Graphics processing unit accelerated non-uniform fast Fourier transform for ultrahigh-speed, real-time Fourier-domain OCT

Kang Zhang* and Jin U. Kang

Department of Electrical and Computer Engineering, The Johns Hopkins University,
3400 N. Charles St., Baltimore, Maryland 21218, USA

Abstract: We implemented fast Gaussian gridding (FGG)-based non-uniform fast Fourier transform (NUFFT) on the graphics processing unit (GPU) architecture for ultrahigh-speed, real-time Fourier-domain optical coherence tomography (FD-OCT). The Vandermonde matrix-based non-uniform discrete Fourier transform (NUDFT) as well as the linear/cubic interpolation with fast Fourier transform (InFFT) methods are also implemented on GPU to compare their performance in terms of image quality and processing speed. The GPU accelerated InFFT/NUDFT/NUFFT methods are applied to process both the standard half-range FD-OCT and complex full-range FD-OCT (C-FD-OCT). GPU-NUFFT provides an accurate approximation to GPU-NUDFT in terms of image quality, but offers >10 times higher processing speed. Compared with the GPU-InFFT methods, GPU-NUFFT has improved sensitivity roll-off, higher local signal-to-noise ratio and immunity to side-lobe artifacts caused by the interpolation error. Using a high speed CMOS line-scan camera, we demonstrated the real-time processing and display of GPU-NUFFT-based C-FD-OCT at a camera-limited rate of 122 k line/s (1024 pixel/A-scan).

2010 Optical Society of America

OCIS codes: (100.2000) Digital image processing; (110.4500) Optical coherence tomography; (170.3890) Medical optics instrumentation.

References and links

1. A. Stephen, Boppart, Mark E. Brezinski and James G. Fujimoto, “Surgical Guidance and Intervention,” in Handbook of Optical Coherence Tomography, B. E. Bouna and G. J Tearney, ed. (Marcel Dekker, New York, NY, 2001).

2. A. F. Low, G. J. Tearney, B. E. Bouna, and I. K. Jang, “Technology Insight: optical coherence tomography–current status and future development,” Nat. Clin. Pract. Cardiovasc. Med. 3(3), 154–162, quiz 172 (2006).

3. M. S. Jafri, R. Tang, and C. M. Tang, “Optical coherence tomography guided neurosurgical procedures in small rodents,” J. Neurosci. Methods 176(2), 85–95 (2009).

4. K. Zhang, W. Wang, J. Han, and J. U. Kang, “A surface topology and motion compensation system for microsurgery guidance and intervention based on common-path optical coherence tomography,” IEEE Trans. Biomed. Eng. 56(9), 2318–2321 (2009).

5. J. U. Kang, J.-H. Han, X. Liu, K. Zhang, C. G. Song, and P. Geihlach, “‘Endoscopic functional Fourier domain common path optical coherence tomography for microsurgery,” IEEE J. Sel. Top. Quantum Electron. 16(4), 781–792 (2010).

6. U. Sharma, N. M. Fried, and J. U. Kang, “All-fiber common-path optical coherence tomography: sensitivity optimization and system analysis,” IEEE J. Sel. Top. Quantum Electron. 11(4), 799–805 (2005).

7. B. Potsaid, I. Gorczynska, V. J. Srinivasan, Y. Chen, J. Jiang, A. Cable, and J. G. Fujimoto, “Ultrahigh speed spectral / Fourier domain OCT ophthalmic imaging at 70,000 to 312,500 axial scans per second,” Opt. Express 16(19), 15149–15169 (2008), http://www.opticsinfobase.org/oe/abstract.cfm?uri=oe-16-19-15149.

8. R. Huber, D. C. Adler, and J. G. Fujimoto, “Buffered Fourier domain mode locking: Unidirectional swept laser sources for optical coherence tomography imaging at 370,000 lines/s,” Opt. Lett. 31(20), 2975–2977 (2006).

9. W.-Y. Oh, B. J. Vakoc, M. Shishkov, G. J. Tearney, and B. E. Bouna, “>400 kHz repetition rate wavelength-swept laser and application to high-speed optical frequency domain imaging,” Opt. Lett. 35(17), 2919–2921 (2010).
S. Vergnole, D. Lévesque, and G. Lamouche, “Experimental validation of an optimized signal processing method to handle non-linearity in swept-source optical coherence tomography,” Opt. Express 18(10), 10446–10461 (2010), http://www.opticsinfobase.org/abstract.cfm?uri=oe-18-10-10446.

