Journal of Biomedical Optics

BiomedicalOptics.SPIEDigitalLibrary.org

Graphics processing unit-based quantitative second-harmonic generation imaging

Mohammad Mahfuzul Kabir
ASM Jonayat
Sanjay Patel
Kimani C. Toussaint, Jr.

SPIE.

Graphics processing unit-based quantitative second-harmonic generation imaging

Mohammad Mahfuzul Kabir,[a] ASM Jonayat,[b] Sanjay Patel,[c] and Kimani C. Toussaint Jr.[a,b,d,*]
[a]University of Illinois at Urbana-Champaign, Department of Mechanical Science and Engineering, Laboratory for Photonics Research of Bio/nano Environments (PROBE), Urbana, Illinois 61801, United States
[b]University of Illinois at Urbana-Champaign, Department of Mechanical Science and Engineering, Urbana, Illinois 61801, United States
[c]University of Illinois at Urbana-Champaign, Department of Electrical and Computer Engineering, Urbana, Illinois 61801, United States
[d]University of Illinois at Urbana-Champaign, Affiliate in the Department of Electrical and Computer Engineering, and Bioengineering, Urbana, Illinois 61801, United States

Abstract. We adapt a graphics processing unit (GPU) to dynamic quantitative second-harmonic generation imaging. We demonstrate the temporal advantage of the GPU-based approach by computing the number of frames analyzed per second from SHG image videos showing varying fiber orientations. In comparison to our previously reported CPU-based approach, our GPU-based image analysis results in ~10× improvement in computational time. This work can be adapted to other quantitative, nonlinear imaging techniques and provides a significant step toward obtaining quantitative information from fast *in vivo* biological processes. © *2014 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JBO.19.9.096009]

Keywords: second-harmonic generation; image analysis; graphics processing unit; quantitative imaging.

Paper 140352R received Jun. 4, 2014; revised manuscript received Aug. 7, 2014; accepted for publication Aug. 15, 2014; published online Sep. 15, 2014.

1 Introduction

Second-harmonic generation (SHG) imaging, based on the second-order nonlinear optical process, in which a noncentrosymmetric material (i.e., a material showing no center of inversion symmetry) converts a portion of incident light to scattered light at exactly twice the frequency, has developed into an important nonlinear imaging technique over the past several years. Among its benefits is its inherent ability to produce volumetric images of biological tissues comprising collagen fibers[1] without the need for labeling with an exogenous contrast agent. We have previously demonstrated quantitative SHG (Q-SHG) imaging as an effective and accurate modality to ascertain quantitative information from such collagen-based biological tissues.[2–6] For example, with respect to collagen fiber organization, the preferred orientation and orientation anisotropy have provided significant information.[2] Indeed, application of quantitative SHG has resulted in assessment of microstructural information of nonpregnant rat cervical tissue, healthy from injured horse tendons,[7] age-related changes in porcine cortical bone,[8] and dissimilarities in stromal collagen fiber organization in human breast biopsy tissues at various pathological stages.[5] Additionally, Q-SHG imaging has also been reported as a suitable method for quantifying the change in dermal collagen fibers in skin burns using a rat skin burn model.[9] In spite of these advancements, full applicability of Q-SHG imaging has been confined to static imaging conditions, thereby leaving its utility for dynamic biological processes largely unexplored. To address this, we recently reported on the experimental and computational requirements for carrying out Q-SHG imaging under dynamic conditions,[10] i.e., simultaneously computing and displaying quantitative information with image acquisition. We found that for a 512×512-pixel area, the preferred orientation of collagen fibers in a tissue specimen captured by SHG imaging can be computed within ~950 ms using a standard multicore CPU.

Recently, graphics processing units (GPUs) have emerged as alternate computation devices for faster processing compared to standard CPUs. GPUs comprise several thousand times more processing units or cores compared to conventional computer CPUs, permitting all cores to be used to carry out the same desired instructions in parallel. This facilitates its usage in various computational imaging applications where processing time is expensive. For example, GPU-based algorithms have been used to develop spectral (Fourier) domain optical coherence tomography[11–14] techniques with a significant reduction in computation time when compared to standard CPU-based algorithms. Similar results have also been observed from using the GPU for reconstruction of x-ray computed tomography images of high contrast and precision[15] as well as performing deconvolution of three-dimensional confocal microscopic images.[16] Additionally, GPUs have also been used to accelerate and optimize Monte Carlo simulations,[14,17] typically used to study the theory of light transport through various media. As such, the GPU would be extremely useful in obtaining quantitative information at the time scales of some of the faster biological processes, such as the propagation of an action potential in neurons occurring on the order of milliseconds, which has been successfully captured with SHG.[18] The approach could also be useful for situations where analysis of multiple, quantitative metrics are incorporated with simultaneous image acquisition. In this work, we incorporate an NVIDIA GPU to our image analysis system, enabling parallel processing of our dynamic SHG image analysis algorithm. As proof-of-concept, we present several synthetic experiments in which we apply our GPU-based approach to quantitatively analyze consecutive frames from several videos

*Address all correspondence to: Kimani C. Toussaint, Jr., E-mail: ktoussai@illinois.edu

of 512 × 512-pixel SHG images. In general, the videos are of varying arrangements of collagen fiber organization. We compare the computation time obtained using GPU versus CPU. The paper is organized as follows. Section 2 describes the experimental methods and image analysis technique used. Section 3 presents the results and discussion, while Sec. 4 provides the conclusion.

