GPU Acceleration of non-iterative and iterative algorithms in Fluorescence Lifetime Imaging Microscopy

Gang Wu,[1,2] Thomas Nowotny,[1] Yu Chen,[3] and David Day-Uei Li[2]

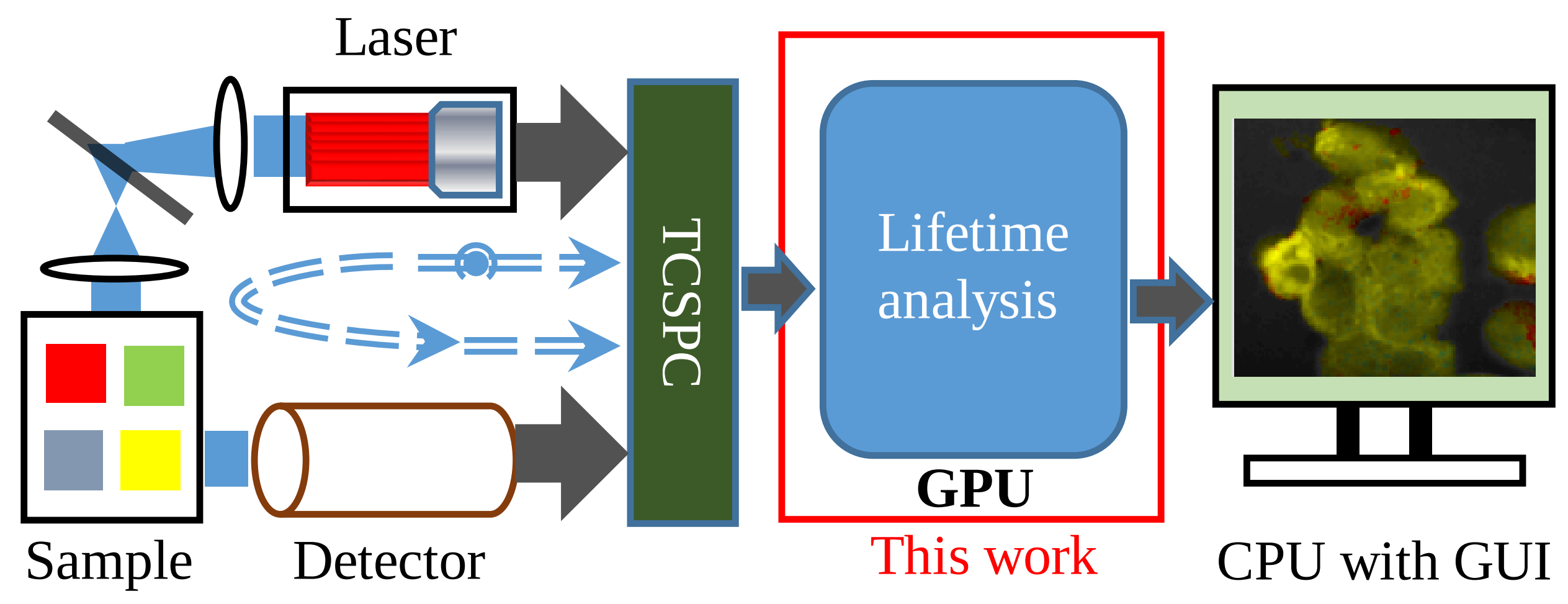
[1]University of Sussex, School of Engineering and Informatics, Brighton, UK
[2]University of Strathclyde, Centre for Biophotonics, Glasgow, UK
[3]University of Strathclyde, Physics, Glasgow, UK

1. Summary

Graphics Processing Unit (GPU) enhanced Fluorescence Lifetime Imaging Microscopy (FLIM) algorithms are presented, and their results are compared with the latest research results. The GPU based approaches are suitable for highly parallelized sensor systems and promising for high-speed FLIM applications.

2. FLIM System

FLIM Analysis System is used to extract the lifetime of fluorescent samples in biological research and medical diagnosis. It contains a light source (e.g. laser), a photon detector, a time-correlated single-photon counting (TCSPC) camera, lifetime analysis software, and a PC with graphical user interface (GUI).