D. Hillmann, G. Huttmann, and P. Koch, “Using nonequispaced fast Fourier transformation to process optical coherence tomography signals,” Proc. SPIE 7372, 73720R (2009).

NVIDIA, “NVIDIA CUDA Compute Unified Device Architecture Programming Guide Version 3.1,” (2010).

NVIDIA, “NVIDIA CUDA CUFFT Library Version 3.1,” (2010).

NVIDIA, “NVIDIA CUDA C Best Practices Guide 3.1,” (2010).

L. Greengard, and J. Lee, “Accelerating the nonuniform fast Fourier transform,” SIAM Rev. 46(3), 443–454 (2004).

C. Dorrer, N. Belabas, J. Likforman, and M. Joffre, “Spectral resolution and sampling issues in Fourier-transform spectral interferometry,” J. Opt. Soc. Am. B 17(10), 1795–1802 (2000).

1. Introduction

Fourier-domain optical coherence tomography (FD-OCT) is capable of providing depth/time-resolved images of biological tissues noninvasively with micron level resolution. These features make FD-OCT systems suitable for applications in microsurgical guidance and intervention [1–6]. For practical clinical applications, FD-OCT systems require raw data acquisition, data processing, and visualization rate in excess of 10’s of kHz speed.

The required raw data acquisition rate (A-scan line rate) of FD-OCT systems has been achieved in recent years where > 100,000 line/s speeds are common [7–14]. Even > 1,000,000 line/s rate has been achieved for multi-channel FD-OCT [15]. The speed of FD-OCT is determined either by the line rate of CMOS/CCD camera (for spectrometer-based systems) or by the spectral sweeping frequency (for swept laser-based systems). Ultrahigh acquisition speed is essential for time-resolved 4D recording of dynamic biological processes such as eye blinking, papillary reaction to light stimulus [10,11], and embryonic heart beating [12–14].

However, data processing and image rendering/display speed have not kept up with the ever increasing rate of data acquisition; this became one of the limiting factors for employing FD-OCT systems in practical microsurgery guidance and intervention applications. Currently, for most FD-OCT systems, the raw data is acquired in real-time and saved for post-processing. For microsurgeries, such imaging protocol provides valuable “pre-operative/post-operative” images, but is incapable of providing real-time, “inter-operative” imaging for surgical guidance and visualization. In addition, standard FD-OCT systems suffer from spatially reversed complex-conjugate ghost images that could severely misguide the users. As a solution, the complex full-range FD-OCT (C-FD-OCT) has been utilized, which removes the complex-conjugate image by applying a phase modulation on interferogram frames [16–20]. A 140k line/s 2048-pixel C-FD-OCT has been implemented for volumetric anterior chamber imaging [20]. However, the complex-conjugate processing is even more time-consuming and presents an extra burden when providing real-time images during surgical procedures.

Several methods have been implemented to improve data processing and visualization of FD-OCT images: Field-programmable gate array (FPGA) has been applied to both spectrometer and swept source-based systems [21,22]; multi-core CPU parallel processing has been implemented and achieved 80,000 line/s processing rate on nonlinear-k polarization-sensitive OCT system and 207,000 line/s on linear-k systems, both with 1024-point/A-scan [23,24]. Moreover, recent progress in general-purpose computing on graphics processing units (GPGPU) makes it possible to implement heavy-duty OCT data processing and visualization on a variety of low-cost, many-core graphics cards.

For standard half-range FD-OCT, GPU-based data processing has been implemented on both linear-k and non-linear-k systems [25–27]. Real-time 4D OCT imaging has also been achieved up to 10 volume/s through GPU-based volume rendering [24,26]. We have found that, based on a 1024-pixel standard FD-OCT system using NVIDIA’s GTX 480 GPU, the
maximum processing line rate is >3,000,000 line/s (effectively >1,000,000 line/s under data transfer limit), which will be shown in detail in Section 4.