2 Methods

2.1 *Image Analysis*

We have previously provided a detailed description of our quantitative SHG image analysis carried out under dynamic conditions, which can be found elsewhere.[10] Briefly, after acquiring a 512 × 512 image, a Gaussian filter is applied and the image is subsequently divided into a 16 × 16 grid, with each grid containing 32 × 32 pixels. For each grid, a preferred orientation is calculated based on the computed intensity gradient for each pixel within a grid. This information can then be used to estimate a global preferred orientation for the whole image. The accuracy of the calculated orientations is estimated by the circular variance,[19–22] a detailed description of which is provided in the Appendix. An intensity threshold is set to discriminate the background from the signal in a manner analogous to what we have previously reported.[2] This same threshold is used for calculations that are done both within a grid and for the global orientation estimate. Finally, an image is displayed with a gridded overlay, with arrows indicating preferred fiber orientation within each grid and the computed average orientation and circular variance being presented.

MATLAB® coupled with the compute unified device architecture (CUDA) parallel computing platform was used to develop the code. The parallel instructions were written in the C programming language, using the NVIDIA CUDA library version 5.0, and implemented in the GPU while MATLAB® was used as the host function to acquire the image, transfer image data to and from the GPU, and subsequently display the results. The hardware used for implementation consisted of an NVIDIA GTX 590 GPU, running on a Windows 7, Core i7-2600K Quad Core CPU running at 3.40 GHz clock speed, 3.8 GHz of maximum Turbo frequency, and 24 GB of DDR3-1066/1033 RAM. This same computer was used for the comparisons where GPU-based calculations were compared with CPU-based ones. A description of the GPU architecture and the CUDA programming model can be found in the CUDA Programming Guide 4.0.[23] To facilitate its adaptation in the GPU architecture, the image analysis procedure was divided into three segments known as CUDA kernels as shown in Fig. 1. In the first kernel, the acquired 512 × 512 image is divided into a 43 × 43 image grid, each of which is 12 × 12 pixels in size. For this kernel, the GPU grid contains 43 × 43 threadblocks, where each threadblock contains 16 × 16 threads. Each threadblock is assigned to apply a Gaussian filter over one image grid. As the pixels in the boundary of a grid require contributions from neighboring pixels to apply the Gaussian filter, the threadblocks contain one extra thread in each boundary. In the second kernel, the filtered image is divided into a 16 × 16 grid of 32 × 32 pixels each, where the preferred orientation is calculated for each individual grid. In both these two kernels, individual GPU threadblocks are assigned to individual image grids, where individual pixels inside the grids are computed upon by individual threads in the threadblock; hence, parallel operation of data points is achieved on two separate levels.

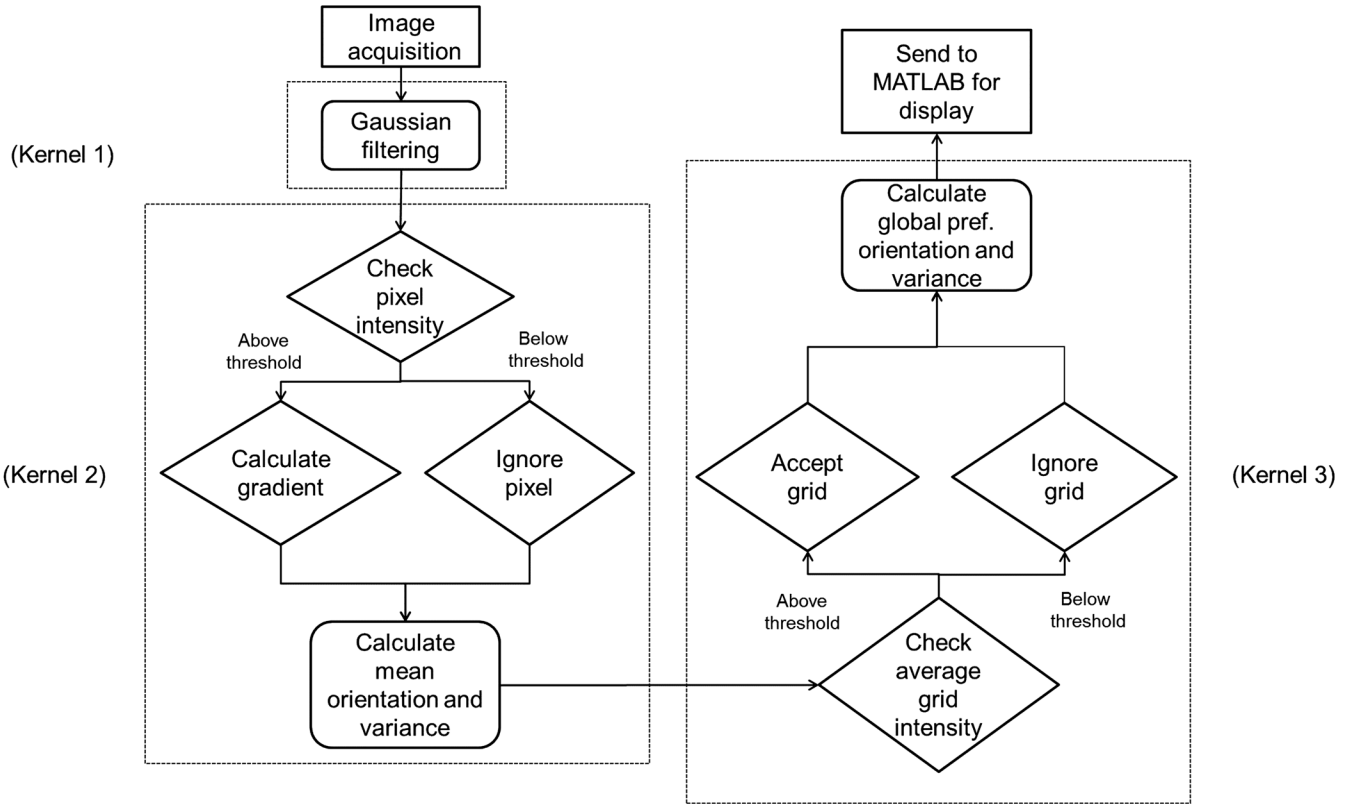

Fig. 1 Flow chart of the steps performed in the graphics processing unit (GPU)-based quantitative image analysis modality.

