GPU-Accelerated Sub-sample Displacement Estimation Method for Real-time Ultrasound Elastography

Bo Peng[1,2], David Rosen[1], Yu Wang[1] and Jingfeng Jiang[1,*]
1.Department of Biomedical Engineering, Michigan Technological University, Houghton, MI 49931, USA
2. School of Computer Science, Southwest Petroleum University, Chengdu, Sichuan, 610500,China

1. Abstract

Ultrasound Elastography is a promising medical imaging modality that estimates mechanical properties of soft tissues. Because tissue elasticity is inferred from ultrasonically-tracked tissue displacements, a highly accurate displacement estimation method is critical. However, most advanced displacement estimation methods are computationally expensive; it is difficult to achieve real-time performance under a CPU-based computer architecture. Recently, under the framework of linear system theory of medical ultrasound systems, our group has developed an improved sub-sample displacement estimation algorithm where axial and lateral motion estimates are simultaneously performed to enhance the accuracy of motion tracking. The proposed tracking method is highly scalable because the sub-sample estimation with the region of interest requires no inter-process communication, thereby becoming a perfect candidate for the GPU-based parallelization. In this study, the proposed method has been implemented in CUDA. Compared to the original C MEX (MATLAB VERSION 2014B, Mathswork Inc., MA, USA) implementation, the GPU implementation showed a factor of 60+ acceleration while maintaining the excellent motion tracking accuracy.

2. Objectives

Our primary objective of this study is to convert our previously published algorithm [1] to a GPU architecture to achieve the real-time performance, while maintaining the motion tracking accuracy.

3. Methods

1) Theory

Using the linear system theory of medical ultrasound systems [2-4], it is easy to demonstrate that the log-compressed cross-correlation function R (between the pre- and post-deformation RF echo signals) within the proximity of the integer-level correlation peak can be written as follows,

$$\log\left(R[\tau_x,\tau_y]\right)\approx\frac{(\tau_x-u)^2}{C_1}+\frac{(\tau_y-v)^2}{C_2} \quad (1)$$

where C_1 and C_2 are related to the ultrasound system parameters and, u and v are the lateral and axial displacements, respectively. A log-compressed magnitude correlation map can be calculated by varying τ_x and τ_y. Eqn. (1) indicates that, in the vicinity the (integer-level) correlation peak, an iso-contour of the (log-compressed) correlation function R can be approximated as an ellipse. If we can obtain the analytical function of the ellipse, the center of the ellipse will correspond to the unknown displacement vector .

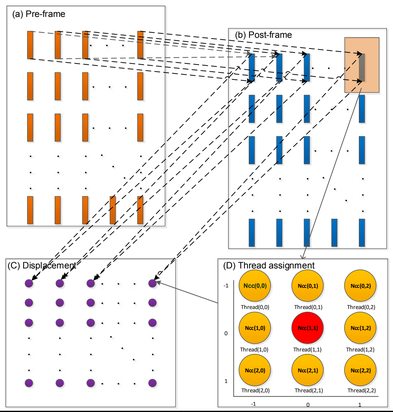


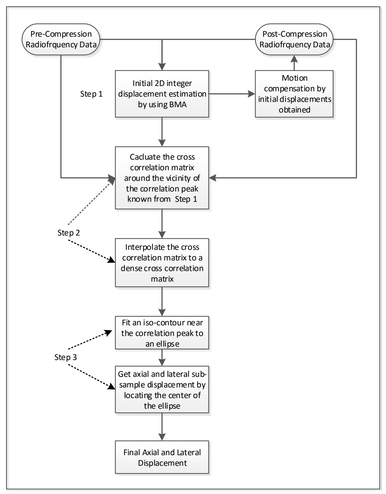
Fig. 1. (a) pre-deformation RF data; (b) post-deformation RF data; (c) output displacement image, the shadow window in (b) shows the available search moving area of a segment of RF data; (d) demonstrates how we assign different threads for the same task at an estimated point (NCC: normalization cross-correlation).

The motion tracking algorithm was implemented in the framework of block-matching algorithm (BMA). The axial correlation window length, axial overlapping ratio, and number of beamlines per correlation window determine the displacement estimation density. Let's assume the final displacement image contains m rows and n columns. As described in the original algorithm [1], there is no need for inter-process communication.

In order to improve the computational speed, we launch (mrows × mcols) × (srow × scols) GPU threads. In the framework of BMA, the (srow) and (scols) denote the size of the search region with units of axial samples and lateral beamlines, respectively. These threads will calculate the correlation values at different location in parallel. Therefore, there is no need for a loop structure inside the kernel function. In order to guarantee the consistency of memory access from these threads, different from the demonstration of Fig.1. (d), a 1D thread structure was applied.

