GPU-accelerated non-uniform fast Fourier transform-based compressive sensing spectral domain optical coherence tomography

Daguang Xu,* Yong Huang, and Jin U. Kang

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

*dxu5@jhu.edu

Abstract: We implemented the graphics processing unit (GPU) accelerated compressive sensing (CS) non-uniform in k-space spectral domain optical coherence tomography (SD OCT). Kaiser-Bessel (KB) function and Gaussian function are used independently as the convolution kernel in the gridding-based non-uniform fast Fourier transform (NUFFT) algorithm with different oversampling ratios and kernel widths. Our implementation is compared with the GPU-accelerated modified non-uniform discrete Fourier transform (MNUDFT) matrix-based CS SD OCT and the GPU-accelerated fast Fourier transform (FFT)-based CS SD OCT. It was found that our implementation has comparable performance to the GPU-accelerated MNUDFT-based CS SD OCT in terms of image quality while providing more than 5 times speed enhancement. When compared to the GPU-accelerated FFT based-CS SD OCT, it shows smaller background noise and less side lobes while eliminating the need for the cumbersome k-space grid filling and the k-linear calibration procedure. Finally, we demonstrated that by using a conventional desktop computer architecture having three GPUs, real-time B-mode imaging can be obtained in excess of 30 fps for the GPU-accelerated NUFFT based CS SD OCT with frame size 2048(axial)$\times$1000(lateral).

OCIS codes: (170.4500) Optical coherence tomography; (100.3010) Image reconstruction techniques; (100.2000) Digital image processing.

References and links

1. D. L. Donoho, "Compressed sensing," IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006).
2. E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," Inf. Theory **52**(2), 489–509 (2006).
3. X. Liu, and J. U. Kang, "Compressive SD-OCT: the application of compressed sensing in spectral domain optical coherence tomography," Opt. Express **18**(21), 22010–22019 (2010).
4. N. Zhang, T. Huo, C. Wang, T. Chen, J. Zheng, and P. Xue, "Compressed sensing with linear-in-wavenumber sampling in spectral-domain optical coherence tomography," Opt. Lett. **37**(15), 3075–3077 (2012).
5. D. Xu, N. Vaswani, Y. Huang, and J. U. Kang, "Modified compressive sensing optical coherence tomography with noise reduction," Opt. Lett. **37**(20), 4209–4211 (2012).
6. S. Schwartz, C. Liu, A. Wong, D. A. Clausi, P. Fieguth, and K. Bizheva, "Energy-guided learning approach to compressive sensing," Opt. Express **21**(1), 329–344 (2013).

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

7. C. Liu, A. Wong, K. Bizheva, P. Fieguth, and H. Bie, "Homotopic, non-local sparse reconstruction of optical coherence tomography imagery," Opt. Express **20**(9), 10200–10211 (2012).
8. L. Fang, S. Li, Q. Nie, J. A. Izatt, C. A. Toth, and S. Farsiu, "Sparsity based denoising of spectral domain optical coherence tomography images," Biomed. Opt. Express **3**(5), 927–942 (2012).
9. D. Xu, Y. Huang, and J. U. Kang, "Compressive sensing with dispersion compensation on non-linear wavenumber sampled spectral domain optical coherence tomography," Biomed. Opt. Express **4**(9), 1519–1532 (2013).
10. K. Wang, Z. Ding, T. Wu, C. Wang, J. Meng, M. Chen, and L. Xu, "Development of a non-uniform discrete Fourier transform based high speed spectral domain optical coherence tomography system," Opt. Express **17**(4) 12121–12131 (2009).
11. L. Greengard and J. Lee, "Accelerating the nonuniform fast Fourier transform," SIAM Rev. **46**(3), 443–454 (2004).
12. D. Hillmann, G. Huttmann, and P, Koch, "Using nonequispaced fast Fourier transformation to process optical coherence tomography signals," Proc. SPIE 7372 73720R (2009).
13. K. K. H. Chan and S. Tang, "High-speed spectral domain optical coherence tomography using non-uniform fast Fourier transform," Biomed. Opt. Express **1**(5), 1308–1319 (2010).
14. K. Zhang, and J. U. Kang, "Graphics processing unit accelerated non-uniform fast Fourier transform for ultrahigh-speed, real-time Fourier-domain OCT," Opt. Express **18**(22), 23472–23487 (2010).
15. K. K. H. Chan and S. Tang, "Selection of convolution kernel in non-uniform fast Fourier transform for Fourier domain optical coherence tomography," Opt. Express **19**(27), 26891–26904 (2011).
16. K. Zhang and J. U. Kang, "Real-time 4D signal processing and visualization using graphics processing unit on a regular nonlinear-k-Fourier-domain OCT system," Opt. Express **18**(11), 11772–11784 (2010).
17. Y. Watanabe, and T. Itagaki, "Real-time display on Fourier domain optical coherence tomography system using a graphics processing unit," J. Biomed. Opt. **14**(6), 060506 (2009).
18. S. Van der Jeught, A. Bradu, and A. G. Podoleanu, "Real-time resampling in Fourier domain optical coherence tomography using a graphics processing unit," J. Biomed. Opt. **15**(3), 030511 (2010).
19. M. Murphy, M. Alley, J. Demmel, K. Keutzer, S. Vasanawala, and M. Lustig, "Fast l_1-SPIRiT compressed sensing parallel imaging MRI: scalable parallel implementation and clinically feasible runtime," IEEE Trans. Med. Imaging **31**(6), 1250–1262 (2012).
20. S. Lee and S. J. Wright, "Implementing algorithms for signal and image reconstruction on graphical processing units," Technical Report, University of Wisconsin-Madison (2008).
21. D. Yang, G. D. Peterson, and H. Li, "Compressed sensing and Cholesky decomposition on FPGAs and GPUs," Parallel Comput. **38**(8), 421–437 (2012).
22. D. Xu, Y. Hunag, and J. U. Kang, "Real-time compressive sensing spectral domain optical coherence tomography," Opt. Lett. **39**(1), 76–79 (2014).
23. S. Vergnole, D. Levesque, 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).
24. P. J. Beatty, D. G. Nishimura, and J. M. Pauly, "Rapid gridding reconstruction with a minimal over-sampling ratio," IEEE Trans. Med. Imaging **24**(6), 799–808 (2005).
25. "NVIDIA CUDA C programming guide version 5.5" (2013).
26. S. J. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," IEEE Trans. Signal Processing **57** 2479–2493 (2009).
27. J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," IMA J. Numer. Anal. **8**, 141–148 (1988).
28. "NVIDIA CUDA CUBLAS library version 5.5" (2013).
29. M. Harris, "Optimizing parallel reduction in CUDA," NVIDIA Dev. Tech (2007).
30. "NVIDIA Visual Profiler version 5.5" (2013).
31. "NVIDIA CUDA CUFFT library version 5.5" (2013).
32. T. Schmoll, C. Kollbitsch, and R. A. Leitgerb, "Ultra-high-speed volumetric tomography of human retinal blood flow," Opt. Express **17**(5), 4166–4176 (2009).
33. American National Standards Institute, *American National Standart for Safe Use of Lasers Z136.1. (2007).*
34. M. Lustig, D. Donoho, and J. M. Pauly, "Saprse MRI: the application of compressed sensing for rapid MR imaging," Magn. Reson, Med. **58**(6), 1182–1195 (2007).