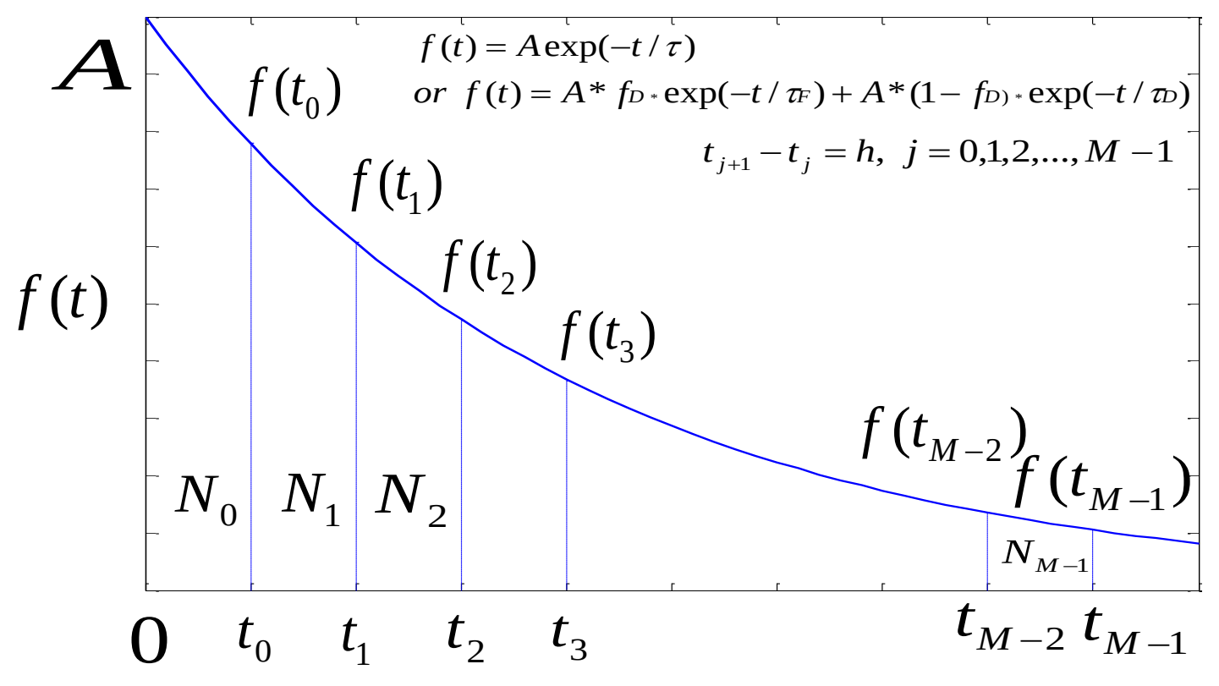
3. FLIM Algorithms

FLIM generates images by analyzing the exponential decay (fluorescence lifetime) of fluorescence intensity (from fluorescent proteins tagged on biological samples) at each camera pixel. Lifetime can be extracted from an exponential histogram, as shown in the figure, by the following algorithms.

A. Iterative algorithms

Algorithm	Function	
LSM[1]	$\chi^2=\sum_{j=0}^{M-1}(\frac{N_j-Y_j}{\sigma_i})^2$	$Y_j=\sum_{k=1}^{n}(A_k e^{-t_j/\tau_k})$
GA[2]	$\chi^2=\sum_{i=1}^{N_{GA}}\sum_{j=0}^{M-1}(\frac{N_{i,j}-Y_j}{\sigma_{i,j}})^2$	

n is the number of lifetime components, $N_{i,j}$ is the photon number of the *j*th bin of the *i*th pixel and NGA is the number of pixels in the same segment for GA.



B. Non-iterative algorithms

IEM [3]	CMM [4]	PM [5]	BCMM$_1$ [6]	BCMM$_2$ (τ_D unknown) [6]
$\tau_{IEM}=\frac{h\sum_{j=0}^{M-1}(C_jN_j)}{N_0-N_{M-1}}$	$\tau_{CMM}=(\frac{\sum_{j=0}^{M-1}(jN_j)}{N_c}+\frac{1}{2})h$	$\tau_F=\frac{1-u-v\tau_D\omega}{\omega(v-u\tau_D\omega)}$ $\tau_{Avg}=f_D\tau_F+(1-f_D)\tau_D$	$\tau_F=\frac{\tau_DN-X}{\tau_DK-N}$ $\tau_{Avg}=\frac{Nh}{N_0}$	$\tau_F=0.5[G-\sqrt{G^2-4(NG-X)/K}]$ $\tau_{Avg}=\frac{Nh}{N_0}$

M is the number of time bins, N_j and t_j are the photon number and the delay time of the *j*th bin, respectively, N_0 is the count number of the first time bin, *h* is the width of the time bin, C_j is the coefficient of Simpson's rule, and τ_D is the lifetime of the donor.

$$u=\int_0^T f(t)\times\cos(wt)dt/\int_0^T f(t)dt,\ v=\int_0^T f(t)\times\sin(wt)dt/\int_0^T f(t)dt,\ f(t)=\sum_{i=1}^{n}(A_ie^{-t/\tau_i})$$

$$f_D=[\tau_D(1+\tau_F^2\omega^2)(1-u-u\tau_D^2\omega^2)]/[(\tau_D-\tau_F)(1-u-u\tau_F^2\omega^2-\tau_F\tau_D\omega^2-u\tau_D^2\omega^2-u\tau_F^2\tau_D^2\omega^4)]$$

$$N=\sum_{j=0}^{M-1}(C_jN_j),\ X=\sum_{j=0}^{M-1}(C_jt_jN_j),\ K=N_0/h,\ G=\frac{KY-NX}{KX-N^2},\ Y=\sum_{j=0}^{M-1}(C_j\frac{t_j^2}{2}N_j)$$