For complex full-range FD-OCT, the processing workload is more than 3 times the standard OCT, since each A-scan requires three fast Fourier transforms (FFT) in different dimensions of the frame, a band-pass filtering, and necessary matrix transpose [16]. In a separate work, we realized a real-time processing of 1024-pixel C-FD-OCT at >500,000 line/s (effectively >300,000 line/s under data transfer limit), and a real-time camera-limited display speed of 244,000 line/s [28]. Very recently, a 27,900 line/s 2048-pixel C-FD-OCT system was reported by Watanabe et al during the preparation of this manuscript [29].

In most FD-OCT systems, the signal is sampled nonlinearly in k-space, which will seriously degrade the image quality if the FFT is directly applied to such signal. So far there have been both hardware and software solutions to the nonlinear-k issue. Hardware solutions such as linear-k spectrometer [30], linear-k swept laser [31] and k-triggering [32] have been successfully implemented, but these methods generally increase the system complexity and cost. Software solutions include various interpolation methods such as simple linear interpolation, oversampled linear interpolation, zero-filling linear interpolation, and cubic spline interpolation. Different GPU-based interpolation methods have also been implemented and compared [25–27]. Alternatively, the non-uniform discrete Fourier transform (NUDFT) has been proposed recently for both swept source OCT [33] and spectrometer-based OCT [34] through direct Vandermonde matrix multiplication with the spectrum vector. Compared with the interpolation-FFT (InFFT) method, NUDFT is simpler to implement and immune to the interpolation-caused errors such as increased background noise and side-lobes, especially at larger image depth [35]. Moreover, NUDFT has improved sensitivity roll-off than the InFFT [34]. However, NUDFT by direct matrix multiplication is extremely time-consuming, with a complexity of O(N^3), where N is the raw data size of an A-scan. As an approximation to NUDFT, the gridding-based non-uniform fast Fourier transform (NUFFT) has been tried to process simulated [36] and experimentally acquired data [35] with reduced calculation complexity of ~O(NlogN). To the best of our knowledge, NUDFT/NUFFT have yet to be utilized in ultra-high speed, real-time FD-OCT systems due to computational complexity and associated latency in data processing.

In this work, we implemented the fast Gaussian gridding (FGG)-based NUFFT on the GPU architecture for ultrafast signal processing in a general FD-OCT system. The Vandermonde matrix-based NUDFT as well as the linear/cubic InFFT methods are also implemented on GPU as comparisons of image quality and processing speed. GPU-NUFFT provides a very close approximation to GPU-NUDFT in terms of image quality while offering >10 times higher processing speed. Compared with the GPU-InFFT methods, we have also observed improved sensitivity roll-off, a higher local signal-to-noise ratio, and absence of side-lobe artifacts in GPU-NUFFT. Using a high speed CMOS line-scan camera, we demonstrated the real-time processing and display of GPU-NUFFT-based C-FD-OCT at a camera-limited speed of 122 k line/s (1024 pixel/A-scan).

2. System configuration

The FD-OCT system used in this work is spectrometer-based, as shown in Fig. 1. A 12-bit dual-line CMOS line-scan camera (Sprint spL2048-140k, Basler AG, Germany) works as the detector of the OCT spectrometer. A superluminescence diode (SLED) (\(\lambda_0 = 825\)nm, \(\Delta\lambda = 70\)nm, Superlum, Ireland) was used as the light source, which provided a theoretical axial resolution of approximately 5.5\(\mu\)m in air, 4.1\(\mu\)m in water. Since the SLED’s spectrum only covers less than half of the CMOS camera’s sensor array, the camera is set to work at 1024-pixel mode by selecting the area-of-interest (AOI). The camera works at the “double-line averaging mode” to get 3dB higher SNR of the raw spectrum [7], and the minimum line period is camera-limited to 7.8\(\mu\)s, which corresponds to a maximum line rate of 128k A-scan/s. The beam scanning was implemented by a pair of high speed galvanometer mirrors.
driven by a dual channel function generator and synchronized with a high speed frame grabber (PCIE-1429, National Instruments, USA). For C-FD-OCT mode, a phase modulation is applied to each B-scan’s 2D interferogram frame by slightly displacing the probe beam off the first galvanometer’s pivoting point (here only the first galvanometer is illustrated in the figure) [17,18]. The scanning angle is 8° and the beam offset is 1.4mm, therefore a carrier frequency of 30 kHz is obtained according to reference [18]. The transversal resolution was approximately 20µm, assuming Gaussian beam profile. A quad-core Dell T7500 workstation was used to host the frame grabber (PCIE x4 interface) and GPU (PCIE x16), and a GPU (NVIDIA GeForce GTX 480) with 480 stream processors (each processor working at 1.45GHz) and 1.5GBytes graphics memory was used to perform the FD-OCT signal processing. The GPU is programmed through NVIDIA’s Compute Unified Device Architecture (CUDA) technology [37]. The FFT operation is implemented by the CUFFT library [38]. The software is developed under Microsoft Visual C++ environment with the NI-IMAQ Win32 API (National Instrument).

