Scanning-mode 2D Acoustic Radiation Force Impulse (s2D-ARFI) Imaging Based on GPU Acceleration

Congzhi Wang, *IEEE Member*, Bo Zeng, Weibao Qiu, *IEEE Member*, Hairong Zheng, *IEEE Member*

Abstract—**Acoustic radiation force impulse (ARFI) technique is a quantitative method for tissue stiffness assessment. It has been proved to be less operator dependent than the quasi-static elastography, and has more simple hardware architecture than the supersonic shearwave imaging (SSI) technique, which make it easier to be miniaturized for some special clinical applications. However, unlike the SSI, ARFI cannot provide real-time 2D images of tissue stiffness distribution mainly due to its data-intensive and time-consuming algorithms. In this study, the algorithms of ARFI were modified and improved to fit for the parallel computation on graphics processing unit (GPU), and the quasi-real-time scanning-mode 2D ARFI images (s2D-ARFI) were implemented on a self-developed compact system. High ratio of the time consumptions between the algorithms using CPU and using GPU has been verified, and it was also proved that there was no distinct difference between the stiffness images obtained by these two methods. The s2D-ARFI provides us an additional choice for quantitatively imaging the tissue stiffness, and has a potential to be miniaturized and used in the emergency treatments in field first-aid and the donor evaluation for organ transplantation.**

I. Introduction

Acoustic radiation force impulse (ARFI) method is also called as point shear-wave elastography (pSWE) [1], which performs a single-point quantitative tissue stiffness measurement and the result is displayed as a small color box superimposed on the B-mode image. In this method, shear wave is generated by the acoustic radiation force and its propagation is tracked by the pulse-echo ultrasound. Then the shear wave velocity is measured in a region of interest (ROI) and converted to Young's modulus [2]. Commercial systems by Siemens and Philips have used this technique. Compared with quasi-static elastography, ARFI is quantitative and less dependent on the operator's experience. However, it cannot provide two-dimensional (2D) images of shear modulus (or Young's modulus) in the field of view, like the supersonic shear-wave imaging (SSI) can do. SSI can achieve a high frame rate of up to 20 kHz by transmitting a plane wave and acquiring echo signals simultaneously from all transducer elements. This makes it have the ability to follow the shear-waves in a 2D field in almost real time. However, this also requires SSI to use more complex hardware architecture and have more difficulty to be miniaturized than ARFI. In fact, compact and portable ultrasound system with 2D stiffness imaging function is demanded in some special clinical applications, such as the emergency treatments in field first-aid and the donor evaluation for organ transplantation, whereas currently there is no device that can meet this requirement. ARFI system has better potential to be such a kind of device than SSI, if the 2D stiffness imaging based on it can be implemented. To fit this gap,, the time-consuming algorithms of ARFI must be substantially accelerated.

Graphics processing units (GPU) has been proved as a good surrogate of CPU for data-intensive computing. Its powerful computing capability comes from the architecture of graphics card containing a large number of processing cores which can work in parallel. In recent decade, GPU has been widely used in medical imaging. Satisfying results have been obtained when it was applied in quasi-static elastography [3][4]. For ARFI, algorithm for small displacement estimation based on Loupas method has been migrated onto GPU [5]. However, when using this method, the raw radio-frequency (RF) signals should be first demodulated into in-phase part *I* and quadrature-phase part *Q*. This procedure will increase the complexity of the system, regardless of achieving it by hardware or by software. Moreover, the *I/Q* data are always down-sampled to reduce the data intensity, this brings more jitters into the results than using the raw RF data, especially when the signal-to-noise ratio (SNR) is small [6]. In addition, besides the most time-consuming part, tissue displacement estimation, there are still some other parts of ARFI algorithms that should be considered to be accelerated, such as the cubic spline interpolation and the shear wave velocity determination.

In this study, complete ARFI algorithms were first implemented on CPU, and then were redesigned and migrated to GPU. Finally, both of them were integrated with our self-developed scanning-mode 2D ARFI (s2D-ARFI) system and their time consumptions were compared. A cross-correlation method based on analytic signal was used to assess the tissue displacements; a cyclic reduction (CR) method was used to calculate the cubic spline interpolation; and a time-of-flight (TOF) method based on Radon transformation was used to determine the shear wave velocity. Both of these methods have been confirmed to be able to deliver good performance on relatively noisy signals [7-9], and in this study they were also proved to be very suitable for parallel processing on GPU. The ratio of the time consumptions between using CPU and using GPU was calculated, and the quality of the stiffness images obtained on the tissue mimicking elastic phantoms was also compared.