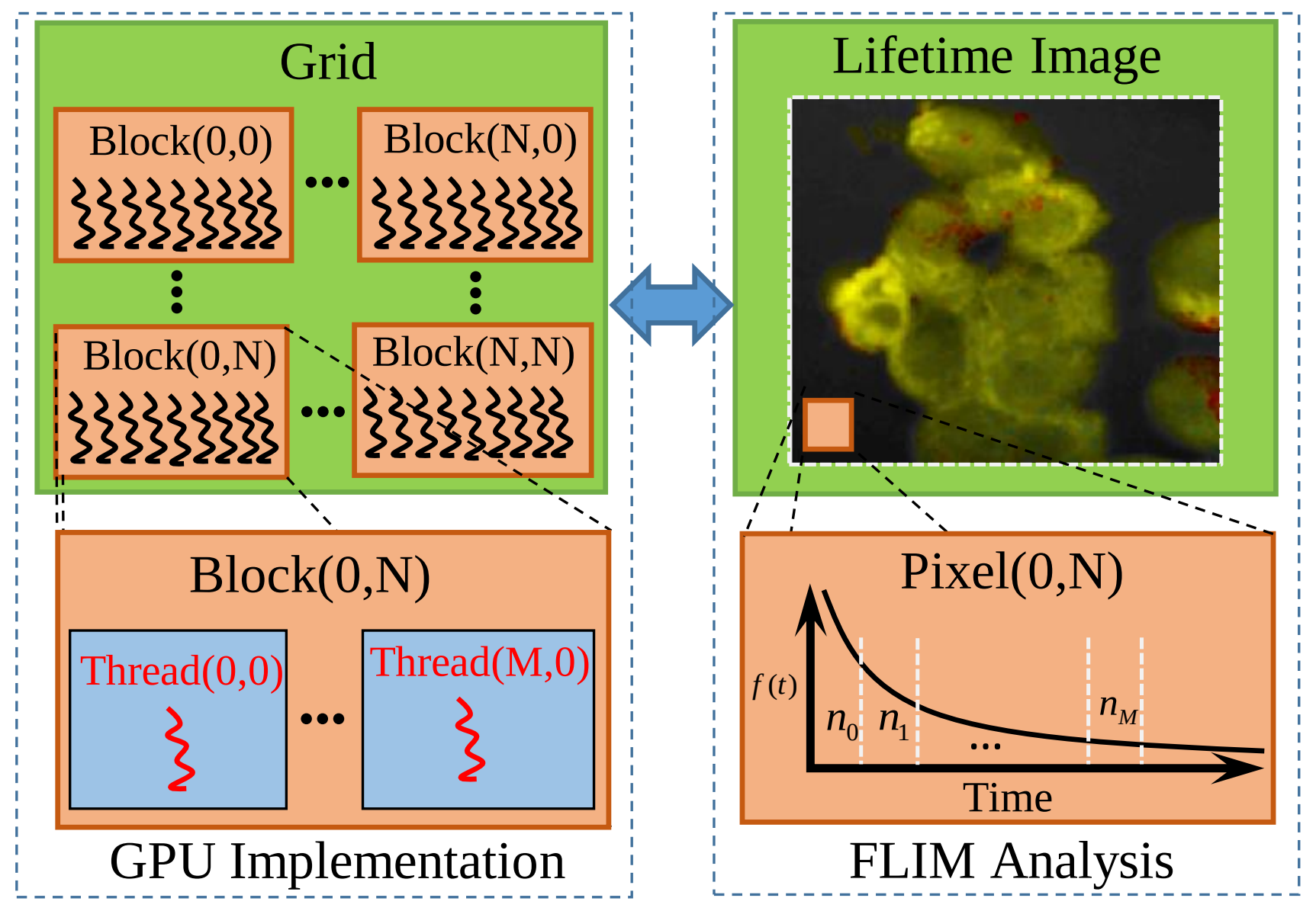
6. Conclusion

In this project, we have proposed a flexible and reliable processing strategy for FLIM analysis using GPU acceleration, which can replace CPU-only solutions, allowing considerable speed improvements without loss of quality. The performance of the tool has been verified with synthesized and experimental data, demonstrating substantial potential for GPU acceleration in rapid FLIM analysis.