1. Introduction

In the past decade, compressive sensing (CS) [1, 2] has emerged as an effective technique for sampling a signal with smaller number of measurements than that required by the classical Shannon/Nyquist theory. For an accurate reconstruction, CS requires that the signal is sparse in some domain that is incoherent to the sampling domain. In recent years, applications of CS in the spectral domain optical coherence tomography (SD OCT) have been reported by many

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

research groups [3–8] which shows that high-quality SD OCT images can be obtained from highly under-sampled k-space data.

Most recent studies on CS SD OCT take the uniform discrete Fourier transform (UDFT) or fast Fourier transform (FFT) as the sensing technique which requires the under-sampled linear wavenumber spectral sampling as the input data. Special operations are needed for obtaining such samplings since in most SD OCT systems, the spectra from the interferometer are nonlinear in wavenumber. In [3], the k-space grid-filling method is proposed which obtains the under-sampled nonlinear wavenumber data from the spectra first; then remaps them to the linear wavenumber grid for the under-sampled linear wavenumber sampling. Another method based on the pre-calibrated k-linear random mask is proposed in [4] which gets the under-sampled linear in wavenumber sampling directly from the nonlinear wavenumber spectra. As stated in [9], these two methods have their own limitations: the previous one requires a complicated and time-consuming spectral calibration with numerical interpolation; the latter one also demands an inflexible pre-calibration process that needs to be repeated even if a slight change on the sampling rate is desired. It also has an upper bound on the under-sampling rate because of the nature of the non-uniformity of the wavenumber in SD OCT.

In [9], it is shown that the under-sampled nonlinear wavenumber sampling can be used directly in the CS SD OCT by taking the modified non-uniform discrete Fourier transform (MNUDFT) matrix as the sensing matrix, instead of using UDFT/FFT. Then the input data for CS can be under-sampled directly from the spectra. The k-space grid filling and k-linear calibration are no longer needed. The MNUDFT matrix is a modification of the non-uniform discrete Fourier transform (NUDFT) matrix [10], which has a sparsity problem and cannot be used as the sensing matrix in the CS SD OCT (refer to [9] for details). The MNUDFT matrix solves this problem by making the A-scans symmetric while preserving the value of the desired half. It is also shown that CS SD OCT with the under-sampled nonlinear wavenumber sampling has better sensitivity roll-off, smaller background noise, and less side-lobes compared to that using the linear wavenumber sampling. However, CS with the MNUDFT matrix is extremely time-consuming if the matrix multiplication is used in the iterative reconstruction algorithm, since the computation complexity of the matrix multiplication is $O(N \times M)$ where M and N ($M < N$) are the row and column number of the under-sampled MNUDFT matrix, respectively.

The gridding-based non-uniform fast Fourier transform (NUFFT) [11] has been shown to be a good approximation to NUDFT and offers better performance than the interpolation+FFT method [12–15]. NUFFT reduces the computation complexity to $\sim O(NIog(N))$ and has already been adapted for the signal processing of regular SD OCT [12–15]. As shown in Section 2.4, the sparsity problem for the NUDFT matrix that excludes its usage in the CS SD OCT does not exist for CS with NUFFT. NUFFT is indeed a faster implementation that approximates MNUDFT when used in the CS SD OCT. By adding the zero-padding/under-sampling steps to the CS reconstruction procedure, NUFFT can be used in the CS SD OCT with the under-sampled nonlinear wavenumber sampling.

Compared to the reconstruction of regular OCT image, CS SD OCT takes significantly more time which hinders its applications in the surgical conditions. CS SD OCT is usually solved by an iterative algorithm that requires numerous matrix-vector computation, which is computationally complex and time-consuming if solved on the CPU based systems. However, such computation is ideal for parallel processing which can significantly reduce the computation time. As a highly effective massively parallel processor, a graphics processing unit (GPU) has been widely used in processing the regular SD OCT signals [14, 16–18] which achieves a speed of more than 320k Ascans/s with A-scan size 2048. GPU has also been adapted for the acceleration of CS reconstruction of various signals [19–21]. GPU-accelerated FFT-based CS SD OCT is studied in [22], which demonstrates real-time B-mode imaging at excess of 70 fps with

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

frame size 2048(axial)×1000(lateral) on a conventional desktop computer architecture having three GPUs.

In this work, we implemented the GPU-accelerated NUFFT-based CS SD OCT on the same triple-GPUs architecture as [22] and demonstrated that real-time B-mode imaging can be reconstructed at more than 30 fps with frame size 2048(axial)×1000(lateral). Both Kaiser-Bessel (KB) and Gaussian function are tested independently as the convolution kernel for NUFFT with different over-sampling ratios and kernel widths. The computation speed is evaluated in both the pre-computed mode and the on-the-fly mode. We also implement the GPU-accelerated MNUDFT matrix-based CS SD OCT as a comparison for speed and image quality. The experimental results show that GPU-NUFFT-CS is a good approximation to the GPU-MNUDFT-CS in terms of the image quality while providing more than 5 times speed enhancement. Compared to GPU-FFT-CS, GPU-NUFFT-CS has improved sensitivity roll-off, better signal-to-noise ratio, and less side-lobes while eliminating the need for the cumbersome k-space grid filling and the pre-calibration process.

The organization of the rest of this paper is as follows: Section 2 demonstrates the mathematical background of MNUDFT-CS and NUFFT-CS with both KB function and Gaussian function; implementation of GPU-MNUDFT-CS and GPU-NUFFT-CS is described in Section 3; the sensitivity roll-off and speed of different implementations are compared in Section 4; in vivo SD OCT imaging with GPU-NUFFT-CS are shown in Section 5, with the conclusion in Section 6.