On the first level, all the threadblocks in the GPU multiprocessors carry out their computation in parallel, while on the second level, the individual threads inside a threadblock also operate in parallel. This essentially means that all pixels in a grid and all grids in the image are computed simultaneously, thus significantly reducing computation time. Finally, in the third kernel, the preferred orientations from the individual grids are used to calculate a global preferred orientation. As there are only 256 (16×16) preferred orientation values, a single GPU threadblock of 256 threads is sufficient. After performing the processing in the GPU, the image along with the quantitative information is returned to the CPU. MATLAB® instructions are used to display the preferred orientation in each block of the 16×16 grid and a circular histogram showing the distribution of the preferred orientation values over the complete image. Finally, the global preferred orientation and the associated circular variance are also displayed.

2.2 *Experiment*

Both the GPU-based and the CPU-based codes are applied on consecutive frames of three videos comprising 512×512-pixel

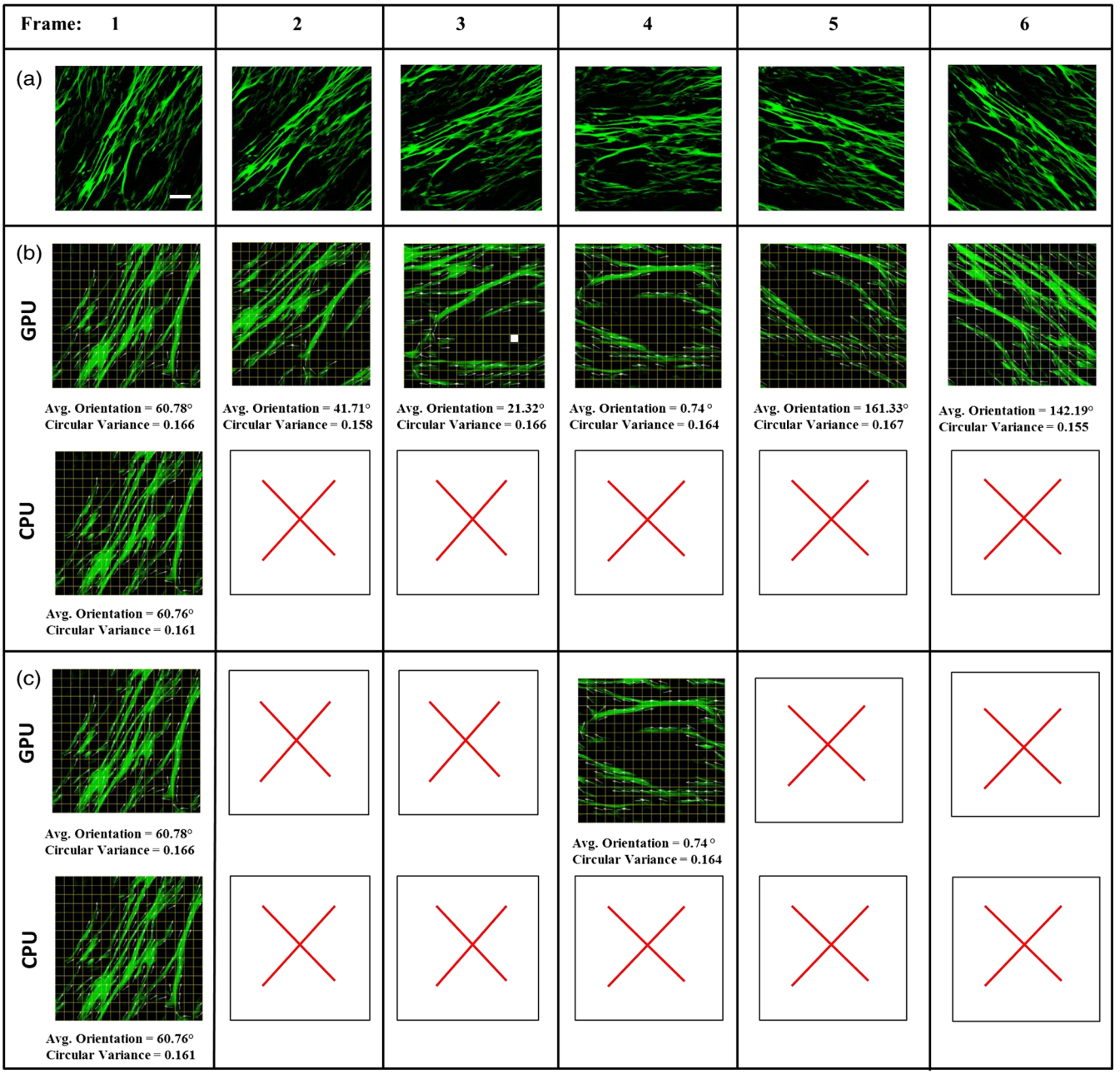

Fig. 2 Representative frames from a video of second-harmonic generation (SHG) images of breast biopsy tissues. (a) The first six consecutive frames in the video, showing fibers that are progressively rotated by 20 deg with respect to the horizontal. The frames captured and analyzed by the GPU-based and CPU-based image analysis are shown in (b) for 10 fps and in (c) for 29 fps. The scale bar corresponds to 10 μm for all images. These results are also compiled in the videos (Video 1 MPEG, 4.2 MB [URL: http://dx.doi.org/10.1117/1.JBO.19.9.096009.1] and Video 2 MPEG4, 7.3 MB [URL: http://dx.doi.org/10.1117/1.JBO.19.9.096009.2]).