3. Implementation of GPU-NUDFT and GPU-NUFFT in FD-OCT systems

In this section, the implementation of both GPU-NUDFT and GPU-NUFFT in a standard FD-OCT system is described. For the implementation, the wavenumber-pixel relation of the system \( k[i] = \frac{2\pi}{\lambda[i]} \) is pre-calibrated accurately, where \( i \) refers to the pixel index.

3.1 GPU-NUDFT in FD-OCT

After pre-calibrating the \( k[i] \) relation, the depth information \( A[z_m] \) can be implemented through discrete Fourier transform over non-uniformly distributed data \( [k_i] \), as in [33–35],

\[
A[z_m] = \sum_{i=0}^{N-1} [k_i] \exp \left[ -j \frac{2\pi}{\Delta k} (k_i - k_0) m \right], \quad m=0,1,2,...,N-1, \quad (1)
\]
where \( N \) is the total pixel number of a spectrum, \( z_m \) refers to the depth coordinate with the pixel index \( m \). \( \Delta k = k_{m+1} - k_m \) is the wavenumber range.

For standard FD-OCT, where \( I[k] \) are real-values, Eq. (1) can be reduced to half-range as,

\[
A[z_m] = \sum_{i=0}^{N-1} I[k_i] \exp \left[ -j \frac{2 \pi}{\Delta k} (k_i - k_m) \right] , \quad m=0,1,2,\ldots,N/2-1.
\] (2)

For C-FD-OCT, where \( I[k] \) are complex-values after Hilbert transform [16], Eq. (1) can be modified to full-range as,

\[
A[z_m] = \sum_{i=0}^{N-1} I[k_i] \exp \left[ -j \frac{2 \pi}{\Delta k} (k_i - k_m) \left( \frac{N}{2} - m \right) \right] , \quad m=0,1,2,\ldots,N-1. \] (3)

Here the index \( m \) is shifted by \( N/2 \) to set the DC component to the center of \( A[z_m] \).

Considering a frame with \( M \) A-scans, Eq. (2) and (3) can be written in matrix form for processing the whole frame as,

\[
A_{\text{half}} = D_{\text{half}} I_{\text{real}}, \quad \text{(Standard half-range FD-OCT)},
\] (4)

where

\[
A_{\text{half}} = \begin{bmatrix}
A_0[z_0] & A_1[z_0] & \ldots & A_M[z_0] & A_{M+1}[z_0] \\
A_0[z_1] & A_1[z_1] & \ldots & A_M[z_1] & A_{M+1}[z_1] \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
A_0[z_{N/2-2}] & A_1[z_{N/2-2}] & \ldots & A_M[z_{N/2-2}] & A_{M+1}[z_{N/2-2}] \\
A_0[z_{N/2-1}] & A_1[z_{N/2-1}] & \ldots & A_M[z_{N/2-1}] & A_{M+1}[z_{N/2-1}]
\end{bmatrix},
\] (5)

\[
I_{\text{real}} = \begin{bmatrix}
I_0[z_0] & I_1[z_0] & \ldots & I_M[z_0] & I_{M+1}[z_0] \\
I_0[z_1] & I_1[z_1] & \ldots & I_M[z_1] & I_{M+1}[z_1] \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
I_0[z_{N/2-2}] & I_1[z_{N/2-2}] & \ldots & I_M[z_{N/2-2}] & I_{M+1}[z_{N/2-2}] \\
I_0[z_{N/2-1}] & I_1[z_{N/2-1}] & \ldots & I_M[z_{N/2-1}] & I_{M+1}[z_{N/2-1}]
\end{bmatrix},
\] (6)