2. Theory

2.1. CS OCT with under-sampled nonlinear wavenumber sampling

CS OCT theory states that A-scan images (denoted as $\mathbf{x}$) can be reconstructed with high accuracy from the under-sampled nonlinear wavenumber sampling (denoted by $\mathbf{y}_u$) by solving the following unconstrained nonlinear convex optimization problem:

$$\underset{\mathbf{x}}{\text{minimize}}\ \frac{1}{2}\|\mathbf{F}_u\mathbf{x}-\mathbf{y}_u\|_2^2+\tau\|\Phi\mathbf{x}\|_1 \tag{1}$$

Φ is the sparsifying operator which is chosen as the identity matrix in our work because OCT signal is usually sparse enough in spatial domain [3, 4, 22]. τ controls the sparsity of the reconstructed A-scan. The notation $\|\bullet\|_1$ is the l_1-norm and $\|\bullet\|_2$ is the l_2-norm. $\mathbf{F}_u$ is the under-sampled transformation matrix.

2.2. MNUDFT-CS

In regular SD OCT, the non-uniform discrete Fourier transform (NUDFT) matrix [10] is used to transform the full-length non-linear wavenumber spectral data $\mathbf{y}$ ($\mathbf{y}_u$ is a subset of $\mathbf{y}$) to the A-scan image, $\mathbf{x}$. However, as stated in [9], NUDFT matrix cannot be used as the sensing matrix in CS SD OCT because its resulting A-scan has much lower sparsity compared to the sparsity of the A-scan obtained with FFT. The sparsity here indicates the number of coefficients that represent the signal that is close to 0. According to CS theory, it will require a much higher sampling rate than the FFT-based CS.

The modified NUDFT (MNUDFT) matrix is proposed in [9] to promote the sparsity of the resulting A-scan obtained from the nonlinear wavenumber sampling. MNUDFT preserves the first half of the displayed A-scan and makes the other half have a low sparsity and symmetric to the first half, as can be seen by comparing Figs. 1(a) and 1(b). The MNUDFT matrix is shown in Eq. (2) and can be used in CS SD OCT with the under-sampled nonlinear wavenumber sampling: $h(p,q)=\exp(i\omega_p\times q)$. $p,q\in[0,\ldots,N-1]$. N is the size of the full-length spectra. i is the imaginary unit. $\omega_p=2\pi/\Delta K\times(k_p-k_0)$. $\mathbf{k}=[k_0,k_1,\ldots,k_{N-1}]$ is the nonlinear wavenumber

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

corresponding to the full-length spectra. $\Delta K = k_{N-1} - k_0$. $h(p,q)^*$ is the conjugate of $h(p,q)$.

$$\begin{bmatrix} h(0,0) & \dots & h(0,N/2-1) & h(0,N/2) & h(0,N/2-1)^* & \dots & h(0,1)^* \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h(N-1,0) & \dots & h(N-1,N/2-1) & h(N-1,N/2) & h(N-1,N/2-1)^* & \dots & h(N-1,1)^* \end{bmatrix} \tag{2}$$

Compared to CS SD OCT using FFT and the under-sampled linear wavenumber sampling, MNUDFT-CS has better sensitivity roll-off, larger signal-to-noise ratio, and less side lobes. Also, MNUDFT-CS can more readily obtain the under-sampled nonlinear wavenumber data since the grid filling [3] or k-linear mask pre-calibration [4] required by FFT-CS is no longer needed.

2.3. NUFFT

MNUDFT-CS enables accurate reconstruction of the SD OCT image from the under-sampled nonlinear wavenumber sampling. It is extremely time-consuming, however, if the matrix multiplication is used in the reconstruction process, since the computation complexity of the matrix multiplication is $O(N \times M)$ where M and N are the row and column number of the under-sampled MNUDFT matrix, respectively. This will limit its application for clinical use that typically requires either real-time or immediate image reconstruction.

The gridding-based NUFFT is a fast algorithm that approximates NUDFT [11] and has been shown to offer better performance than the interpolation+FFT method [12–15]. Two different kinds of NUFFT will be discussed in this paper. Following the definition in [11], they are called type-1 and type-2 NUFFT. The type-1 NUFFT transforms the nonlinear wavenumber data to a linear spatial representation while type-2 NUFFT transforms the linear spatial data to a nonlinear wavenumber representation. They are a mutually inverse transformation.

The computation steps for type-1 NUFFT are summarized in the following:

(1) Compute the new coefficients on an over-sampled uniform grid by convolution with a kernel function:

$$F_r(\hat{k}_i) = \sum_n F(k_n) G(\hat{k}_i - k_n), i = 0, \dots, M_r - 1 \tag{3}$$

where $\hat{\mathbf{k}} = [\hat{k}_0, \dots, \hat{k}_{M_r-1}]$ is the uniform grid covering the same range as $\mathbf{k}$. $F_r(\hat{k}_i)$ is the new coefficient with linear wavenumber while $F(k_n)$ is the nonlinear wavenumber data. $G(\kappa)$ is the kernel function. $M_r = R \times N$ is the size of linear wavenumber grid. R is the over-sampling ratio. Usually, Eq. (3) is evaluated within a window defined by the kernel width W for each $\hat{k}_i$: the range of n satisfies $|k_n - \hat{k}_i| / \delta\hat{k} \le W/2$, $\delta\hat{k} = \Delta K / M_r$.

(2) Apply regular FFT to the new coefficients $F_r(\hat{\mathbf{k}})$.

(3) Deconvolve the resulting signal by a division of the Fourier transform of the kernel function:

$$\tilde{f}_r(x_m) = f_r(x_m) / g(x_m), m = 0, \dots, M_r - 1 \tag{4}$$

where $f_r(\mathbf{x})$ and $g(\mathbf{x})$ are the Fourier transform of $F_r(\hat{\mathbf{k}})$ and $G(\kappa)$, respectively.

(4) Truncate the signal to the original size.

The computation steps for type-2 NUFFT are also summarized:

(1) Zero-pad the signal to the length of the over-sampling grid.

(2) Deconvolve the signal by a division of the Fourier transform of the kernel function (Eq. (4)).

(3) Apply regular FFT to the resulting signal.