4. GPU Implementation

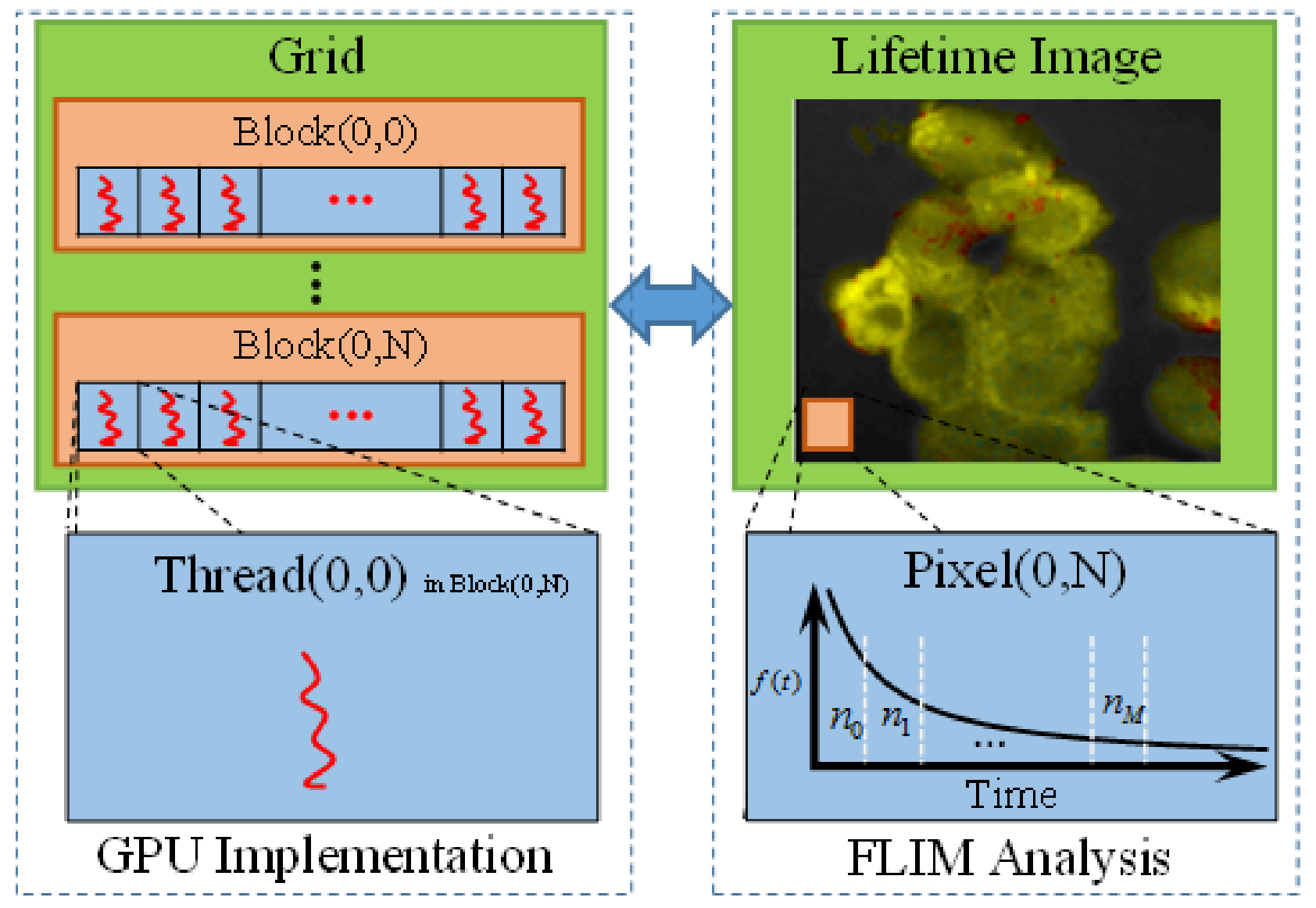
A. Block-based iterative algorithms

To realize parallel FLIM analysis in a GPU, the histogram for each pixel is analyzed by a separate block of CUDA, as shown in the Figure and each such block contains 256 threads that roughly correspond to the 256 time bins.



B. Thread-based non-iterative algorithms

The histogram of each pixel is analyzed by an independent CUDA thread, as shown in figure below, and each block contains 512 threads. This configuration allows analyzing a large number of pixels simultaneously, the exact number being determined by the number of streaming multi-processors



7. References

[1] A. A. Istratov et al., Rev. Sci. Instrum. 70(2), 1233-1257 (1999).

[2] P. J. Verveer et al., Biophys. J. 78(4), 2127–2137 (2000).

[3] D. Li et al, J. Opt. Soc. Am. 25(5), 1190-1198 (2008).

[4] D. Li, et al., J. Opt. Soc. Am. 26(4), 804-814 (2009).

[5] A. Leray, S. Padilla, et al., PLoS ONE. 8(7), e69335 (2013).