\[
D_{\text{half}} = \begin{bmatrix}
1 & 1 & \ldots & 1 & 1 \\
p_0^{(N/2-2)} & p_1^{(N/2-2)} & \ldots & p_{N/2-2}^{(N/2-2)} & p_{N-1}^{(N/2-2)} \\
p_0^{(N/2-1)} & p_1^{(N/2-1)} & \ldots & p_{N/2-1}^{(N/2-1)} & p_{N-1}^{(N/2-1)}
\end{bmatrix},
\] (7)

and

\[
A_{\text{full}} = D_{\text{full}} I_{\text{complex}}, \quad \text{(Standard half-range FD-OCT)},
\] (8)

where
where the subscript of $A[z_i]$ and $I[k_i]$ denotes the index of A-scan within one frame, the complex factor $\left[ \begin{array}{c} \exp \left( j \frac{2\pi}{\Delta k^*} (k - k_0) \right) \end{array} \right]$, $D_{full}$ and $D_{half}$ are the Vandermonde matrix, which can be pre-calculated from $k[i]$.

To realize C-FD-OCT mode, a phase modulation $\phi(x) = \beta x$ is applied to each B-scan’s 2D interferogram frame $I(k, x)$ by slightly displacing the probe beam off the galvanometer’s pivoting point, as shown in Fig. 1. Here $x$ indicates A-scan index in each B-scan and by applying Fourier transform along x direction, the following equation can be obtained [16]:

$$ F_{x \rightarrow u} [I(k, x)] = \left| E_x(k, x) \right|^2 \delta(u) + \Gamma_u \{ F_{x \rightarrow u} [E_x(k, x)] \} + F_{x \rightarrow u} [E_x^*(k, x) E_x(k)] \otimes \delta(u + \beta) + F_{x \rightarrow u} [E_x(k, x) E_x^*(k)] \otimes \delta(u - \beta), $$

where $E_x(k, x)$ and $E_x(k)$ are the electrical fields from the sample and reference arms, respectively. $\Gamma_u \{ \}$ is the correlation operator. The first three terms on the right hand of Eq. (12) present the DC noise, autocorrelation noise, and complex-conjugate noise, respectively. The last term can be filtered out by a proper band-pass filter in the $u$ domain and then convert back to $x$ domain by applying an inverse Fourier transform along $x$ direction.

Here to implement the standard Hilbert transform, we use the Heaviside step function as the band-pass filter and the more delicate filters such as super Gaussian filter can also be designed to optimize the performance [29]. Finally, the OCT image is obtained by NUDFT in $k$ domain and logarithmically scaled for display.
Fig. 2. Processing flowchart for GPU-NUDFT based FD-OCT: CL, Camera Link; FG, frame grabber; HM, host memory; GM, graphics global memory; DC, DC removal; MT, matrix transpose; FFT-x, Fast Fourier transform in x direction; IFFT-x, inverse Fast Fourier transform in x direction; BPF-x, band pass filter in x direction; Log, logarithmical scaling. The solid arrows describe the main data stream and the hollow arrows indicate the internal data flow of the GPU. The blue dashed arrows describe the direction of inter-thread triggering. The hollow dashed arrow denotes standard FD-OCT without the Hilbert transform in x direction. Blue blocks: memory for pre-stored data; Yellow blocks: memory for real-timely refreshed data.

The GPU-CPU hybrid processing flowchart for standard/complex FD-OCT using GPU-NUDFT is shown in Fig. 2, where three major threads are used for the data acquisition (Thread 1), the GPU processing (Thread 2), and the image display (Thread 3), respectively. The three threads are triggered unidirectionally and work in the synchronized pipeline mode, as indicated by the blue dashed arrows in Fig. 2. Thread 2 is a GPU-CPU hybrid thread which consists of hundreds of thousands of GPU threads. The solid arrows describe the main data stream and the hollow arrows indicate the internal data flow of the GPU. The DC removal is implemented by subtracting a pre-stored frame of reference signal. The Vandermonde matrix $D_{full}/D_{half}$ is pre-calculated and stored in graphics memory, as the blue block in Fig. 2. The NUDFT is implemented by an optimized matrix multiplication algorithm on CUDA using shared memory technology for the maximum usage of the GPU’s floating point operation ability [39]. The graphics memory mentioned at the current stage of our system refers to global memory, which has relatively lower bandwidth and is another major limitation of GPU processing in addition to the PCIE x16 bandwidth limit. The processing speed can be further increased by mapping to texture memory, which has higher bandwidth than global memory.