Resrach supported by the National Natural Science Foundation of China (Grant No. 81027006, 61302038 and 61302039).

CZ. Wang, B. Zeng, WB. Qiu and HR. Zheng are all with the Paul C. Lauterbur Research Center for Biomedical Imaging, Institute of Biomedical and Health Engineering, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. (Corresponding author: CZ. Wang; phone: +86-755-86392264; fax: +86-755-86392299; e-mail: cz.wang@siat.ac.cn; the other authors' emails: bo.zeng@siat.ac.cn; wb.qiu@siat.ac.cn; hr.zheng@siat.ac.cn).

978-1-4244-7929-0/14/$26.00 ©2014 IEEE

II. Methods

A. ARFI algorithms on CPU

Tissue displacement estimation: This part contains two stages, coarse and sub-sample estimation, corresponding to the integral and fractional parts of the sampling periods. The accuracy of tissue displacement estimation mainly depends on the sub-sample part. Using raw RF data, the analytic signal based cross-correlation method can achieve an accurate estimation of the sub-sample part by calculating the phase shift of the maximal cross-correlation coefficient to the zero-crossing point, which is the exact position representing the relative displacement [7]. The accumulated displacement of tissue can be cumulatively summed up from these relative displacements of adjacent frames.

Cubic spline interpolation: To improve the temporal resolution of the tissue displacement signal and to achieve a more accurate estimation of the shear wave velocity, the signal should be first interpolated using a cubic spline method, which has continuous one-order and second-order derivatives, making the interpolated points more approximate to the signals in the real world. The kernel of the interpolation algorithm is solving the tridiagonal systems and the traditional lower-upper decomposition (LU) method was used.

Shear wave velocity estimation: TOF is the conventional method for shear wave velocity estimation in ARFI. However, it is vulnerable to physiology motion, low SNR signal and spatial inhomogeneity. A more robust TOF algorithm based on Radon transformation was used to mitigate the influence of these problems. Shear wave velocity can be estimated in a time-location displacements matrix along the trajectory correspond to the maximum of Radon transformation [8].

B. Parallel ARFI algorithms on GPU

NVIDIA (CA, USA) has introduced Compute Unified Device Architecture (CUDA) to facilitate the use of GPU and CUDA has become the most easily used toolkits for programming on GPU with fewer requirements of *C* language knowledge. To harness the full power of GPU, it is necessary to redesign the algorithms mentioned above to make them fit for working in parallel. The detail procedures are described as follows.

Tissue displacement estimation: For *K* frames of data each containing *N* samples, *K*N* threads on GPU were assigned to accomplish the analytic signal construction based on Hilbert transformation. This can be easily performed using the fast Fourier transform (FFT) which can be facilitated with the built-in library in CUDA. Although theoretically this procedure should be performed on the small data sections divided for the following cross-correlation calculation, we found that when we did it on a long vector signal combined with all the *K*N* data points and reshaped the result back to a *K*N* matrix, the estimated displacement signals were not significantly influenced. It is very crucial for reducing the computation time because of that for FFT calculation on GPU, the longer the input vector, the more time can be saved.

Spectrum-domain cross-correlation method was selected since it is more computing-efficient than the time-domain method and the FFT had already been performed at last step. Because of the independence of the frame-pairs, *(K-1)*Q* threads were used to accomplish the cross-correlation, where *Q* meant that the cross-correlation coefficients was over the lag range of [-*Q*/2, *Q*/2]. Each thread was responsible to the calculation for one frame-pair at one lag value. Since the small sections divided in the data frame were partially overlapped, the calculation of subsequent section included many unnecessary redundancies. Therefore, only the appended data were loaded and their sum-of-products was added to the part reserved from the former section using a first-in-first-out (FIFO) strategy. This small trick can dramatically reduce the time-consumption in such kinds of "sliding-window" cross-correlation algorithms.