(4) Compute the coefficients with nonlinear wavenumber using the kernel convolution:

$$F(k_n) = \sum_i F_r(\hat{k}_i) G(k_n - \hat{k}_i), n = 0, \dots, N - 1 \tag{5}$$

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

The same as Eq. (3), Eq. (5) is usually evaluated within a window with the same width W for each k_n.

Both the Gaussian function [13,14] and Kaiser-Bessel (KB) function [12,15] have been used as the convolution kernel in the processing of the regular SD OCT signal with NUFFT. KB kernel is found to be a better choice in terms of image quality [12, 15, 23] while Gaussian kernel offers a better accuracy-speed trade-off [15]. The definition of the Gaussian kernel and its Fourier transform are [11, 13–15, 23]:

$$G(\kappa) = \exp(-\frac{\kappa^2}{4\theta}) \tag{6}$$

$$g(x_m) = \sqrt{2\theta}\exp(-x_m^2\theta) \tag{7}$$

The optimal value for θ is [11]:

$$\theta = \frac{1}{N^2}\frac{\pi}{R(R-0.5)}\frac{W}{2} \tag{8}$$

The definition of KB kernel and its Fourier transform are listed as follows [15, 23, 24]:

$$G(\kappa) = I_0\left(\beta\sqrt{1-(2\kappa/W)^2}\right)/W \tag{9}$$

where $I_0(\kappa)$ is the zero-order modified Bessel function of the first kind.

$$g(x_m) = \frac{\sin\left(\sqrt{(m\pi W/M_r)^2-\beta^2}\right)}{\sqrt{(m\pi W/M_r)^2-\beta^2}} \tag{10}$$

The optimal value for β is given by [23, 24]:

$$\beta = \pi\sqrt{\frac{W^2}{R^2}(R-0.5)^2 - 0.8} \tag{11}$$

The over-sampling ratio R and kernel width W influence not only the computation complexity or the speed, but also the accuracy of the reconstruction, regardless of the kernel used. Increasing R and W will reduce the truncation error introduced by using a finite width window [15]. It has been reported [23, 24] that R between 1 and 2 is allowed, which shows the potential to reduce the computation time while retaining the image quality. In this paper, NUFFT-CS is tested independently using both the Gaussian and KB function with different sampling ratios (R=1.5 and 2) and kernel widths (W=3 and 5).

2.4. NUFFT-CS

To incorporate NUFFT into CS reconstruction, two issues need to be resolved: the first one is that CS demands the under-sampled version of NUFFT. Both the input and output signals of type-1 and type-2 NUFFT have the length N. Thus, during the CS reconstruction, we cannot apply type-1 NUFFT to $\mathbf{y}_u$ for $\mathbf{F}_u^T\mathbf{y}_u$ or type-2 NUFFT to $\mathbf{x}$ for $\mathbf{F}_u\mathbf{x}$. $\mathbf{F}_u^T$ is the adjoint matrix of $\mathbf{F}_u$. Instead of computing the under-sampled versions of type-1 and type-2 NUFFT, we use a similar procedure as in [22] to solve this. Denote $\mathbf{I}$ as the sampling mask corresponding to $\mathbf{y}_u$. $\mathbf{F}_u\mathbf{x}$ is computed in two steps: (1) apply the type -2 NUFFT to $\mathbf{x}$, denote the result as $\mathbf{t}$; (2) under-sample $\mathbf{t}$ with $\mathbf{I}$. $\mathbf{F}_u^T\mathbf{y}_u$ is computed in a similar way: (1) zero-padding $\mathbf{y}_u$ to length N according to $\mathbf{I}$, denote the result as $\mathbf{y}_z$; (2) apply the type-1 NUFFT to $\mathbf{y}_z$. Clearly, this implementation does not change the result of $\mathbf{F}_u\mathbf{x}$ and $\mathbf{F}_u^T\mathbf{y}_u$. In fact, this implementation can be considered as

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

a convenient interface for incorporating any transformation into CS reconstruction. No under-sampled version of that transformation needs to be computed. It is especially useful in some cases like NUFFT: the under-sampled version of the transformation may not exist or its closed form is difficult to compute.

Another issue for NUFFT-CS is to check to see whether NUFFT has a similar sparsity problem as NUDFT [9]. NUFFT is a good approximation to the NUDFT. If its resulting A-scan has a similar A-scan sparsity as the one from NUDFT, NUFFT cannot be applied to CS reconstruction. We compared the A-scans obtained by applying NUDFT, MNUDFT, type-1 NUFFT to a full-length nonlinear wavenumber data (belonging to a multi-layer polymer phantom scanning) and FFT to the full-length linear wavenumber data (obtained from the full-length nonlinear wavenumber data using cubic interpolation). The results are shown in Fig. 1.

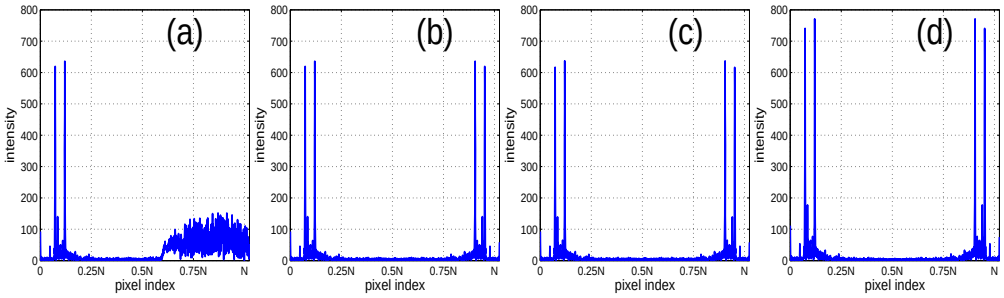

Fig. 1. Sparsity comparison of A-scans by applying (a) NUDFT, (b) MNUDFT, (c) type-1 NUFFT to the full-length nonlinear wavenumber spectral data, and (d) FFT to the full-length linear wavenumber spectral data.

The A-scan obtained from NUFFT has similar sparsity as the A-scans from MNUDFT and FFT. All of them are much higher than the sparsity of the A-scan using NUDFT. In SD OCT, only the left half of the A-scan is displayed. The displayed halves of Figs. 1(a)–1(c) are very close. However, for the right half, the NUFFT result has much higher sparsity than NUDFT and is symmetric to the left half, which is the same as MNUDFT. The reason is that NUFFT is based on FFT. If the input data is in real number, after the computation step (1) and (2), the data is still real. Symmetric structure signal will be achieved when applying FFT to real number vector, as described in step (3). Step (4) simply removes some redundant data; thus, its resulting signal will be symmetric. There is no sparsity problem for NUFFT-CS. In fact, it is easy to figure out that NUFFT is a good approximation to MNUDFT in this application, not NUDFT. Although a special case is used in Fig. 1, the same phenomenon will happen when applying the above four transforms to other SD OCT data.