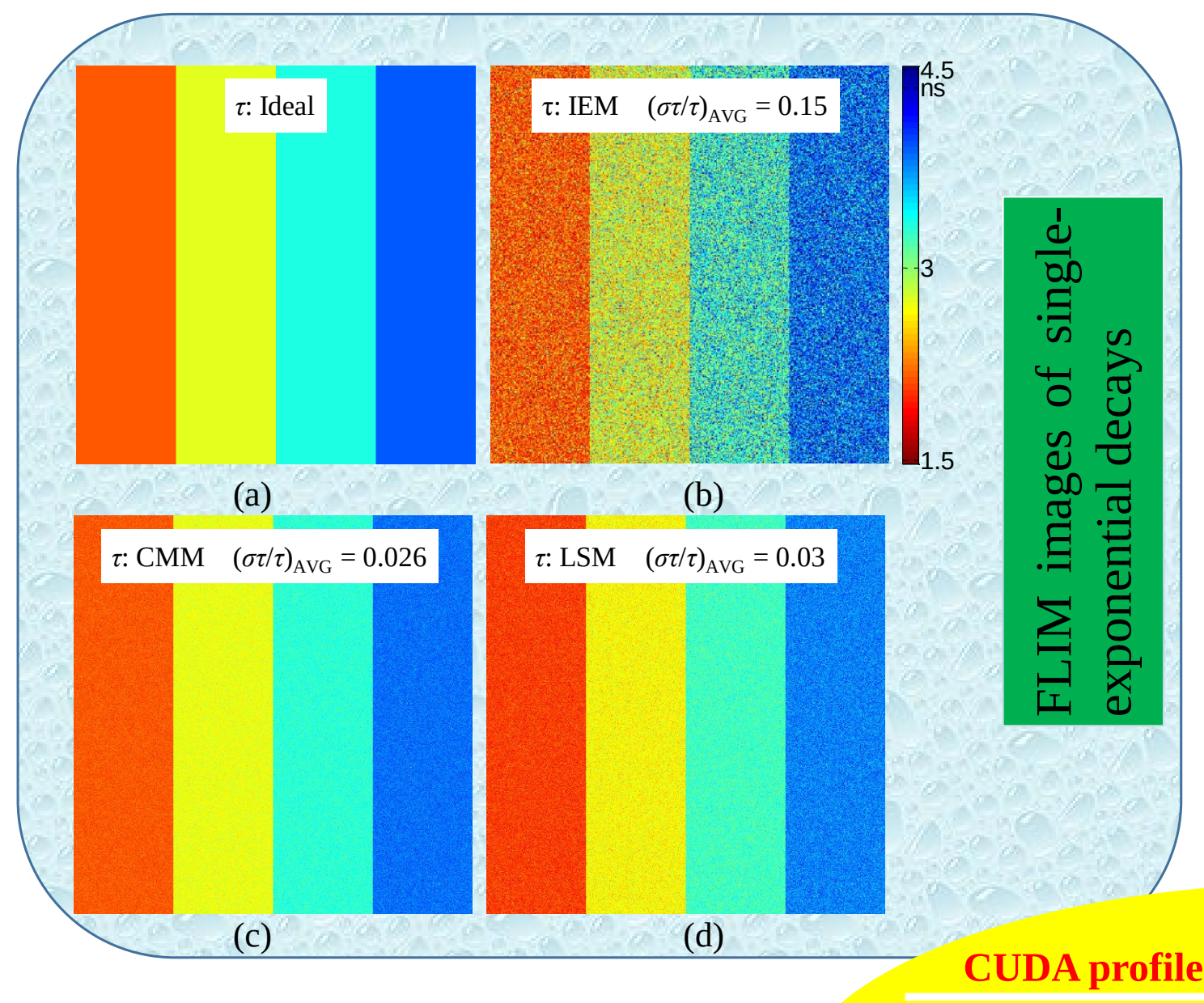
[6] D. Li, H. Yu, and Y. Chen, Opt. Lett. 40(3), 336-339 (2015).