SHG images. Information on the optical setup used to collect the SHG images can be found elsewhere.[7] In the first video, we have consecutive frames showing SHG images of breast biopsy tissue being rotated at increments of 20 deg (relative to the horizontal) between each frame. The frame rate of the video is 10 frames per second (fps) and contains 20 images running for a total duration of 2 s. The contents of the second video are the same as the first one, except that the frame rate of the video is set at 33 fps (note that this is approximately the standard video rate), and it contains 66 images running for 2 s. The third video is of SHG images of various collagen-based biological tissues, namely porcine tendon, rat cervix, and breast biopsy tissues. The frame rate is set at 10 fps, while the number of images and total run time is set at 20 images and 2 s, respectively.

3 Results and Discussion

Figure 2 depicts the representative frames from the first video, as well as the results obtained by operating the two different computational modalities on it. Figure 2(a) shows the first six frames of the video, while Fig. 2(b) shows the frames that are captured and analyzed by the GPU-based and CPU-based codes in two consecutive rows, respectively. It is clear from Fig. 2(b) that the GPU-based code successfully captures and analyzes all six consecutive frames of the video. For the given video frame rate (10 fps), this is consistent with the expected computation time of ~100 ms for each image. In comparison to this result, the CPU-based code only acquires the first frame and fails to capture any of the subsequent frames. From Video 1, it is also observed that the CPU is successful in capturing the first and the tenth frames of the video. This observation supports our previously reported fact that the computation time for a CPU-based code is ~950 ms for a 512×512-pixel image,[10] indicating that at 10 fps, it would fail to capture the next eight consecutive frames. The same analytical steps were carried out for the second sample video, the results of which are depicted in Fig. 2(c). The two consecutive rows in Fig. 2(c) show the frames that are captured by the GPU-based and CPU-based modalities, respectively. It is observed from Fig. 2(c) that for the new video frame rate of 33 fps, the GPU fails to capture two out of every three frames of the video. It is also observed from the second row in Fig. 2(c) that the CPU-based code could only analyze the first frame in this case, too, and fails to capture any subsequent frames shown in this figure. Video 2 reveals that the next frame successfully captured by the CPU is frame 32.

Figure 3 shows the results obtained from analyzing consecutive frames of a video comprising SHG images of a variety of collagen-based tissues. Here, the goal is to evaluate the performance of the GPU-based code when SHG images vary (in fiber density, orientation, and organization) greatly between each consecutive image. Again, we observe in Fig. 3(b) that the GPU-based code captures and analyzes all consecutive frames from the video irrespective of fiber orientation or density, while in Fig. 3(c) and Video 3, we see that the CPU-based code could only perform its analysis on the first and the eleventh frames. Thus, the processing time for the GPU-based approach is not influenced by fiber density and spatial organization.

Figure 4 compares the performance in processing time between the GPU- and CPU-based approaches for each segment