Thus by adding the zero-padding/under-sampling steps to the CS procedure, NUFFT can be used in CS SD OCT with the under-sampled nonlinear wavenumber sampling.

3. GPU-MNUDFT-CS and GPU-NUFFT-CS

In this section, we describe the implementation of the GPU-accelerated NUFFT-based CS reconstruction, as well as the GPU-accelerated MNUDFT-based CS reconstruction for the purpose of comparing the image quality and speed. All the programs are implemented through NVIDIA's Compute Unified Device Architecture (CUDA) technology [25].

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

3.1. SpaRSA

The selected CS reconstruction algorithm is SpaRSA [26]. It tries to solve the CS problem in Eq. (1) through an iterative strategy in which the new iterate $\mathbf{x}^{p+1}$ is obtained by solving an optimization subproblem with the current iterate $\mathbf{x}^p$:

$$\mathbf{x}^{p+1} = \min_{\mathbf{z}} \frac{1}{2}\|\mathbf{z}-\mathbf{u}^p\|_2^2 + \frac{\tau}{\alpha^p}\|\mathbf{z}\|_1 \tag{12}$$

for some $\alpha^p > 0$. $\mathbf{u}^p = \mathbf{x}^p - (1/\alpha^p)\mathbf{F}_u^T(\mathbf{F}_u\mathbf{x}^p - \mathbf{y}_u)$. Equation (12) can be viewed as the quadratic separable approximation of Eq. (1) [26] and can be solved separablely in the component of $\mathbf{z}$:

$$x_i^{p+1} = \min_{z_i} \frac{1}{2}(z_i - u_i^p)^2 + \frac{\tau}{\alpha^p}|z_i| = \text{soft}(u_i^p, \frac{\tau}{\alpha^p}) \tag{13}$$

for $i \in [0,1,\ldots,N-1]$. soft(x,a)$\hat{=}$max$\{|x|-a,0\} \times x/$max$\{|x|-a,0\}$ is the complex soft-threshold function. α^p is chosen as the approximation of the Hessian $\mathbf{F}_u^T\mathbf{F}_u$ [26,27]:

$$\alpha^p = \|\mathbf{F}_u\mathbf{s}^p\|_2^2/\|\mathbf{s}^p\|_2^2 \tag{4}$$

where $\mathbf{s}^p = \mathbf{x}^p - \mathbf{x}^{p-1}$.

The reconstruction procedure for a B-scan using one GPU is listed in the next page. L is the number of A-scans in one B-scan. The reconstructions of A-scans are independent in CS SD OCT with SpaRSA. To obtain a bigger acceleration, we reconstruct the A-scans in one B-scan together. It can be easily figured out that the computation procedure works for parallel reconstruction of arbitrary number of A-scans. Higher acceleration can be achieved with a bigger L. L is chosen as the number of A-scans in one B-scan to smooth the B-mode display.

The stopping criterion in our implementation is different from that used in [26]. The reconstruction stops when the predefined stopping iteration number η is achieved. As stated in [22], this modification will smooth the reconstruction rate since the reconstruction time is determined by η and is independent of the input data as well as other parameters. Choosing η is a trade-off between the computation speed and image quality. Usually, setting η to 10-20 is enough for a high-quality reconstruction.

τ determines the sparsity of the A-scan. Bigger τ is desired for SpaRSA for a high-quality reconstruction [20, 26]; however, it will also lead to more loss of the low-contrast features. Thus, τ should be chosen according to the image quality. How to optimize τ is still an open field in the CS reconstruction of real signals. Please refer to [22] for the influence of η and τ on the image quality and computation time.

The l_2-norm operator in the CUBLAS library [28] does not provide a batched operation that computes the l_2-norm of multiple A-scans simultaneously. We implement an l_2-norm operator that can compute the l_2-norm of arbitrary number of A-scans simultaneously. It is based on the tree-based reduction in shared memory, as advocated in [29]. Our implementation is more than 100 times faster than computing the l_2-norm of multiple A-scans sequentially using the l_2-norm operator in CUBLAS.

The thread number in our implementation is 128. It is tuned using the NVIDIA's Visual Profiler [30] for the highest reconstruction speed. The block number is chosen based on the size of the data. For example, for the operation in line 15 of the computation procedure, the block size will be the smallest integer bigger than $2048\times1000/128$. Here, 2048 is the size of the A-scan; 1000 is the number of A-scans in one B-scan. Thus, the block number changes for different operations in our implementation.

GPU-accelerated CS reconstruction based on different transformations can be realized by replacing $\mathbf{F}_u\mathbf{x}$ and $\mathbf{F}_u^T\mathbf{r}$ in the computation procedure with the instance of sensing transformation. There is no difference between the result obtained using GPU-accelerated CS reconstruction and that using the CS reconstruction with MATLAB or C++.

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

Computation procedure of SpaRSA for a B-scan on one GPU

Input:

$\mathbf{y}_u^1,\ldots,\mathbf{y}_u^L$	# under-sampled data for each A-scan
$\mathbf{I}$ or $\mathbf{F}_u$	# sampling mask or the sensing matrix
η	# stopping iteration number
τ	# CS regulation parameter

Output:

$(\mathbf{x}^1)^\eta,\ldots,(\mathbf{x}^L)^\eta$	# $(\mathbf{x}^i)^j$ is the j-th iterate of the i-th A-scan