3.2 GPU-NUFFT in FD-OCT

The direct GPU-NUDFT presented in Section 3.1 has a computation complexity of $O(N^2)$, which greatly limits the computation speed and scalability for real-time display even on GPU, as experimentally shown in Section 4. Alternative to direct GPU-NUDFT, here we implemented fast Gaussian gridding-based GPU-NUFFT to approximate GPU-NUDFT: the raw signal $I[k_i]$ is first oversampled by convolution with a Gaussian interpolation kernel on a uniform grid, as [40],

$$I_i[u] = \sum_i I[i] g_c[k_i, [u] - k[i]], u=0,1,2,\ldots,Mr-1,$$

(13)

$$g_c[k] = \exp \left[-\frac{k^2}{4\tau}\right].$$

(14)
\[
\tau = \frac{1}{N^2} \frac{\pi}{R(R-0.5)} M_{sp} ,
\]

(15)

\[
G_i[n] = \exp[n^2 \tau] ,
\]

(16)

where \( k_i[u] \) is uniform grid covering the same range as \( k[i] \), \( g_i[k] \) is the Gaussian interpolation kernel, \( M_i = R \times N \) is the uniform gridding size, and \( R \) is the oversampling ratio. On each \( k_i[u] \) grid, the source data \( I[i] \) is selected such that \( k_i[u] \) is within the nearest \( 2 \times M_{sp} \) grids to \( k[i] \), where \( M_{sp} \) is the kernel spread factor. The calculation of Eq. (13) is illustrated in Fig. 3, where \( R = 2 \), \( M_{sp} = 2 \) and values of the blue dots on the same \( k_i[u] \) grid are summed together to obtain \( I_i[u] \). The selection of proper \( R \) and \( M_{sp} \) will be further discussed in Section 4.2. Subsequently \( I_i[u] \) is processed by the regular uniform FFT and finally undergo a deconvolution of \( g_i[k] \) by multiplying the factor \( G_i[n] \), as in Eq. (16), where \( n \) denotes the axial index. If \( R > 1 \), there will be redundant data due to oversampling and the final data needs to be truncated to the size of \( N \). The processing flowchart of GPU-NUFFT-based FD-OCT is shown in Fig. 4.

Here it is worth noting that the Kaiser-Bessel function is found to be the optimal convolution kernel for the gridding-based NUFFT shown in recent works [35,36]. The implementation of Kaiser-Bessel convolution on GPU is similar to the Gaussian kernel and will be studied and evaluated in our future work.

Fig. 3. Convolution with a Gaussian interpolation kernel on a uniform grid when \( R = 2 \), \( M_{sp} = 2 \).
4. GPU processing test and comparison of different FD-OCT methods

4.1 GPU processing line rate for different FD-OCT methods

First we performed benchmark line rate test of different FD-OCT processing methods as follows:

- LIFFT: Standard FD-OCT with linear spline interpolation;
- LIFFT-C: C-FD-OCT with linear spline interpolation;
- CIFFT: Standard FD-OCT with cubic spline interpolation;
- CIFFT-C: C-FD-OCT with cubic spline interpolation;
- NUDFT: Standard FD-OCT with NUDFT;
- NUDFT-C: C-FD-OCT with NUDFT;
- NUFFT: Standard FD-OCT with NUFFT;
- NUFFT-C: C-FD-OCT with NUFFT;

All algorithms are tested on the GTX 480 GPU with 4096 lines of both 1024-pixel spectrum and 2048-pixel spectrum. Here the 2048-pixel mode is tested as reference only and we will use the 1024-pixel mode for the real-time imaging tests in this work. For each case, both the peak internal processing line rate and the reduced line rate considering the data transfer bandwidth of PCIE x16 interface are listed in Fig. 5. The processing time is measured using CUDA functions and all CUDA threads are synchronized before the measurement. In our system, the PCIE x16 interface between the host computer and the GPU has a data transfer bandwidth of about 4.6 GByte/s for both transfer-in and transfer-out. The data size is 8.4 MByte for 4096 lines of 1024-pixel spectrum and 16.8 MByte for 4096 lines of 2048-pixel spectrum, by allocating 2 Byte for each pixel.
Fig. 5. Benchmark line rate test of different FD-OCT processing method. (a) 1024-pixel FD-OCT; (b) 2048-pixel FD-OCT; (c) 1024-pixel NUFFT-C with different frame size; (d) 2048-pixel NUFFT-C with different frame size; Both the peak internal processing line rate and the reduced line rate considering the data transfer bandwidth of PCIE x16 interface are listed.