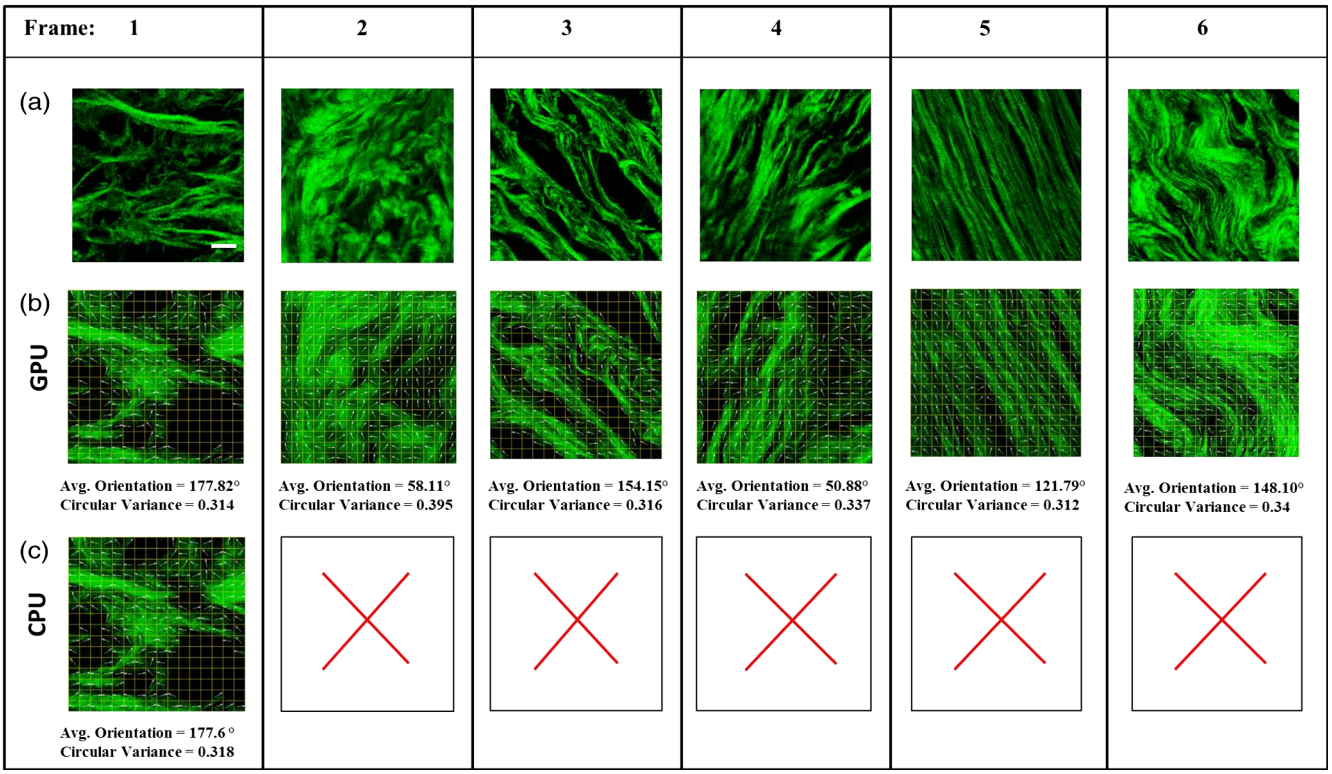

Fig. 3 Representative frames from a video of SHG images of several collagen-based tissues (a). Frames 1, 3, and 4 are SHG images of human breast biopsy tissues, while frames 2 and 5 are of rat cervix and porcine tendon tissues, respectively. (b) and (c) show the frames captured and analyzed by the GPU-based and CPU-based codes, respectively. These results are also compiled in a video (Video 3 MPEG4, 5.3 MB) [URL: http://dx.doi.org/10.1117/1.JBO.19.9.096009.3]. The scale bar corresponds to 10 μm for all images.

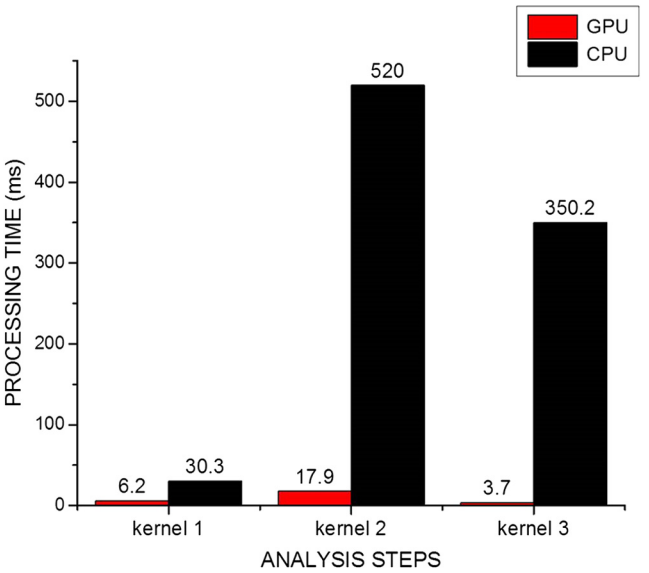

Fig. 4 Image processing time required for the GPU-based analysis compared with the CPU-based analysis for three different analysis steps.

of the analysis. To get an accurate estimate of the processing times, the two modalities are used to analyze 20 SHG images of size 512×512 pixels showing varying degrees of fiber organization and density. The processing times for each step for all the images are obtained and their average values are used in this comparison. It is observed from Fig. 4 that the calculation of the preferred orientation of individual image grids (kernel 2) takes the longest amount of time for both approaches. As such, to obtain a significant reduction in overall computation time, it would be important to reduce the time required by kernel 2. Our GPU-based implementation achieves a time improvement of ~20× for kernel 2, reducing it from ~520 ms used by the CPU to ~18 ms. Application of a Gaussian filter and calculation of the global preferred orientation were performed ~5× and ~35× faster, respectively. The time required to display the results (not shown) is the same for both the modalities and it was observed to be ~50 ms. Overall, the GPU-based code performs the analysis at an average of ~10× faster than the CPU-based code.

It is worth noting that although 512×512-pixel images were used for the proof-of-concept, the GPU-based code can also be used to analyze images of higher pixel density without any modification. The image grid size and the number of threadblocks in the GPU would scale according to the image size. Individual image grids and threadblocks would also contain the same number of pixels and threads, respectively. However, this would not be possible in the current version of the GPU due to limitations in the number of threadblocks and grids that can operate in parallel. In certain cases, the individual image grid sizes may need to be increased to facilitate clear visualization. This would require each thread in a threadblock to process more than one pixel. For example, if the image grid contains 64×64 pixels, a threadblock of 32×32 threads would assign one thread to analyze data obtained from four pixels. Note that this would require an increased amount of memory in the GPU, which is not supported in the current version of the GPU that was used in this work. Apart from this, if any further quantitative analysis is desired from the SHG images, additional CUDA kernels can be constructed and conveniently added to the existing code.

4 Conclusion

In this paper, we demonstrated GPU-based quantitative analysis of SHG images. We showed that the preferred orientation of collagen fibers can be determined in ~100 ms, either at the level of individual elements in a 16×16 grid or globally for a 512×512-pixel image. As proof-of-concept, we have applied this modality to analyze consecutive frames of two videos of 10 and 33 fps, respectively. In the first case, the GPU-based system successfully captured and analyzed all the frames of the video, while in the second case, it succeeded in capturing one in every three frames. In contrast, the same analysis using a standard CPU failed to capture all but the first frame from both the videos for the representative frames shown. Both approaches were again compared for a video of SHG images from more complex collagen-based structures, with the GPU-based method clearly outperforming the standard method. This improvement in processing time makes our approach attractive compared to other nonlinear imaging modalities that are modified for quantitative analysis.

