculating the autocorrelation function is to divide the total measurement time, $t_N - t_1$, into intervals of unit length δt , and to sort the detected photons into these intervals corresponding to their arrival times t_j . The result is a synchronous photon detection intensity file I_j with j running from 1 through $t_N - t_1$, where the I_j can only adopt the values one or zero. The fluorescence autocorrelation can then be calculated as given by Eq. (1). In practice, such an approach is prohibitively memory demanding and computationally expensive. As an example, consider an experiment with an average count rate of 10^5 counts per second and a typical time resolution of $\delta t = 100$ ns. A measurement lasting one minute would result in an average number of $6 \cdot 10^6$ detected photons. Converting the photon arrival data into a synchronous binary intensity file with 100 ns temporal resolution results in the huge number of $6 \cdot 10^8$ time intervals, or little less than 100 MByte of data, whereas only 1 % of these will contain a nonzero value. In contrast, the asynchronous file of photon arrival times takes only an amount of 12 MByte, if one assumes that every photon is tagged with a 2 Byte long label containing its arrival time and some overflow information.

2.4 Asynchronous photon intensity data time-tag-to-correlation algorithm

An optimal FCS algorithm will work directly on the arrival times $\{t_1, t_2, \dots, t_N\}$, without converting them into time-binned data. In its simplest form, our algorithm is rather straightforward. For a given lag time τ , a second vector of arrival times $\{t'_1, t'_2, \dots, t'_N\}$ is generated, containing the time values $t'_j = t_j + \tau$. In the beginning, the value of the autocorrelation at lag time τ is set to zero. The algorithm starts with the time t_1 in the original vector and moves to consecutive time entries in that vector until it encounters a value t_j that is equal to or larger than t'_1 . If $t_j = t'_1$, the value of the autocorrelation at lag time τ is increased by one. Next, the algorithm switches to the entries of the second vector and, starting with t'_1 , moves to consecutive time entries in that vector until it encounters a value t'_{j_1} that is equal to or larger than t_{j_1} . If $t_{j_1} = t'_{j_1}$, the value of the autocorrelation at lag time τ is increased by one. The algorithm switches back to the first vector and, starting with t_{j_1} , moves to consecutive time entries in that vector until it encounters a value t_{j_2} that is equal to or larger than t'_{j_1} , and so on until the last entry in one of both vectors is reached. In this simplest form, the autocorrelation algorithm calculates, up to some constant factor, the probability of detecting a photon at some time $t + \tau$ if there was a photon detection event at time t .

2.5 Correlation time coarsening

When applying the algorithm only at the increasingly spaced lag times as given by Eq. (2), it will completely miss, e.g., the strong autocorrelation of any periodic signal with a repetition time not included within the vector of used lag times. To avoid that, one usually applies an averaging procedure by coarsening the time resolution of the photon detection times t_j when coming to the calculation of the autocorrelation function at increasingly larger lag time. This is equivalent to the multiple-tau and multiple-sampling time correlation method employed in hardware correlators [12]. Such a procedure is easily incorporated into the present algorithm. Besides working only with the original and shifted vectors of arrival times, $\{t_1, t_2, \dots, t_N\}$ and $\{t'_1, t'_2, \dots, t'_N\}$, all time entries t_j and t'_j are associated with weight values w_j and w'_j that are all set to one at the start of the algorithm. In case of an equality $t_j = t'_k$, the autocorrelation is increased by the weight product $w_j w'_k$ and not by one. A time coarsening step is inserted each time when finishing the calculations for one cascade of B lag times τ_j with equal spacing and before starting with the next cascade of B lag times with doubled spacing: All values $\{t_1, t_2, \dots, t_N\}$ used in the previous cascade are divided by two and rounded to the nearest lower integer value, which will occasionally leads to the occurrence of consecutive entries with the same time value. Before continuing the autocorrelation computation, such double entries are reduced to one entry, and the corresponding weight of that remaining entry is increased by the weight of the eliminated one. Thus, with increasing lag time τ_j , the time scale underlying the autocorrelation calculation becomes increasingly coarser, and the total number of time entries to be processed increasingly smaller. To correct for the varying time scale of the autocorrelation calculation, one has finally to divide, at each lag time τ_j , the calculated autocorrelation value by the corresponding time scale factor $2^{\lfloor j/B \rfloor}$. As pointed out in Ref. [12], this time coarsening leads to a triangularly weighted average of the true value of the autocorrelation function.