Initialization: set $(\alpha^1)^0,\ldots,(\alpha^L)^0$, $(\mathbf{x}^1)^0,\ldots,(\mathbf{x}^L)^0$
copy $\mathbf{y}_u^1,\ldots,\mathbf{y}_u^L$, $\mathbf{I}$ or $\mathbf{F}_u$ to GPU
$p \leftarrow 0$
for all $j \in [1,\ldots,L]$ **do in parallel**
    $(\mathbf{r}^j)^0 = \mathbf{F}_u(\mathbf{x}^j)^0 - \mathbf{y}_u^j$
end for
while $p \neq \eta$ **do**
    for all $j \in [1,\ldots,L]$ **do in parallel**
        $(\mathbf{u}^j)^p = (\mathbf{x}^j)^p - (1/(\alpha^j)^p)\mathbf{F}_u^T(\mathbf{r}^j)^p$
    end for
    for all $i \in [0,\ldots,N-1]$ **and** $j \in [1,\ldots L]$ **do in parallel**
        $(x_i^j)^{p+1} = \text{soft}\Big((u_i^j)^p, \tau/(\alpha^j)^p\Big)$ # x_i^j is the i-th component of $\mathbf{x}^j$
    end for
    for all $j \in [1,\ldots,L]$ **do in parallel**
        $(\mathbf{s}^j)^{p+1} = (\mathbf{x}^j)^{p+1} - (\mathbf{x}^j)^p$
        $(\mathbf{r}^j)^{p+1} = \mathbf{F}_u(\mathbf{x}^j)^{p+1} - \mathbf{y}_u^j$
        $(\alpha^j)^{p+1} = \|(\mathbf{r}^j)^{p+1} - (\mathbf{r}^j)^p\|_2 / \|(\mathbf{s}^j)^{p+1}\|_2$
    end for
    $p \leftarrow p+1$
end while
Finish: copy $(\mathbf{x}^1)^\eta,\ldots,(\mathbf{x}^L)^\eta$ to the host memory

3.2. GPU-NUFFT-CS

The implementation of $\mathbf{F}_u\mathbf{x}$ and $\mathbf{F}_u^T\mathbf{r}$ with NUFFT has been discussed in Section 2.4. The CUFFT library [31] is used for the batched FFT transform. Another benefit of this implementation is that instead of the whole sensing matrix only the one-dimensional binary sampling mask $\mathbf{I}$ is stored in the GPU memory, which is more compact.

For SD OCT, the nonlinear wavenumber $\mathbf{k}$ is static. The value of $G(\kappa)$ and $g(\mathbf{x})$ can be pre-computed. Also, the indices of the uniform grid (range of n) for each $\hat{k}_i$ in Eq. (3) and the indices of the nonlinear wavenumber (range of i) for each k_n in Eq. (5) can be obtained before-hand. The computation time then is independent of the choice of kernel function.

Although our program is tested on the reconstruction of SD OCT signal, it can also be used in the swept source OCT (SS OCT) that has nonlinear wavenumber. In the case that frequency jitters or low repeatability of frequency scanning happens in SS OCT, pre-computation of the NUFFT parameters are not feasible and will degrade the reconstruction accuracy if they are used in the reconstruction. Instead, the NUFFT parameters should be computed on-the-fly. In this work, the GPU-NUFFT-CS with the Gaussian kernel is implemented in both the pre-computed mode and on-the-fly mode while that with the KB kernel is only implemented in the pre-computed mode because the zero-order modified Bessel function of the first kind has not

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

been incorporated to CUDA. The results obtained in the pre-computed mode and on-the-fly mode are the same for GPU-NUFFT-CS with the Gaussian kernel.

3.3. GPU-MNUDFT-CS

The implementation of $\mathbf{F}_u\mathbf{x}$ and $\mathbf{F}_u^T\mathbf{r}$ with the MNUDFT matrix is straightforward. MNUDFT matrix should be under-sampled according to the sampling mask first before being transferred to the GPU. The matrix multiplication operator in CUBLAS [28] is used in the program.

4. Sensitivity roll-off and speed comparison

Our methods are evaluated using the spectral data from an SD-OCT system. A superluminescent laser diode (SLED) with a center wavelength of 845 nm and a bandwidth of 105 nm is used as the light source. The spectrometer contains a 12-bit CMOS line scan camera (EM4, e2v, USA) with 2048 pixels at 70 kHz line rate. The beam scanning speed in our experiment is 70mm/s. The experimental axial resolution of the system is 4.0 μm in air and the transversal resolution is approximately 12 μm. The sensitivity of our system is 92.5dB, which is tested by placing a mirror as a sample at the position of 80μm from the zero-delay (corresponding to the first peaks in Figs. 2(a)-2(f)) and a ND filter (-35.5dB) in front of the mirror while the reference level was set at the level for shot-noise limited detection. It is computed using the method in [32]. For all experiments, the power of light incidents on the samples is 3.7mW, which is conformity with the American National Standard Institute (ANSI) standard for safe use of lasers [33] on the experimental samples in our work.

The experimental data in this section is measured on a workstation with an Intel Xeon CPU (3.46 GHz), 16 GB memory and a NVIDIA GeForce GTX 580 GPU with 512 stream processors, 1.5 GHz processor clock and 1.5 GB global memory. The under-sampled nonlinear wavenumber spectral data used in the CS reconstruction is obtained by sampling from the pre-generated k-space data set using a variable density sampling mask [6] and stored in the host memory of the computer. This is consistent with the desired practical application in which the under-sampled k-space data will be grabbed from the camera, transferred, and stored in the RAM of the computer. The under-sampling rate is chosen to balance the image quality and input data size.

The following processing methods are compared using the same input data ($\mathbf{y}_u$): (1) GPU-MNUDFT-CS; (2) GPU-NUFFT-CS with different kernel functions (KB and Gaussian), over-sampling ratios ($R = 1.5$ and 2) and kernel widths ($W = 3$ and 5). For comparison, the data obtained by applying NUDFT to the 100% spectra (denoted as "NUDFT" in Fig. 2) and GPU-FFT-CS [22] on the same amount of under-sampled linear wavenumber spectral data are also displayed. The same parameters τ and η are used in the CS experiments. The starting values are set as $\mathbf{x}^0 = 0$ and $\alpha^0 = 1$. These two starting values do not show obvious influence on the image quality and computation time.

4.1. Sensitivity roll-off

The point spread function (PSF) and sensitivity roll-off with different processing methods are presented in Figs. 2(a)–2(f). Here 128 A-scans each with 2048-pixel were averaged for each position. CS reconstruction uses 30% spectral data. $\tau = 3$. $\eta = 15$. As can be seen from Figs. 2(e) and 2(f), there are only subtle observation differences between the data with the same kernel but different R and W; thus, only three NUFFT-CS results are displayed: "NUFFT-CS Gaussian R2 W5", "NUFFT-CS KB R1.5 W3", and "NUFFT-CS KB R2 W5".