Appendix: Calculating Preferred Orientation

The estimation of the preferred orientation of an image grid is carried out in the following steps:

1. In the first step, horizontal $[dI_{x(i,j)}]$ and vertical $[dI_{y(i,j)}]$ intensity gradients are calculated for each pixel. The method used is best demonstrated with the help of a 3×3-pixel image block, schematically displayed in Fig. 5. Based on the location of each pixel within the image block, a centered, forward or a backward difference method is used to calculate the intensity gradient in the horizontal (x axis) and the vertical (y axis) directions. The relevant equations are given in Table 1, where i and j refer to the x- and y-coordinates of the pixel location, respectively, while h represents each pixel width. As an example, the intensity gradient for the pixel located in (1,3) can be considered. In this case, the forward difference method is used to calculate the intensity gradient in the horizontal direction, utilizing the intensities of pixels 1 and 2—$I_{(1,3)}$ and $I_{(2,3)}$. Using a similar reasoning, the backward difference method is used to calculate the intensity gradient in the vertical direction by using the intensities of pixels located in (1,2) and (1,3). But in the case of the pixel located in (2,2), the centered difference method was used to calculate intensity

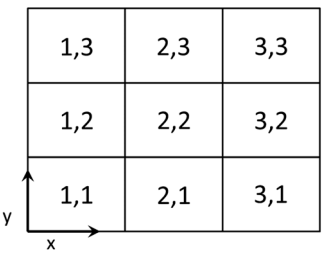

Fig. 5 Schematic diagram of a 3×3 pixel image block.

Table 1 Equations used for calculating intensity gradients of image pixels.

Gradients	Centered difference	Forward difference	Backward difference
$dI_{x(i,j)}$	$dI_x = [I_{(i+1,j)} - I_{(i-1,j)}]/2\ h$	$dI_x = [I_{(i+1,j)} - I_{(i,j)}]/h$	$dI_x = [I_{(i,j)} - I_{(i-1,j)}]/h$
$dI_{y(i,j)}$	$dI_y = [I_{(i,j+1)} - I_{(i,j-1)}]/2\ h$	$dI_y = [I_{(i,j+1)} - I_{(i,j)}]/h$	$dI_y = [I_{(i,j)} - I_{(i,j-1)}]/h$

gradient in both the horizontal and vertical directions. Here, to calculate $dI_{x(2,2)}$, we have utilized the intensities $I_{(1,2)}$ and $I_{(3,2)}$, while the intensities of pixels 2 and 8 were used for calculating the dI_y for pixel 5.

2. The angular orientation, θ, of each pixel is calculated using

$$\theta \text{ deg} = \tan^{-1}\frac{dI_y}{dI_x}. \quad (1)$$

3. The preferred orientation for each block is calculated using the angular orientation of each of its constituent pixels. The procedure is best explained with an example. Here an image block of 5 pixels is considered with arbitrary angular orientation values for each pixel. The angular orientations are displayed in Table 2 and the orientation is depicted graphically as orientation lines in Fig. 6.

Table 2 Angular orientation of pixels in an image block.

Pixel number	Angular orientation, θ (rad)
1	0.52
2	2.1
3	2.36
4	–2.61
5	1.92

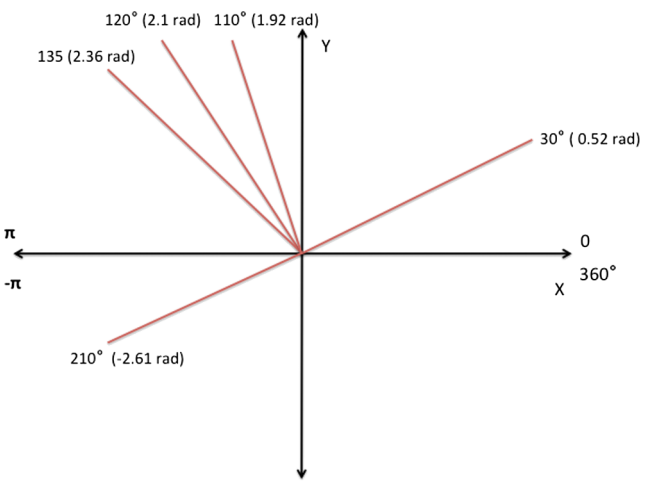

Fig. 6 Angular orientation of the pixels in an image block, represented in a Cartesian co-ordinate system.

For clarity in visualization, any angular orientation in the third and fourth quadrant is shifted to the first quadrant. In this example, the angular orientation of pixel 4 (−2.61 rad) is in the third quadrant, and thus, it is shifted by π to the first quadrant. The new set of angular orientation values are mentioned in Table 3 and graphically displayed in Fig. 7.

Next, the angular domain of 0 to π is divided into six equally spaced regions. Figure 7 shows the regional divisions with dotted blue lines along with the original orientation lines. The number of regions chosen is based on the level of accuracy required in the calculation of circular variance. For each region, a separate set of angular orientation data is obtained. Here, the lower of the two orientation lines is denoted as the division line (θ) and any pixel with its orientation below θ is shifted by π,

Table 3 Angular orientation of pixels in the first and second quadrants.