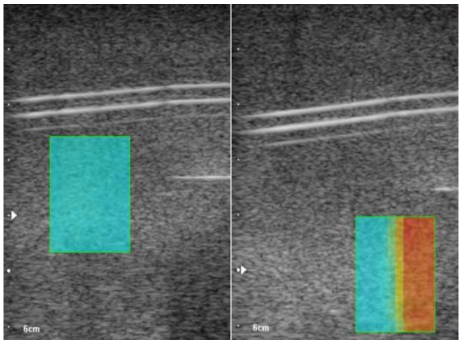
Cubic spline interpolation: Cyclic reduction (CR) method was selected because it can perform much more units of work in parallel than the LU method and is very suitable for being accelerated by GPU [9].The algorithm included two stages: first, one kernel was used to calculate the tridiagonal linear systems; second, another kernel was launched to compute the interpolation values using the results of the first kernel. Each linear system was solved by one block of threads whose dimension was half of the linear system's size. Each thread in the block was responsible for calculating the coefficients of one equation. Each thread stored its intermediate results into the block's shared memory to communicate with other threads during the cyclic reduction iteration, and wrote the final results into the global memory after the whole computation. Then the coefficients of the cubic splines were loaded into one block's shared memory and this block was used to calculate the interpolation values between two known data points. The number of the blocks was equal to the number of the intervals needed to be interpolated and the number of the threads in one block was determined by the interpolation rate.

Shear wave velocity estimation: The complexity of Radon transformation is proportional to the square of the time-dimension in the time-location matrix. The blocks with 2D indexed threads were used and the summation along one trajectory was implemented in one thread, whose *x*, *y* indexes were corresponding to the start-time point and the end-time point. However, in those threads whose *x*, *y* indexes were equal, the workflow was totally different from the other threads since the slope of the trajectory cannot be determined as infinity. In GPU, threads are generally executed in warps (32 threads indexed in the same row make up a warp), with all threads in the warp executing the same instruction at the same time. However, when different threads in a warp need to do different things, a "warp divergence" occurs. Under this situation, all threads need to execute both conditional branches and this means a potential large loss of performance. To solve this issue, diagonal indexes threads were relocated into a same warp to avoid the conditional branches in the workflow. In these blocks, the thread (*i*, *i*) was relocated into (1, *i*). After the calculation, the data positions should be rearranged in the block's shared memory. Then the results would be written into the global memory.

C. Computation speed and image quality comparison

Two independent programs with and without GPU acceleration were developed and tested on a computer with an eight-cores CPU (i7 2600, Intel, CA, USA) and a novel

GPU card (GTX Titan, NVIDIA, CA, USA), which composing of 2688 computing units and 6GB memory. CUDA 5.0 (NVIDIA, CA, USA) and Visual Studio 2010 (Microsoft, CA, USA) were selected as the software platform and a self-developed ARFI imaging system was used as the hardware platform. A rectangular ROI was first selected on the B-mode image, which was combined with several rows and columns of small boxes like in the original single-point ARFI. The size of each box was about 2mm in depth and 1.56mm in width. Then the quantitative stiffness assessments were performed in each box with a top-to-bottom and left-to-right scanning sequence. At each box position, one data set was collected including the echo lines of 4 different lateral locations and each containing 100 frames of signals with 512 data points in each frame. The lag range of cross-correlation was defined as [-40, 40]. The mean shear wave velocity in one box was measured and the Young's modulus was calculated. After the whole scanning, the ROI was pseudo-colored to represent the stiffness mapping of the tissue, as shown in Figure 1.



(a) (b)

Figure 1. Scanning-mode 2D ARFI stiffness images of a uniform elastic phantom (a) and another phantom including a hard region (b)

To compare the computation speed and the image quality of the two methods, the experiments were performed on two self-made elastic phantoms: one was uniformly soft and the other one was plugged by a hard "lesion" in the soft background. The ROI was set to be constructed by 10 * 10 small boxes. The execution time of each step and the total processing on one single data set were assessed. Furthermore, since the GPU program used a single-precision float computation but CPU used a double-precision float type, the quality of the stiffness images they generated should be evaluated. For the uniform phantom, SNR was calculated to evaluate their measurement reliability by equation (1), where S denotes the mean Young's modulus in ROI and δ_u is the standard deviation. For the "lesion" phantom, a ROI crossing the soft background and the hard region was selected and CNR (carrier-to-noise ratio) was calculated to compare their sensitivity by equation (2), where S_l and S_b correspond to the mean Young's modulus of the "lesion" and the background, and δ_l and δ_b indicate their standard variance, respectively.