2.6 Generation of cross-correlation data

The above algorithm can easily be generalized to more complex situations. In most photon detection experiments, it is desirable to detect the fluorescence within two detection channels, either for obtaining more information about the fluorescence (detection of fluorescence polarization or different emission colours) [13], or to eliminate the adverse effects of detector afterpulsing [14] on the short-time behaviour of the FCS curve. In all these cases, cross-correlating the signals of both detection channels instead of autocorrelating the signals of each channel becomes an important task. This is easily realized with the above algorithm by using the time values of the first channel as the entries for the first vector $\{t_1, t_2, \dots, t_N\}$ and the time values of the second channel for calculating the lag-time shifted vector $\{t'_1, t'_2, \dots, t'_N\}$. In that case, one obtains the cross-correlation of the second channel with positive lag times against the first channel. By reversing the order, i.e., assigning the time values of the second channel to $\{t_1, t_2, \dots, t_N\}$ and using the time values of the first channel for $\{t'_1, t'_2, \dots, t'_N\}$, one obtains the cross-correlation of the first channel with positive lag times against the second channel.

2.7 Combination with time-correlated single-photon counting

An even more challenging task is the combination of the fluorescence lifetime-data that are recorded in the TTTR mode with the FCS analysis. Two new methods have been reported recently. Firstly, by time-gating the fluorescence signal on the nanosecond timescale before performing the FCS analysis, the impact of non-fluorescent scatter background on the FCS can be reduced [4,7]. Secondly, by designing smart filters for processing the detected photons on the basis of their fluorescence lifetime tag, one can extract two (or more) independent FCS traces from the measured signal, allowing the simultaneous retrieval of the complete and independent FCS curves for two or more different dyes with distinct fluorescence lifetime

(TCSPC-FCS) [8]. For both applications, the algorithm presented here is easily adapted. In case of time-gating, the procedure is straightforward: all entries in the vector $\{t_1, t_2, \dots, t_N\}$ are eliminated where the lifetime tag of the corresponding photons lie outside the set time-gate. In case of applying a filter function to the recorded photons that acts on the lifetime tag, the procedure is similarly simple: the initial weight values w_j are set to the filter output instead to one, subsequently applying the FCS algorithm without any alteration. Thus, the presented calculation method is most general and allows arbitrarily complex correlation analysis of the TTTR data.

3. Computation time

The computational efficiency of the presented algorithm can theoretically be compared with that of the traditional bin-and-correlate approach.

3.1 Bin-and-correlate algorithm

Assuming a total measurement time T , and neglecting the computational load of transforming the time-tagged data into binned data, one has to perform roughly T/τ_k multiplications for calculating the autocorrelation at $\tau = \tau_k$ (measurement time over minimum possible bin width). Thus, the total number of multiplications is estimated as

$$L_{bin} \sim \sum_k \frac{T}{\tau_k} \quad (3)$$

where k runs over all lag times used.

3.2 Time-tag-to-correlation algorithm

It is more complicated to estimate the computational load for the algorithm presented in this paper, because it depends on the number of detected photons and their spacing in time. Let us assume a homogeneous photon detection rate (equally spaced photons), which is the worst possible case, because then the time coarsening and data reduction step has the smallest data reduction efficiency. If the complete number of detected photons is denoted by N , the average rate of photon detection events is N/T . On a time scale of 2^k , i.e., after k time-coarsening steps leading to a temporal resolution of 2^k (see above), the length of the vector $\{t_1, t_2, \dots, t_N\}$ will be approximately equal to the total number of time intervals at that temporal resolution times the probability to find at least one photon detection event during time interval of length 2^k . The number of time intervals of width 2^k equals the total measurement time T divided by 2^k . Assuming furthermore a Poissonian statistics of photon detection, the probability to find at least one photon within a time interval of length 2^k is given by $1 - \exp(-2^k N/T)$. Finally, assuming that the number of necessary multiplications for calculating the autocorrelation at a given lag time value is roughly proportional to the length of the vector $\{t_1, t_2, \dots, t_N\}$, one arrives at the following estimation of the total number of multiplications for the calculation of the complete autocorrelation curve

$$L_{tttr} \sim B \sum_k \frac{N}{2^k} [1 - \exp(-2^k N/T)] \quad (4)$$

where the integer k runs from zero up to $n_{\text{casc}} - 1$.