Pixel number	Angular orientation, θ (rad)
1	0.52
2	2.1
3	2.36
4	0.52
5	1.92

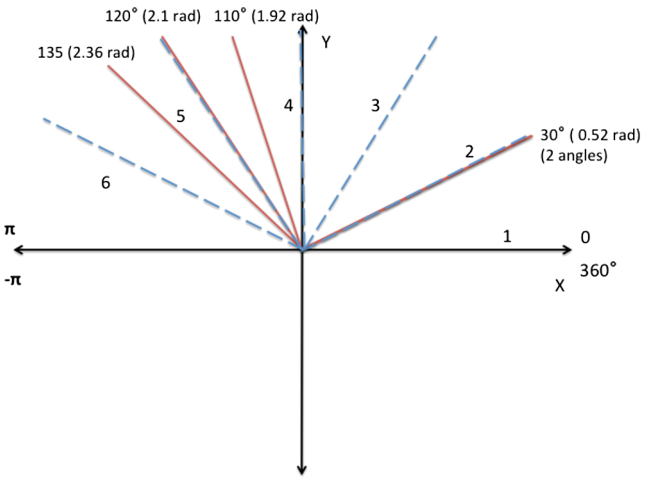

Fig. 7 The angular orientation of all pixels after they are shifted to the first and second quadrants. The angular domain of 0 to π is shown as divided into six regions. See text for details.

while any orientation value above $(\theta + \pi)$ is shifted by $-\pi$. It should be noted that this step does not change the angular orientation, but rather expresses it in a different angular domain. After this step, the changed set of angular orientations for all the regions is shown in Table 4. As an example, for set 5, the division line is 2.10 rad. So, the angular orientations of the first three pixels—0.52, 0.52, and 1.92—are observed to fall below the division line and were shifted by π to 3.66, 3.66, and 5.06. These results are shown in Table 4 and Fig. 8.

4. In the next step, the circular variance is calculated. A detailed discussion of this procedure is provided elsewhere.[19,20] Briefly, for each set, the angular orientations are expressed in the complex form, $(Ae^{i\theta})$. As only the angular orientation is of importance, the amplitude A is considered 1 for all cases.

$$e^{i\theta} = \cos\,\theta + i\,\sin\,\theta, \tag{2}$$

$$\sum e^{i\theta} = \sum \cos\,\theta + i\sum \sin\,\theta. \tag{3}$$

Table 4 Six sets of angular orientation for six regions.

Pixels		1	2	3	4	5
Sets	Division line	Angular orientation, θ (rad)				
1	0	0.52	0.52	1.92	2.10	2.36
2	0.53	0.52	0.52	1.92	2.10	2.36
3	1.06	3.66	3.66	1.92	2.10	1.36
4	1.59	3.66	2.66	1.92	2.10	2.36
5	2.10	3.66	3.66	5.06	2.10	2.36
6	2.65	3.66	3.66	5.06	5.24	5.50

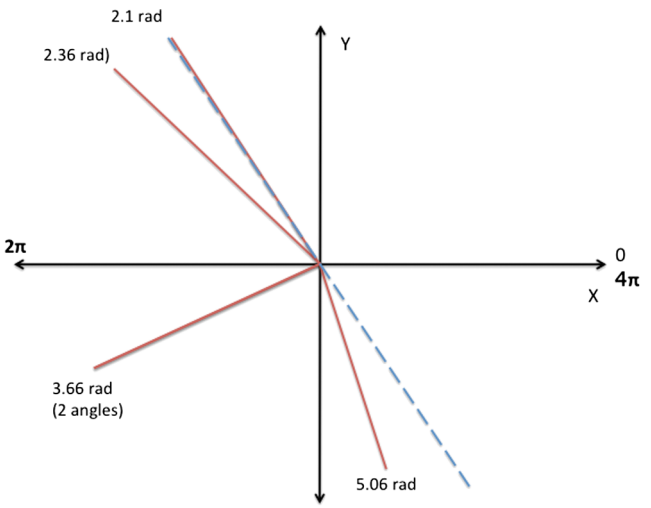

Fig. 8 Angular orientation for set 5 corresponding to a division line of 2.1 rad. For this set, the angular orientation for each pixel falls within a region of $2.10 < \theta < (2.10 + \pi)$.

Table 5 Calculated circular variance for six sets of angular orientation values.

Sets	1	2	3	4	5	6
Circular variance	0.2989	0.2989	0.2748	0.2748	0.4727	0.2989

The circular variance is calculated using the following equation:

$$C = 1 - R, \tag{4}$$

where

$$R = \frac{1}{n} \times \left(\sum \cos^2\,\theta + \sum \sin^2\,\theta \right). \tag{5}$$

The circular variance for the six sets of angular orientations in this example is shown in Table 5.

5. Finally, the mean angular orientation is individually calculated for each set. All the mean orientation values obtained for the different transformed sets correspond to the same original set of angular orientations. However, the mean with the lowest circular variance is identified as the preferred orientation. It is seen in Table 5 that the third and fourth sets have the lowest variance in this case. If the mean orientation for these two sets is calculated, it is observed to be the same value of 2.71 rad or ~155 deg. This value is defined as the preferred orientation for the given set of angular orientations.

Acknowledgments

M.M.K. acknowledges support from the National Science Foundation CAREER award (DBI 09-54155).