The NUFFT-CS results in Figs. 2(d), 2(e), and 2(f) show flatter sensitivity roll-off with a smaller background noise and side-lobes at larger image depths than the FFT-CS result

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

in Fig. 2(b). Compared to the NUDFT result on 100% spectral data in Fig. 2(a), NUFFT-CS results exhibit lower background noise, which is approximately -70dB against -60dB for NUDFT. NUFFT-CS provides a similar performance to MNUDFT-CS, as shown in Fig. 2(c). The NUFFT result using KB kernel with $R = 2$ and $W = 3$ is added to Figs. 2(g)–2(h) to illustrate the influence of W on the image quality. The comparison between PSFs at an imaging depth of 0.85 mm is presented in Fig. 2(g). The intensity of each A-scan is normalized. FFT-CS shows large side-lobes while NUFFT-CS, MNUDFT-CS and NUDFT have relatively flat background and no noticeable side-lobes. Compared to NUDFT, NUFFT-CS and MNUDFT-CS have much lower background noise. The maximum amplitude of PSFs from different methods is exhibited in Fig. 2(h). MNUDFT-CS and NUFFT-CS show a very close sensitivity roll-off to NUDFT while FFT-CS has much faster sensitivity attenuation as the image depth increases. Figure 2(i) shows the SNR versus axial position for various processing methods with the SNR definition in [23]. All CS methods have a very close SNR at the imaging depth less than 0.8 mm, while the SNR of FFT-CS becomes lower as the image depth increases. Compared to NUDFT, CS methods usually show better SNR since CS is effective in reducing noise [3, 5].

All NUFFT-CS implementations reconstruct the peak of PSFs successfully. NUFFT-CS with KB kernel shows better denoising effect compared to the one with Gaussian kernel. Bigger over-sampling ratio R and kernel width W result in better suppression of the background noise level. Figures 2(g) and 2(i) show that the "NUFFT CS KB R2 W5" result is very close to the result of "NUFFT CS KB R2 W3". Both of them have a lower background noise than "NUFFT CS KB R1.5 W3" result. R shows bigger influence on the image quality than W.

4.2. Speed comparison

The averaged computation time for a single A-scan using GPU-NUFFT-CS is listed in Table 1. The time is averaged based on 100 runs of B-scan reconstruction. The size of B-scan is 2048(axial)$\times$1000(lateral) pixels. As stated above, the computation time of our implementation is determined by the stopping iteration number which is set as $\eta = 15$. The pre-computed mode is tested for both the Gaussian and KB kernel while only the Gaussian kernel is evaluated in the on-the-fly mode since the modified Bessel function has not been incorporated into CUDA. In both modes, the kernel width W has more influence on the processing speed compared to the over-sampling ratio R. For comparison, the averaged computation time of GPU-MNUDFT-CS and GPU-FFT-CS is 626.67 μs and 47.2 μs, respectively.

Table 1. Computation time (μs) for an A-scan

	pre-computed mode Gaussian and KB				on-the-fly mode Gaussian			
NUFFT-CS	R=1.5		R=2		R=1.5		R=2	
	W=3	W=5	W=3	W=5	W=3	W=5	W=3	W=5
	99.17	122.27	100.03	122.87	154.47	199.92	167.15	223.84

The time for transferring the under-sampled spectral data from host to device and the image result from device to host is contained in the time listed in Table 1. In fact, the time listed in Table 1 is the integration of the time for data transferring, CS reconstruction, and data conversion. We measured the data transferring time by running experiments using 50% spectral data. The time required to transfer the under-sampled spectral data of a B-scan from host to device is 346.93μs where the size of the under-sampled spectral data of one B-scan is 1.953MB (1024$\times$1000$\times$2 Bytes) with data type being unsigned short. The time required to transfer the reconstructed image from device to host is 314.59μs where the size of the image is also 1.953MB

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

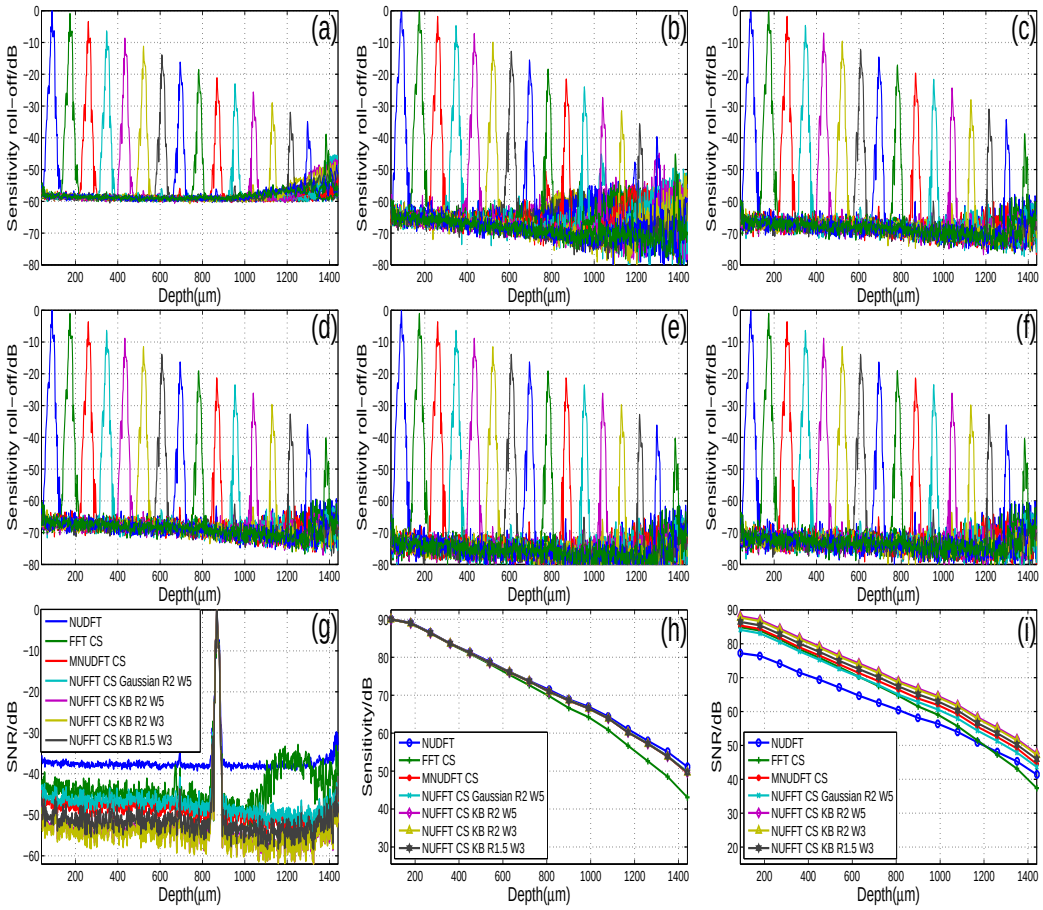

Fig. 2. Sensitivity roll-off of different processing methods: (a) NUDFT on 100% data; (b) FFT-CS; (c) MNUDFT-CS; (d) NUFFT-CS with Gaussian kernel (R=2, W=5); (e) NUFFT-CS with KB kernel (R=2, W=5); (f) NUFFT-CS with KB kernel (R=1.5, W=3). (g) comparison of PSFs at a certain image depth using different processing methods; (h) maximum amplitude of PSFs using different processing methods; (i) SNR versus image depth for different processing methods.