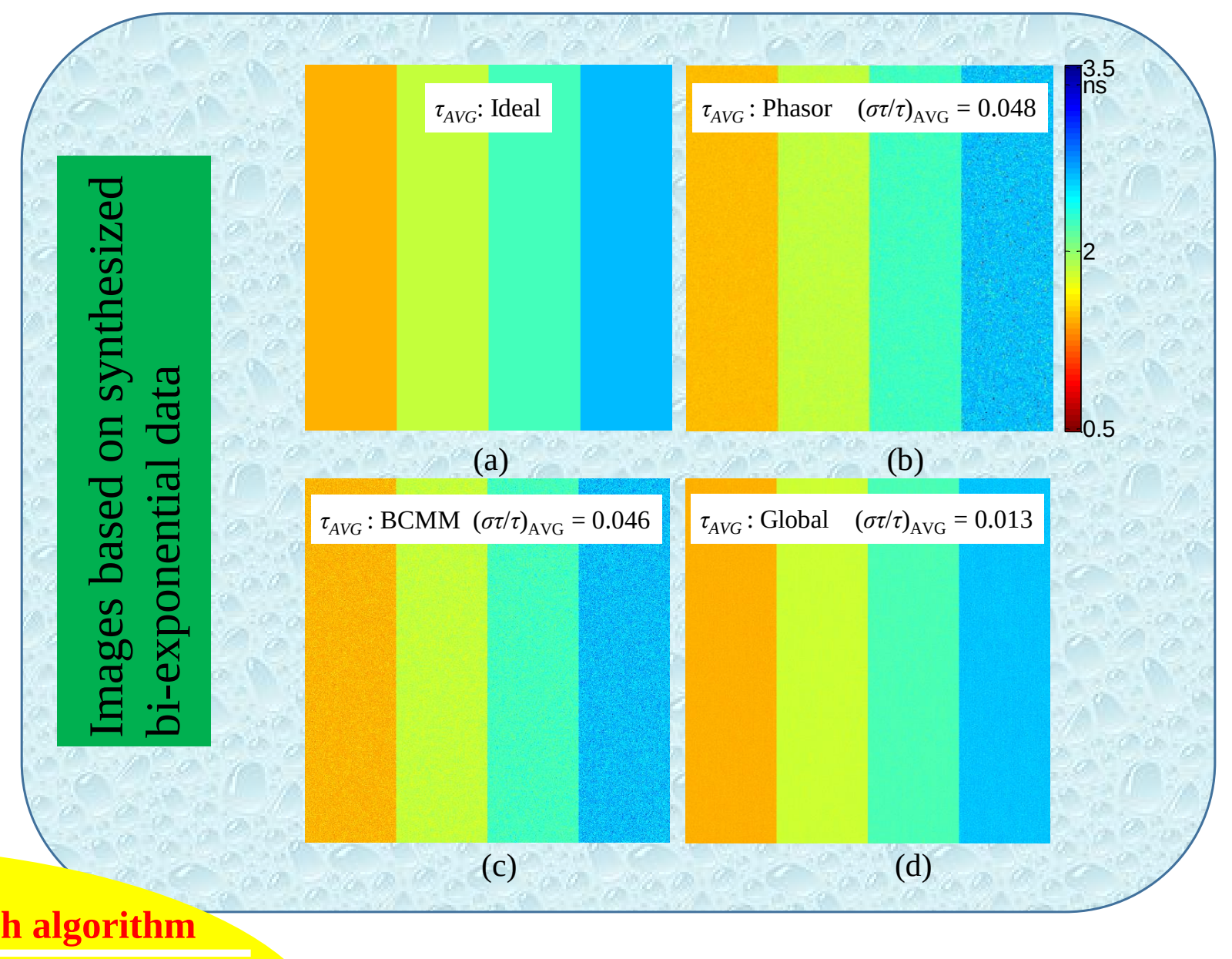
[7] Y. Zhang et al., Faraday Discuss., doi: 10.1039/C4FD00199K (2014).

5. Simulations and FLIM data analysis

A. Simulation

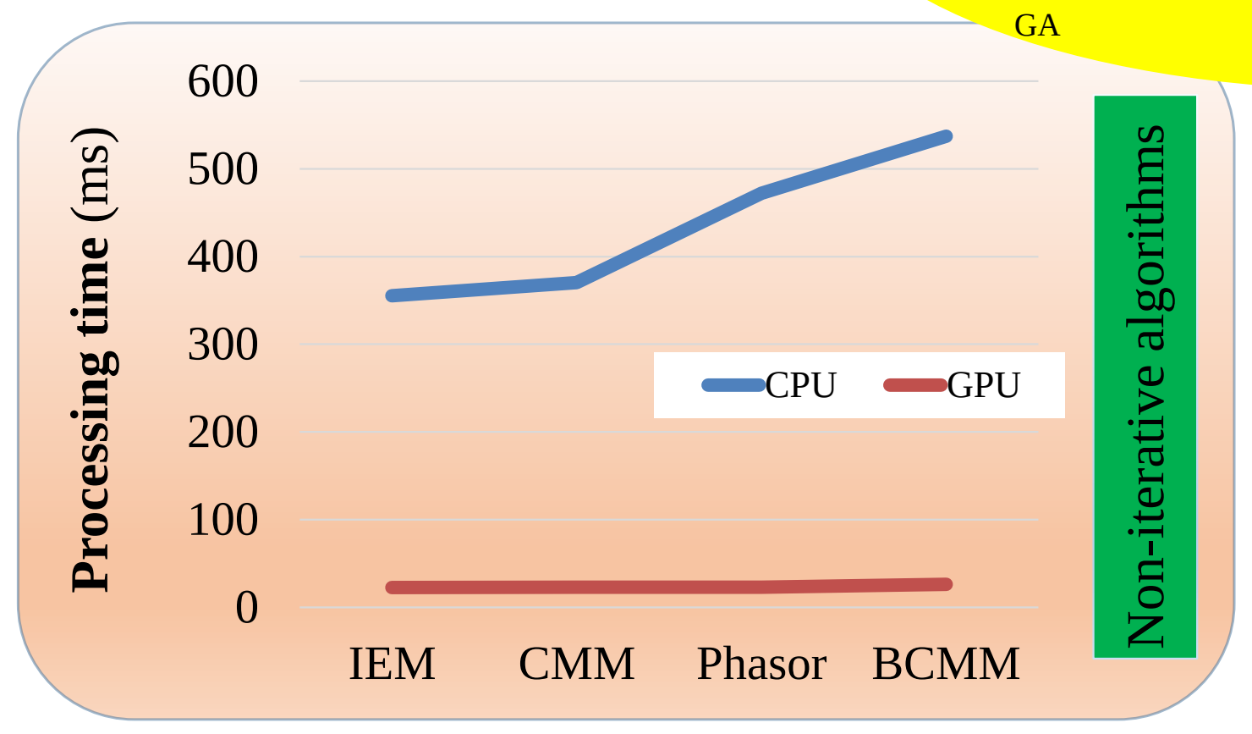
2000 photons were collected, the number of time bin was 256, the width of each time bin was 100ps and the size of the image was 512 by 512 pixels.