References

1. I. Freund and M. Deutsch, "Second-harmonic microscopy of biological tissue," *Opt. Lett.* **11**(2), 94–96 (1986).
2. R. A. Rao, M. R. Mehta, and K. C. Toussaint, "Fourier transform-second-harmonic generation imaging of biological tissues," *Opt. Express* **17**(17), 14534–14542 (2009).
3. R. A. R. Rao et al., "Quantitative analysis of forward and backward second-harmonic images of collagen fibers using Fourier transform second-harmonic-generation microscopy," *Opt. Lett.* **34**(24), 3779–3781 (2009).
4. R. Ambekar Ramachandra Rao, M. R. Mehta, and J. K. C. Toussaint, "Quantitative analysis of biological tissues using Fourier transform-second-harmonic generation imaging," *Proc. SPIE* **7569**, 75692G (2010).
5. R. Ambekar et al., "Quantifying collagen structure in breast biopsies using second-harmonic generation imaging," *Biomed. Opt. Express* **3**(9), 2021–2035 (2012).
6. T. Y. Lau, R. Ambekar, and K. C. Toussaint, "Quantification of collagen fiber organization using three-dimensional Fourier transform-second-harmonic generation imaging," *Opt. Express* **20**(19), 21821–21832 (2012).
7. M. Sivaguru et al., "Quantitative analysis of collagen fiber organization in injured tendons using Fourier transform-second harmonic generation imaging," *Opt. Express* **18**(24), 24983–24993 (2010).
8. R. Ambekar et al., "Quantitative second-harmonic generation microscopy for imaging porcine cortical bone: comparison to SEM and its potential to investigate age-related changes," *Bone* **50**(3), 643–650 (2012).

9. R. Tanaka et al., "In vivo visualization of dermal collagen fiber in skin burn by collagen-sensitive second-harmonic-generation microscopy," *J. Biomed. Opt.* **18**(6), 061231 (2013).
10. M. M. Kabir et al., "Application of quantitative second-harmonic generation microscopy to dynamic conditions," *Biomed. Opt. Express* **4**(11), 2546–2554 (2013).
11. C. M. O. Y. Wang et al., "GPU accelerated real-time multi-functional spectral domain optical coherence tomography system at 1300 nm," *Opt. Express* **20**(14), 14797–14813 (2012).
12. N. H. Cho et al., "High speed SD-OCT system using GPU accelerated mode for in vivo human eye imaging," *J. Opt. Soc. Korea* **17**(1), 68–72 (2013).
13. J. Li et al., "Performance and scalability of Fourier domain optical coherence tomography acceleration using graphics processing units," *Appl. Opt.* **50**(13), 1832–1838 (2011).
14. E. Alerstam et al., "Next-generation acceleration and code optimization for light transport in turbid media using GPUs," *Biomed. Opt. Express* **1**(2), 658–675 (2010).
15. L. A. Flores et al., "Parallel CT image reconstruction based on GPUs," *Radiat. Phys. Chem.* **95**, 247–250 (2014).
16. M. A. Bruce and M. J. Butte, "Real-time GPU-based 3D deconvolution," *Opt. Express* **21**(4), 4766–4773 (2013).
17. O. Yang and B. Choi, "Accelerated rescaling of single Monte Carlo simulation runs with the graphics processing unit (GPU)," *Biomed. Opt. Express* **4**(11), 2667–2672 (2013).
18. M. N. Shneider, A. A. Voronin, and A. M. Zheltikov, "Action-potential-encoded second-harmonic generation as an ultrafast local probe for non-intrusive membrane diagnostics," *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* **81**(3 Pt 1), 031926 (2010).
19. N. I. Fisher, *Statistical Analysis of Circular Data*, Cambridge University Press, Cambridge (1995).
20. A. S. S. R. Jammalamadaka, *Topics in Circular Statistics*, World Scientific, Singapore (2001).
21. R. Gonzalez and R. Woods, *Digital Image Processing*, 3rd ed., Prentice Hall, New Jersey (2007).
22. G. Stockman and L. G. Shapiro, *Computer Vision*, Prentice Hall PTR, New Jersey (2001).
23. NVIDIA Corporation, *CUDA Programming Guide 5.0* (2014).

Mohammad Mahfuzul Kabir is a doctoral candidate in the Department of Electrical and Computer Engineering at University of Illinois at Urbana Champaign (UIUC). He has a master's degree in mechanical engineering from UIUC and a bachelor's degree in mechanical engineering from Bangladesh University of Engineering and Technology (BUET), in Dhaka, Bangladesh. His current research focus is on developing quantitative nonlinear harmonic imaging techniques for potential applications in characterizing biological tissues.

A. S. M. Jonayat is a graduate research assistant at Illinois Applied Research Institute. He received his MS in mechanical engineering from University of Illinois at Urbana-Champaign in 2014. He worked in numerical modeling of thermo-fluid phenomena in continuous casting process, spray paint thin-film layer formation, and electro-osmotic flow in nanochannels. His broader research interest includes high-performance computing, numerical methods, and multiscale modeling.

Sanjay Patel is a professor of electrical and computer engineering and Sony Faculty Scholar at the University of Illinois at Urbana-Champaign. He is also CEO and co-founder of Nuvixa, a company that is delivering innovative video communications technologies. He has done architecture, hardware verification, logic design, and performance modeling at Digital Equipment Corporation, Intel Corporation, and HAL Computer Systems, as well as provided consultation for Transmeta, Jet Propulsion Laboratory, HAL, Intel, and AGEIA Technologies.

Kimani C. Toussaint, Jr. is an associate professor in the Department of Mechanical Science and Engineering, and an affiliate in the Departments of Electrical and Computer Engineering, and Bioengineering at the University of Illinois at Urbana-Champaign. He directs an interdisciplinary lab that focuses on developing optical techniques for quantitatively imaging collagen-based tissues, and investigating the properties of plasmonic nanostructures for control of near-field optical forces. He is a senior member in SPIE, OSA, and IEEE.