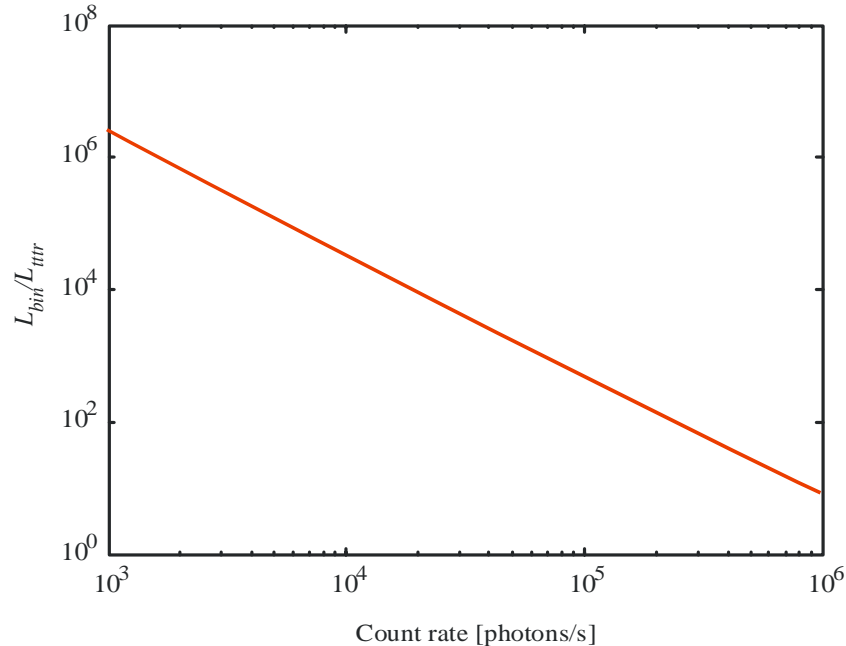


Fig. 1. Count-rate dependence of the estimated ratio L_{bin}/L_{ttr} of the computation times, Eqs.(3,4), for the conventional bin-and-correlate algorithm and the algorithm presented in this paper. The assumed minimum time resolution of the experiment is 100 ns, and the lag time vector is given by Eq.(2) with $B = 10$ and $n_{casc} = 17$.

3.3 Comparison of bin-and-correlate and time-tag-to-correlation algorithms

As an example, consider a vector of delay times as given by Eq. (2) with $B = 10$ and $n_{casc} = 17$, thus covering a time range of nearly six orders of magnitude (from $\tau_1 = 1$ up to $\tau_{170} = 1310710$). Let the smallest lag time of $\tau_1 = 1$ correspond to 100 ns in experimental time, a temporal resolution typical for TTTR photon counting measurements. Typical average photon count rates range between 1 kHz and 1 MHz. The result for the ratio of L_{bin} to L_{ttr} is shown in Fig. 1. The ratio L_{bin}/L_{ttr} does not depend on total measurement time T but only on the count rate N/T which can be seen when inserting Eq. (3) and Eq. (4) into L_{bin}/L_{ttr} and noticing that the final result contains only the ratio N/T but not T alone. As can be seen, the time-tag-to-correlation algorithm is superior to the bin-and-correlate approach even at high count rates. Moreover, one has to bear in mind that the made estimates are very conservative and biased in favour of the bin-and-correlate method: Any computational load of converting the tagged into binned data was neglected, and the rather unphysical assumption of evenly spaced photons was considered for the time-tag-to-correlation approach. In a real experiment, such as the detection of sparse numbers of molecules within a confocal detection volume, photon arrivals will be rather bunched into bursts every time a molecule diffuses through the detection region.