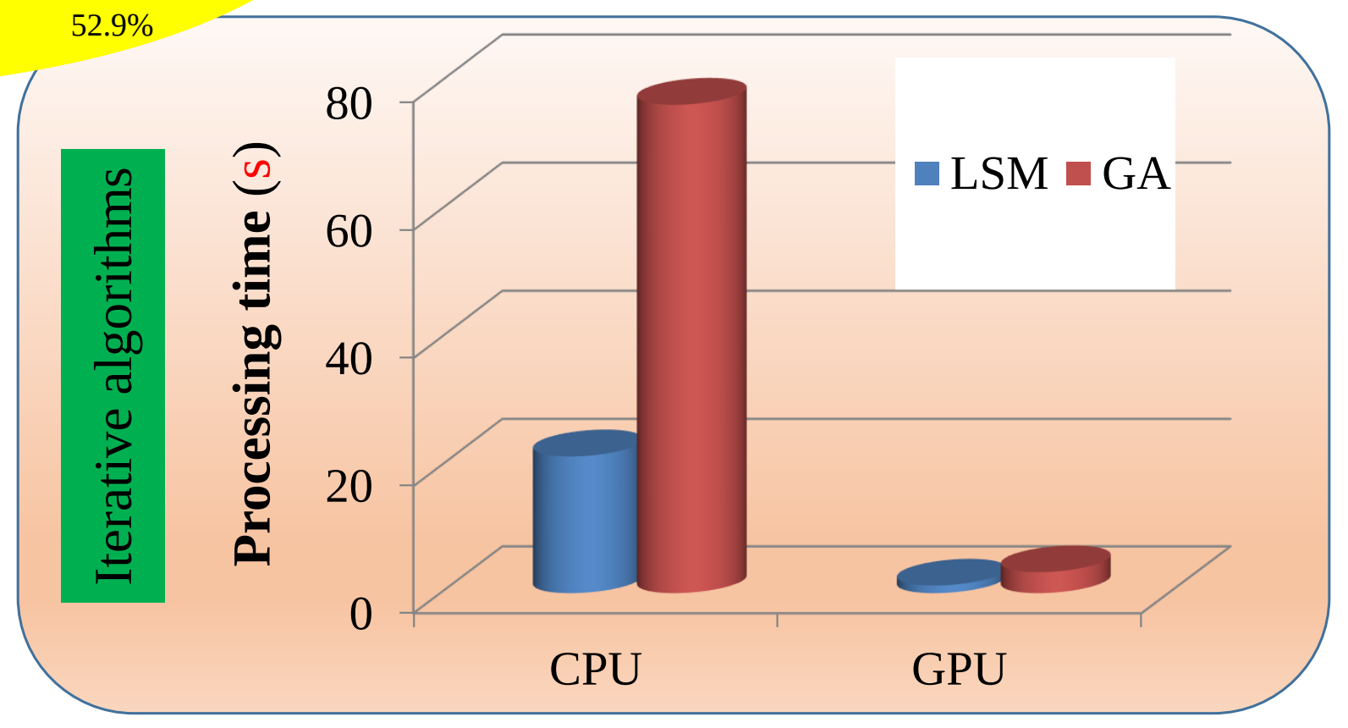




CUDA profiler features for each algorithm

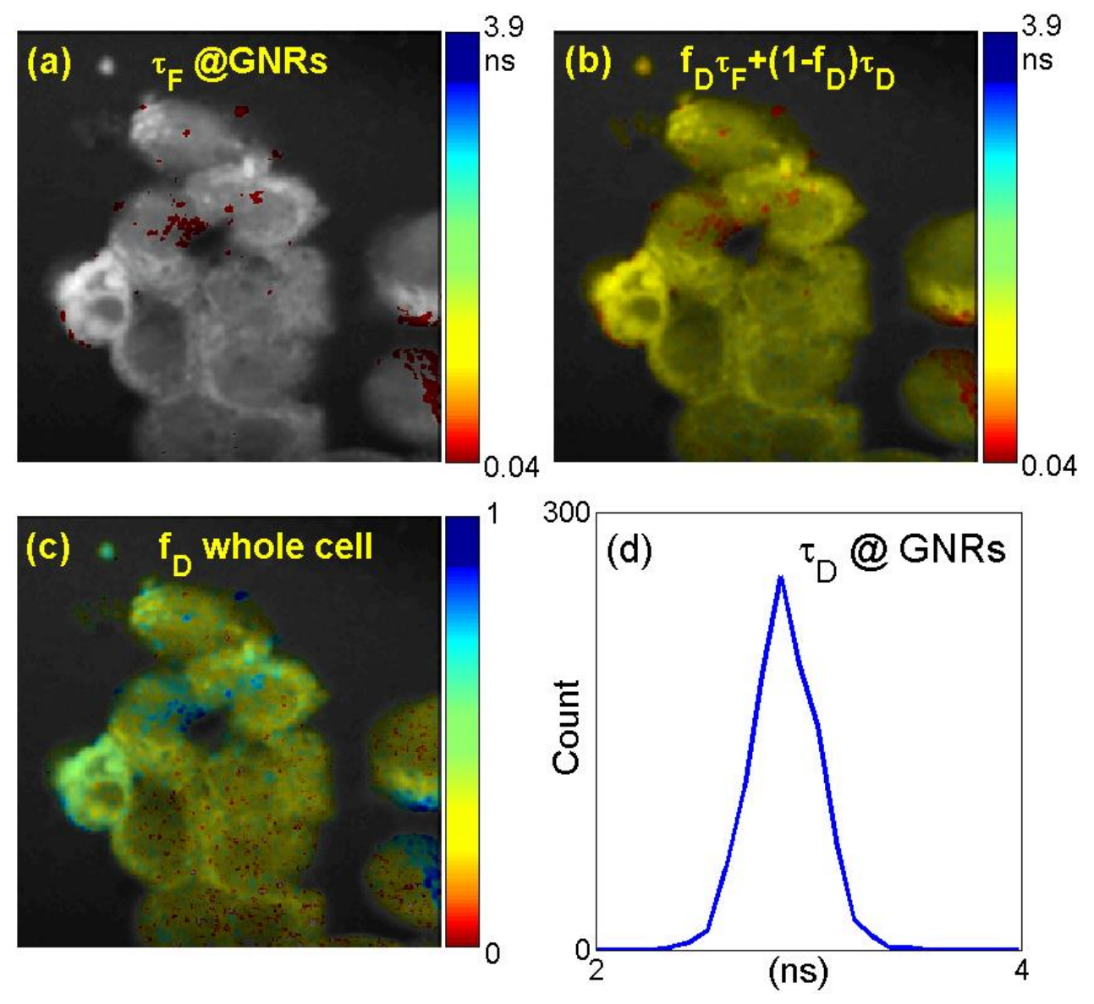
Algorithm	Transfer (ms)	Computation (ms)	Occupancy
IEM	18.7	3.7	99.0%
CMM		4.7	96.0%
Phasor		4.1	99.0%
BCMM		7.6	99.2%
LSM		1151.5	54.1%
GA		3295.1	52.9%





B. Experiment

We demonstrate the performances of the GPU based BCMM on two-photon FLIM images of gold nanorods (GNRs)-Cy5 labelled A375 cells. GNRs were conjugated with Cy5 labelled oligonucleotide through a procedure described elsewhere [7]. The A375 cells were incubated with nanoprobes (GNR-Cy5) and fixed with paraformaldehyde. FLIM was performed using a confocal microscope (LSM 510, Carl Zeiss) equipped with a time-correlated single photon counting (TCSPC) module (SPC-830, Becker & Hickl GmbH).



Experiment configurations

Image Size	Bin Number	Bin Width (ps)	Laser Pulse (MHz)	τD (ns)	τF (ns)
256x256	256	39	80	~ 2.93 +/- 0.16	~ 0.1

Experiment results with CPU-OpenMP and GPU

Target	Algorithm	CPU (ms)	GPU (ms)	Speedup (times)
τ_F	BCMM$_1$ (τ_D fixed)	111.24	5.4	20.6
	BCMM$_2$ (τ_D unknown)	169.3	9.3	18.2

C. Discussion

FLIM analysis is well-suited for GPU acceleration because it is highly parallelizable. Each pixel in a FLIM frame can be processed independently of any other pixel, and, depending on the details of the algorithm, there is a lot of room for parallelization even within the processing of an individual pixel.