$$SNR = \frac{S}{\delta_u} \tag{1}$$

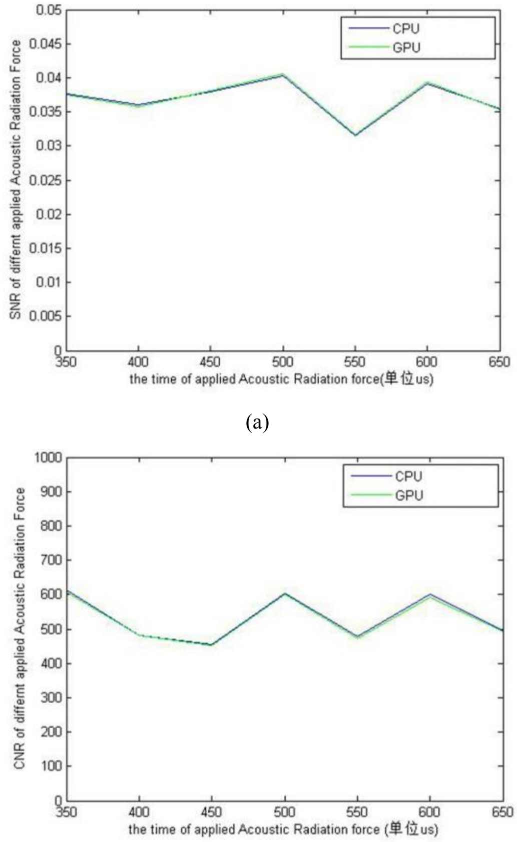
$$CNR = \frac{2(S_l^2 - S_b^2)^2}{\delta_l^2 + \delta_b^2} \tag{2}$$

III. Results

Table 1 shows the execution time of the two programs running on CPU and GPU, dealing with one same data set. Each step of the algorithms was included. For the most time-consuming parts such as cross-correlation, filtering, Radon transformation and cubic spline interpolation, large acceleration ratio has been confirmed between the two programs. The total time listed in the table also includes some additional time consumption when connecting the multi-algorithms, thus it is a little larger than the sum of the above steps.

Table I comparison results between CPU and GPU

Steps of the algorithm	**Comparison results**		
	Execution Time on CPU (ms)	***Execution Time on GPU (ms)***	***Acceleration ratio***
Data transfer	0	14.0	
Cross-correlation	2120.0	50.0	42.4
Filtering	1036.0	4.0	259.0
Radon transformation	1550.0	13.2	117.4
Cubic spline interpolation	1055.0	0.6	1758.3
Total time	5763.0	82.2	70.1



(a)

(b)

Figure 2. Comparison of SNRs and CNRs of the stiffness images obtained using GPU and CPU.

Figure 2 shows the SNRs and CNRs of the stiffness images when emitting focused ultrasound with different length pulses to generate different amplitude displacements.

The SNRs and CNRs of the images obtained using GPU are well-fitted with those obtained using CPU. This proved that little quality loss of the stiffness images occurred when the algorithms were accelerated with GPU.

Finally, a quasi-real-time s2D ARFI imaging system was implemented by the help of GPU acceleration. For a ROI constructed by 100 measurement boxes and 20mm*15mm size, the total imaging time including signal processing and displaying is about 25 to 30 seconds.

IV. Discussion

High ratio of the time consumptions between the algorithms using CPU and using GPU has been verified according to our results. In the study of Rosenweig et al. on accelerating ARFI's displacement estimation algorithm by GPU, the data sets had 52 total push locations, 80 track pulses per push, and 493 I and Q samples per track. Thus the data set they tested is about 10 times larger than ours. The total time their computation cost was 267ms. Although our result is 82.2ms for 4 tracking locations, 100 frames per location, and 512 data points per frame, our algorithm included more steps than theirs, such as the filtering and the Radon transformation. The data transfer time from the underlying hardware to the PC was neglected since this consumption is the same for both CPU and GPU algorithms.

For the cross-correlation part, our method cost 50ms in comparison to their time consumption of 61ms on ten times larger data set. However, Loupas method they used was developed based on the Doppler phase shift estimation and only needs calculating the sum-of-products once for one section of signals. On the contrary, the analytic based method we used calculates all the cross-correlation coefficients of one data section in a large lag range, [-40, 40], to search the maximal value. Therefore, the calculation amount of our method is about 80 times larger than theirs, and as a compensation, the tracking range of the tissue displacements is also 80 times larger. Loupas method not only increases the system's complexity since it needs *I/Q* decomposition, but also induces incorrect estimation due to its demodulation and down-sampling process, especially when the SNR of raw RF signals is low. In addition, Loupas method will also make mistakes when the phase wrapping occurs, like those generally observed in Doppler blood flow signals. Another reason for the poor acceleration ratio of cross-correlation algorithm is that it contains much more memory accessing operations than the other steps. Although the memory accesses has been much reduced by the utilization of FIFO strategy, uncoalesced memory accesses still exist, which reduce the acceleration efficiency.