4. Experimental application

The experimental set-up is described in detail in Refs. [3,5]. Briefly, the light of a pulsed diode laser (PDL 800, PicoQuant) with 635 nm wavelength is focused, through a water immersion objective with 1.2 numerical aperture ($60 \times$ PlanApo IR, infinity corrected, Olympus), into the diluted bead solution ($\sim 10^{-9}$ M). The fluorescence light is collected through the same objective and, after passing a dichroic mirror (DRLP 650, Omega Optical), imaged onto a confocal aperture (50 μ m diameter). After splitting the transmitted light into

two detection channels by a 50/50-beam splitter, it is refocused onto the active area of two single-photon avalanche diodes (SPAD, SPCM-AQR 13, Perkin Elmer). The photon count signals of the SPADs are processed by a fast TCSPC electronics (TimeHarp 200, PicoQuant) and converted into a TTTR data stream of photon arrival and TCSPC times. The macroscopic temporal resolution of the TTTR data is 100 ns. The FCS algorithm was applied to data obtained from buffered aqueous solutions of fluorescently labelled latex beads (Crimson Red FluoSpheres® by Molecular Probes of ~36 nm diameter). These beads deliver a bright fluorescence signal and a well-defined slowly decaying autocorrelation curve due to their slow diffusion. Figure 2 shows the result of different autocorrelation algorithms applied to 10 seconds of a measurement at 200 μ W excitation power.

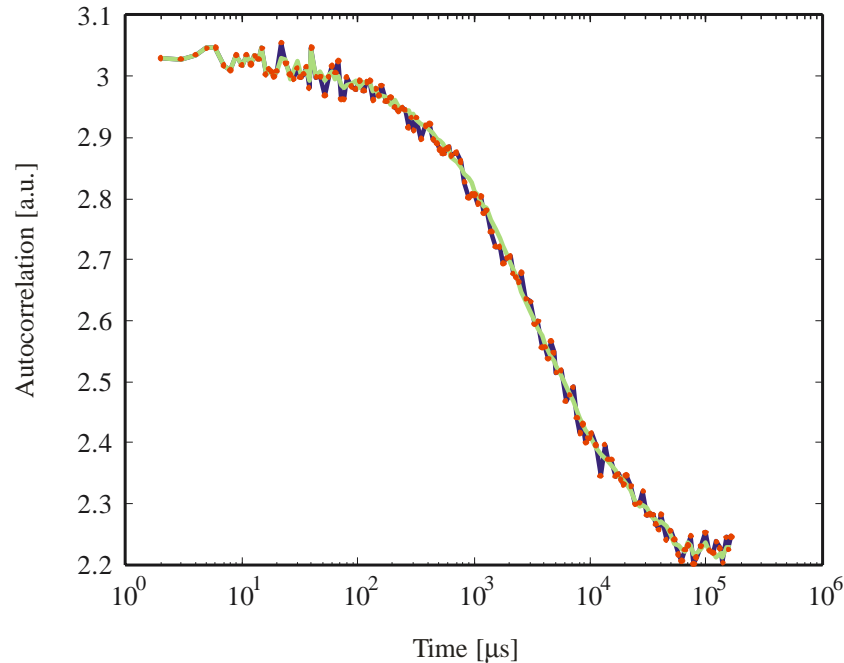


Fig. 2. Comparison of autocorrelation functions calculated with different algorithms. The blue line shows the result of the bin-and-correlate algorithm, red dots indicate the result of the time-tag-to-correlation algorithm without time-scale coarsening, and the light-green line is the result of the time-tag-to-correlation algorithm with time-scale coarsening.

Shown are the results of the conventional bin-and-correlate (blue line), and the time-tag-to-correlation algorithm *without* (red dots) and *with* time-scale coarsening (light-green line). The figure demonstrates that the bin-and-correlate and the time-tag-to-correlation algorithms yield identical results, as it should be. It also shows the effect of time-scale coarsening: The resulting autocorrelation curve is smoothed at larger time scales (at the lowest lag-time values, where no time-scale coarsening applies, all algorithms give of-course the same result).