(1024×1000×2 Bytes) with data type being unsigned short, since only half of the B-scan needs to be displayed. The transfer time introduces a negligible latency to the system and does not affect the image result.

The comparisons on the sensitivity roll-off and computation time show that NUFFT-CS has similar performance as the MNUDFT-CS while providing a speed enhancement of more than 5 times when conducted in the pre-computed mode. Compared to FFT-CS, it shows much lower background noise and less side-lobes at the large image depths which also indicates a higher local SNR at deeper axial positions. Also, it employs a more flexible and time-saving sampling pattern than FFT-CS. Compared to NUDFT with 100% data, NUFFT-CS has negligible difference on the sensitivity roll-off but exhibits better SNR at any image depth using only a portion of the spectral data.

In the GPU-NUFFT-CS SD OCT, KB kernel is a better choice which offers better image quality. *R* has bigger influence on the image quality while *W* impacts more on the program

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

speed. KB kernel with $R = 2$ and $W = 3$ would be a good choice for GPU-NUFFT-CS SD OCT.

5. In vivo SD OCT imaging

In vivo SD OCT imaging of biological samples was conducted to confirm the performance of GPU-NUFFT-CS. The experiments were executed on the same triple-GPU architecture as in [22].

The SD OCT images of an orange and human skin obtained with GPU-NUFFT-CS are presented in Figs. 3(b) and 3(d), respectively. The sampling rate was 40%. $\tau = 3.0$. $\eta = 10$. The KB kernel with over-sampling ratio 2 and kernel width 3 is used in NUFFT. The reconstruction results by applying NUDFT to 100% data are shown in Figs. 3(a) and 3(c), respectively as a reference. The images of the same sample are shown in the same dynamic range. Figures 3(a) and 3(b) are the average of 100 frames while Figs. 3(c) and 3(d) are single frame results. The NUFFT-CS achieves an accurate reconstruction which is very close to the reference image. Also, smaller background noise is observed in the NUFFT-CS results. The peak signal-to-noise ratio (PSNR) is computed for Figs. 3(a)–3(d) for a quantitative comparison, with the definition $\text{PSNR} = 10\log_{10}(\max^2(f(\mathbf{x}))/\text{var})$. $f(\mathbf{x})$ is the intensity of the B-scan while var is the standard deviation of the selected background regions outlined by the white dash rectangles in Fig. 3. The comparison of PSNR is 73.10dB to 79.92dB for Fig. 3(a) to 3(b), respectively and 69.48dB to 76.85dB for Fig. 3(c) to 3(d), respectively. GPU-NUFFT-CS shows an average of 7dB improvement on the SNR. CS tries to maximize the sparsity of the signal in the transformed domain while preserving the data fidelity in the measurement domain. Higher sparsity indicates that a larger percentage of the signal coefficients are zero or close to zero. Thus CS has been shown to be good at reducing background noises [3, 5, 8, 34]. For the same reason, CS reconstruction also results in loses of low-intensity features of the image.

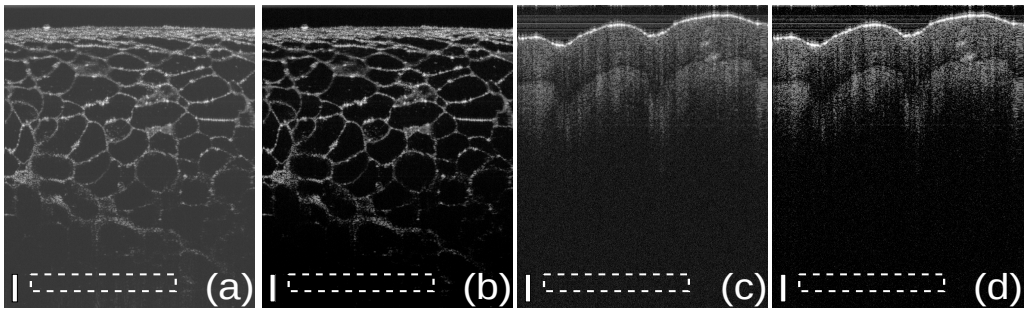

Fig. 3. B-scans of an orange: (a) original image; (b) NUFFT-CS reconstruction result see (Media 1 for real-time imaging display), and human skin: (c) original image; (d) NUFFT-CS reconstruction result (see Media 2 for real-time imaging display). The scale bars represent 100 μm. The image size is 900(axial)×950(lateral).

The real-time B-mode imaging display for both samples is screen captured at Media 1 and Media 2 with frame rate 33.5 fps. The original B-scan size is 2048(axial)×1000(lateral) pixels and is rescaled to 512(axial)×500(lateral) pixels to accommodate the monitor display. The video shows the logarithm of the rescaled B-scan and no further processing was applied. These two results show that the speed of our program is independent of the imaging object which enables the development of a system that incorporates the NUFFT-CS technique. The sampling rate is 40%. $\tau = 3.0$. $\eta = 10$. For comparison, GPU-MNUDFT-CS on the same architecture has a speed of 6.5 fps. KB kernel with over-sampling ratio 2 and kernel width 3 is pre-computed

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014

for NUFFT.

6. Conclusion

In this work, we proposed CS reconstruction of SD OCT image based on NUFFT and non-linear wavenumber under-sampling, which was successfully implemented on the GPU architecture. GPU-NUFFT-CS provides a close approximation to the GPU-MNUDFT-CS in terms of imaging quality but can achieve more than five times speed enhancement. Compared to the GPU-FFT-CS, GPU-NUFFT-CS has better sensitivity roll-off, higher signal-to-noise ratio, and smaller side-lobes.

Acknowledgments

This work was supported in part by an NIH grant 1R01EY021540-01A1.

#209081 - $15.00 USD Received 27 Mar 2014; revised 2 May 2014; accepted 3 Jun 2014; published 10 Jun 2014