As in Fig. 5(a), the final processing line rate for 1024-pixel C-FD-OCT with GPU-NUFFT is 173k line/s, which is still higher than the maximum camera acquisition rate of 128k line/s, while the GPU-NUDFT speed is relatively lower in both standard and complex FD-OCT. Also it is notable that the processing speed of LIFFT goes up to >3,000,000 line/s (effectively >1,000,000 line/s under data transfer limit), achieving the fastest processing line rate to date to the best of our knowledge.

For C-FD-OCT, the Hilbert transform, which is implemented by two Fast Fourier transforms, has the computational complexity of ~O(M*logM), where M is the number of lines within one frame. Therefore the processing line rate of C-FD-OCT is also influenced by the frame size M. To verify this, here we tested the relation between processing line rate of
NUFFT-C mode versus frame size, as shown in Fig. 5(c) and 5(d), and a speed decrease is observed with increasing frame size.

The zero-filling interpolation with FFT is also effective in suppressing the side-lobe effect and background noise for FD-OCT. However, the zero-filling usually require an oversampling factor of 4 or 8, and two additional FFT, which considerably increase the array size and processing time of the data [41]. From Fig. 5(b), for 2048-pixel NUFFT-C mode, a peak internal processing speed of about 100,000 line/s is achieved. This is >40% faster than 70,000 line/s (1024 lines for 14.75ms) achieved in reference [29], which implemented GPU-based zero-filling interpolation for C-FD-OCT using a 480-core GPU (NVIDIA GeForce GTX 295, 2 × 240 core, 1.3GHz for each core).

4.2 Comparison of point spread function and sensitivity roll-off

Then we compared the point spread function (PSF) and sensitivity roll-off of different FD-OCT processing methods, as shown in Fig. 6(a)–6(d), using GPU based LIFFT, CIFFT, NUDFT and NUFFT, respectively. A mirror is used as an image sample for evaluating the PSF. Here 1024 of GPU-processed 1024-pixel A-scans are averaged to present the mean noise level for the PSF in each axial position [35]. From Fig. 6(a), it is noticed that using LIFFT method introduces significant background level and side-lobes around the signal peak. The side-lobes tend to be broad and extend to their neighboring peaks, which results in a significant degrade of OCT image. When CIFFT is used instead, as shown in Fig. 6(b), the side-lobes are suppressed in the shallow image depth, but still considerably high in deeper depth. Figure 6(c) and 6(d) shows significant suppression of side-lobes over the whole image depth which utilized NUDFT and NUFFT, respectively. Figure 6(e) presents an individual PSF at a certain depth close to the imaging depth limit (intensity of each A-scan is normalized). Both LIFFT and CIFFT have large side-lobes, while both NUDFT and NUFFT have flat background and no noticeable side-lobes, which also indicate a higher local SNR at deeper axial positions. The measured sensitivity roll-off is summarized in Fig. 6(f). All methods have a very close sensitivity level at shallow image depth <400µm, while the interpolation based methods presents faster roll-off rate as axial position is increased. Here the sensitivity roll-off of LIFFT and CIFFT can be further improved by deconvolution of the linear/cubic interpolation kernel, and the result would be more close to NUDFT and NUFFT [35,36]. It is also notable from Fig. 6(e) and 6(f) that the NUFFT method exhibit negligible difference from the more time-consuming NUDFT, indicating a very close and reliable approximation. The full width at half maximum (FWHM) values of individual A-scans processed by different methods are presented in Fig. 6(g), which indicates very close axial resolution for all methods.