Comparing to the previous study, the best improvement of our algorithm occurs at the cubic spline interpolation part. Rosenweig et al. adopted the method of dividing long vector into several overlapped subsets to implement parallel computing for the interpolation. However, their time cost is 46ms versus our time cost is only 0.6ms. CR method assures the maximum parallelism of solving tridiagonal linear systems by assigning two nearby equations to one thread. Although in our study CR method was limited by the dimension of the blocks and the consequential additional data communication, it still performs much better than the other traditional methods. We also tried the LU decomposition method and the time cost is 45ms. This is because the CR method needs less iterative steps than the LU method.

For the Radon transformation part, our method realized a high acceleration ratio of 117.4, but unfortunately there is no previous results can be compared to. The good parallelism of our method is achieved by assigning each thread to the computation of one trajectory to avoid synchronization time between threads, and also by the relocation of the diagonal indexes threads to avoid warp divergence.

Although the imaging speed of our s2D-ARFI system is still much slower than SSI, which can generate one stiffness image with 30ms [10], our system can provide an additional choice for quantitatively imaging the tissue stiffness. Furthermore, since ARFI has much more simple hardware architecture than SSI, it is easier to be miniaturized for some special clinical applications such as the emergency treatments in field first-aid and the donor evaluation for organ transplantation. The further improvement on the imaging speed of ARFI and the miniaturization of s2D-ARFI system will be attempted in the future studies.

References

[1] J. Bamber, D. Cosgrove, C. F. Dietrich, J. Fromageau, J. Bojunga, F. Calliada, et al., "EFSUMB Guidelines and Recommendations on the Clinical Use of Ultrasound Elastography. Part 1: Basic Principles and Technology," Ultraschall in Der Medizin, vol. 34, pp. 169-184, Apr 2013.

[2] K. Nightingale, M. Palmeri, G. Trahey, "Analysis of contrast in images generated with transient acoustic radiation force," Ultrasound in Medicine and Biology, vol. 32, pp. 61–72, 2006.

[3] Y. Xu, S. Deka, R. Righetti, "A hybrid CPU-GPU approach for real-time elastography," IEEE Trans.Ultrasonics ferreoelectronic and frequency control, vol. 58, no.12, pp. 2631-2645, 2011.

[4] B. Peng, Y. Zhan, D. Liu, "investigations of GPU-based elastography algorithms," Journal Opto-electronic Engineering, vol. 40, no. 5, pp. 97-105, 2011.

[5] S. Rosenzweig, M. Palmeri, K. Nightingale., "GPU-Based Real-time small displacement estimation with ultrasound," IEEE Trans. Ultrasonics ferreoelectronic and frequency control, vol.58, no. 2, pp. 399-405, 2011.

[6] F. Gianmarcro, J, Pinton, E. Gregg, "Rapid tracking of small displacement estimation with ultrasound," IEEE Trans.Ultrasonics ferreoelectronic and frequency control, vol. 53, no. 6, pp.1103-1107, 2006

[7] C. Simon, P. VanBaren, and E. S. Ebbini, "Two-dimensional temperature estimation using diagnostic ultrasound," IEEE Trans. Ultrasonics ferroelectrics and frequency control, vol. 45, pp. 1088-1099, Jul 1998.

[8] N. Rouze, MH. Wang, ML. Palmeri, "Robust estimation of Time-of-Flight shear wave speed using a radon sum transformation," IEEE Trans. Ultrasonics ferreoelectronic and frequency control, vol. 57, no. 12, pp. 2662-2670, 2010

[9] Y. Zhang, J. Cohen, A. Davidson, JD. Owens, "A Hybrid Method for Solving Tridiagonal Systems on the GPU," In Wen-mei W. Hwu, editor, GPU Computing Gems, volume 2, chapter 11. Morgan Kaufmann.

[10] J. Bercoff, M. Tanter, M. Fink, "Supersonic shear imaging: a new technique for soft tissue elasticity mapping," IEEE Trans. Ultrasonics ferreoelectronic and frequency control, vol. 51, no. 4, po. 396-409, 2004.