For studying the computation time of the time-tag-to-correlate algorithm at different fluorescence intensities, measurements were carried out at six different excitation intensities of 0.7, 2, 7, 20, 70, and 200 μ W and lasted 60 minutes each. Thus, a large range of average photon count rates is covered. For FCS analysis, a vector of lag times as given by Eq. (2) with $B = 10$ and $n_{\text{casc}} = 20$ and τ_1 equal to 100 ns was used, and the signal of both detection channels was cross-correlated (which is usually necessary for preventing any distortion of the final FCS curve by SPAD-afterpulsing). The algorithm was written in C, and it was compiled and executed on a standard PC with an AMD Athlon 1800+ processor running at 1.5 GHz.

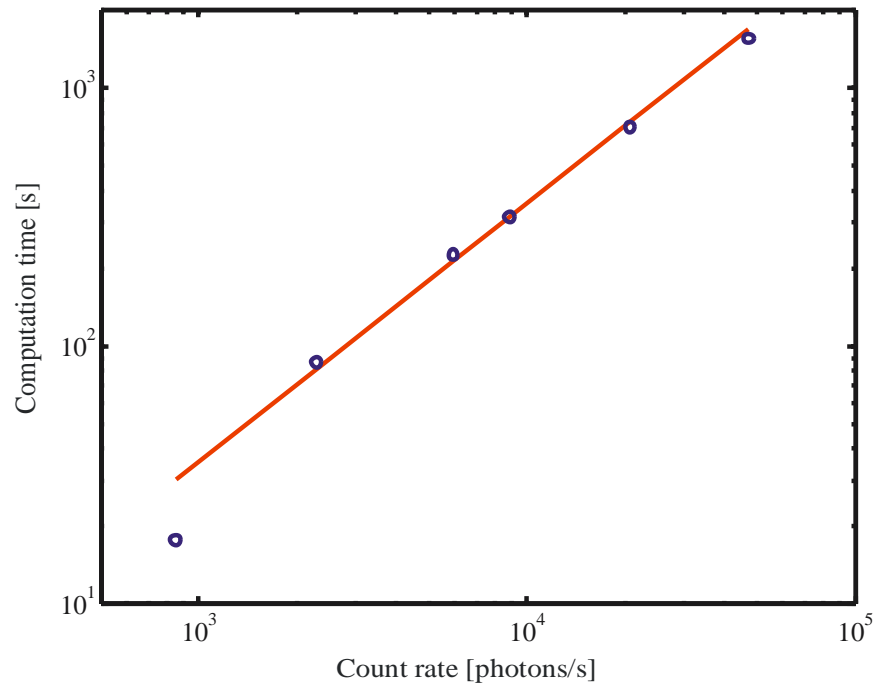


Fig. 3. Dependence of computation time on count rate. The blue dots correspond to six measurements at the different excitation powers as indicated in the text. The line is a unity slope fit to the last five data points (assuming a linear dependence of computation time on count rate).

The computation times for the six measurements are shown in Fig. 3. A linear curve was fitted against the determined computation-times with a standard least-squares fit. As can be seen, the computation time scales approximately linearly with the count rate (and thus number of detected photons), as expected from Eq. (4). Only at the lowest count rate (excitation power of $0.7 \mu\text{W}$) the computation time falls below the fitted line in Fig. 3. This is due to the fact that for very sparse data, the time coarsening steps will lead to a much faster data reduction than at larger fluorescence intensities. Thus, the algorithm is most efficient at low light intensities typical for single molecule spectroscopy.

5. Conclusion

We have presented a powerful and complex algorithm for converting asynchronous photon-counting data into correlation curves. The algorithm is perfectly suited to low-intensity measurements as usually encountered in single-molecule and most fluorescence-correlation experiments. The computational time of the algorithm scales linearly with the number of detected photons but is completely independent on the total measurement time. A major advantage of the presented algorithm is its flexibility, allowing for, besides standard autocorrelation, the computation of more complex correlation function such as cross-correlation, time-gated correlation, or sophisticated filtered correlation like TCSPC-FCS.

Acknowledgments

We thank U. B. Kaupp (IBI 1, Forschungszentrum Jülich) for his generous support of our research. Financial support by the Deutsche Forschungsgemeinschaft (DFG) and the Deutsche Volkswagenstiftung is gratefully appreciated.