Figure 6(h) presents the processing result of a certain A-scan using NUFFT with R = 2 and different \( M_{sp} \) from 1 to 3. Here R is set to 2 for the convenience of applying GPU-based FFT functions (for length of \( 2^N \)). From Fig. 6(h), it can be noticed that for \( M_{sp} \geq 2 \), the NUFFT result is close enough to NUDFT result, therefore \( M_{sp} = 2 \) is selected to decrease the computational load.
4.3 Comparison of real-time image quality

We compared the real-time image quality of GPU-accelerated C-FD-OCT using different processing methods. Here a multi-layered polymer phantom is used as a sample. In the scanning protocol, each frame consists of 4296 A-scans in acquisition, but the first 200 lines are disposed before processing, since they are within the fly-back period of the galvanometer. Therefore each frame-size is 4096 pixel (lateral) × 1024 pixel (axial).
First, a single frame is captured and GPU-processed using different methods, shown in Fig. 7. The red arrows in Fig. 7(a) and 7(b) indicate the ghost image due to the side-lobes of the reflective surface at a deeper image depth relative to ZDL. The red lines correspond to the A-scans extracted from the same lateral position of each image, shown collectively in (i). The side-lobes of LIFFT/CIFFT are indicated by the blue arrow in (i).
lines correspond to the same A-scan position extracted from each image for comparison and shown collectively in (i). The resulting LIFFT/CIFFT images exhibit side-lobes in the order of 10/5 dB high compared to NUDFT/NUFFT images, as indicated by the blue arrow in Fig. 7(i).

Then we screen-captured the real-time displayed scenarios using different GPU-accelerated C-FD-OCT, shown in Media 1, Media 2, Media 3, and Media 4. The image frames are rescaled to 1024 pixel (lateral) × 512 pixel (axial) to accommodate the monitor display. LIFFT/CIFFT/NUFFT modes are running at 29.8 fps, corresponding to a camera-limited line rate of 122k line/s, while the NUDFT mode is GPU-limited to 9.3 fps (38k line/s). As shown in both Fig. 7 and the corresponding movies, in LIFFT/CIFFT modes, the ghost image is evident when the phantom surface moves further away from the ZDL. However, there is no evidence of ghost images in NUDFT/NUFFT modes wherever the surface is placed.

From these result, it is clear that the GPU-NUFFT method is a very close approximation of GPU-NUDFT while offering much higher processing speed. GPU-NUFFT can be achieved at a comparable processing speed to GPU-CIFFT and is immune to interpolation error-caused ghost images.

5. In vivo human finger imaging using GPU-NUFFT based C-FD-OCT

Finally we conducted the in vivo human finger imaging using GPU-NUFFT-based C-FD-OCT, displayed at 29.8fps with original frame size of 4096 pixel (lateral) × 1024 pixel (axial). Figure 8(a) and 8(b) present the coronal scans of the finger tip and palm, where the epithelial structures such as sweat duct (SD), stratum corneum (SC) and stratum spinosum (SS) are clearly distinguishable. Figure 8(c) and 8(d) present the coronal scans of the finger nail fold region, showing the major dermatologic structures such as epidermis (E), dermis (D), nail bed (NB), and nail root (NR), as well as in the sagittal scans in Fig. 8(e) and 8(f). The real-time display for each figure is captured as Media 5, Media 6, Media 7, Media 8, and Media 9, at 1024 pixel (lateral) × 512 pixel (axial). Compared to standard FD-OCT, the GPU-NUFFT C-FD-OCT image is free of conjugate artifact, DC noise, and autocorrelation noise. These noises are problematical to remove in standard FD-OCT. Moreover, due to the implementation of the complex OCT processing, the image depth is effectively doubled, with the highest SNR region in the zero delay point. Such ultra high-speed real-time C-FD-OCT could be highly useful for microsurgical guidance and intervention applications.
6. Conclusion

In this work, we implemented and successfully demonstrated the FGG-based NUFFT on the GPU architecture for online signal processing in a general FD-OCT system. The Vandermonde matrix-based NUDFT as well as the linear/cubic InFFT methods were also implemented on GPU as comparisons of image quality and processing speed. GPU-NUFFT provides an accurate approximation to GPU-NUDFI in terms of image quality while offering >10 times higher processing speed. Compared to the GPU-InFFT methods, GPU-NUFFT has better sensitivity roll-off, a higher local signal-to-noise ratio, and is immune to the side-lobe artifacts caused by interpolation error. Using a high speed CMOS line-scan camera, we demonstrated the real-time processing and display of GPU-NUFFT-based C-FD-OCT at a camera-limited speed of 122 k line/s (1024 pixel/A-scan). The GPU processing speed can be increased even higher by implementing a multiple-GPU architecture using more than one GPU in parallel.

Acknowledgments

This work was supported by National Institutes of Health (NIH) grant R21 1R21NS063131-01A1.