2) Implementation Details using CUDA

Fig.2. shows the coupled subsample displacement estimation method [1] which involved three steps for a sub-sample displacement estimate:



Step 1: a common block matching algorithm [BMA] was used to obtain initial displacement vectors as well as axial strain estimates. Furthermore, in order to improve the correlation, initial displacements were used to shift the search kernel on the post-deformation RF frame by the amount of the estimated integer displacements.

Step 2: we first calculated a discretely-sampled correlation function with integer resolution (e.g. [-3:1:3, lateral] × [-3:1:3, axial]) around the vicinity of the correlation peak found in the integer displacement estimation in Step 1, and, we interpolated the correlation function with improved resolution (e.g. [-3:0.1:3, lateral] × [-3:0.1:3, axial]) by using a B-spline interpolation method..

Step 3: in order to simultaneously obtain the sub-sample displacement vector (u, v) , we fit coordinates of the selected iso-contour to an ellipse [5]. The center of the resulting ellipse corresponds to the sub-sample displacement vector of interest. The final high quality displacement vector (dx, dy) was the sum of the integer displacement vector from Step 1 and the sub-sample displacement vector from the current step (i.e. Step 3).

Fig. 2. A Flow chart summarizing the implementation of the proposed couple sub-sample displacement estimation method

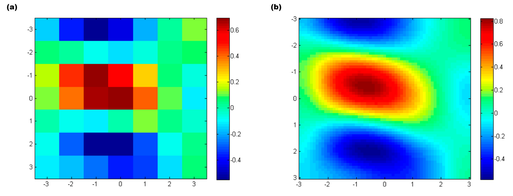


Fig. 3. (a) Obtained normalized correlation coefficients matrix with integer resolution (2) Interpolated normalized correlation coefficients matrix of (a) with improved resolution.

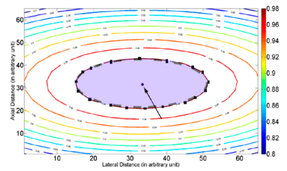


Fig. 4. A graphic representation of the sub-sample estimation method [2]. The color bar represents the correlation. A fitted correlation contour (at 0.98) is depicted by points labelled using rectangles. Coordinates of those points are fitted to an ellipse (see the dashed curve). The small circle (see the arrow) is the center of the fitted ellipse.

4. Results

1) Speedup

Table 1 Comparison of running time for two different implementations. The computational outcomes are measured by seconds.

Displacement Image size	CPU	GPU	Speedup
	Matlab (C MEX code)	Cuda C	
50x50	34.449	0.520	66X
50x100	68.836	0.930	74X
50x150	103.319	1.354	76X
50x200	137.737	1.949	70X
50x250	172.145	2.478	69X
50x300	206.478	3.032	68X

NOTE:
The GPU-parallelized couple subsample displacement estimation method was implemented on a Tesla M2090 card (Nvidia; Santa Clara, CA). The Tesla M2090 was installed in a personal super-computing work-station with a 3.10-GHz Xeon CPU E3-1220 V2 and 16GB memory (Intel Corp.; Santa Clara, CA). All versions were evaluated on this platform.

2) Application

EXAMPLE 1. Strain Elastography

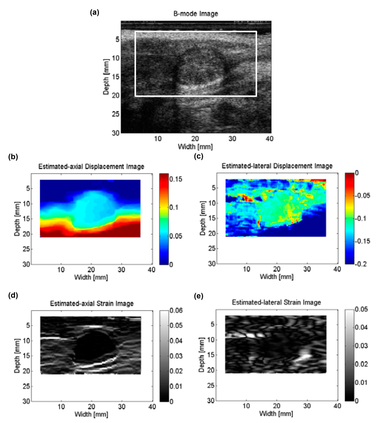


Fig. 5 (a) B-mode image including the region of interest (ROI) (white rectangle). (b) and (c) are the corresponding displacements along axial and lateral directions .The displacement color bars are in millimeters. (d) and (f) are the corresponding strain along axial and lateral directions . The strain color bars are unitless.

To demonstrate accurate tissue motion can be tracked, we show an in vivo Fibroadenoma (FA) acquired under free-hand scanning. The FA was compressed by approximately 0.7%. Details of the scanning protocol and equipment can be found in the previous work by Hall et al. [7] Figures 5(b)-(c) show the axial and lateral displacement estimates can be obtained using the proposed algorithm. Particularly, the lateral displacement image in Fig. 5(c) seems consistent with the tissue landmarks visible in the B-mode image (Fig. 5(a)), i.e. the gross appearance of the lesion and tissue interfaces around it. The FA lesion is also visible both in the axial and lateral strain images (Fig. 5(d)-(e)).

EXAMPLE 2. Ultrasound Shear Wave Elastography

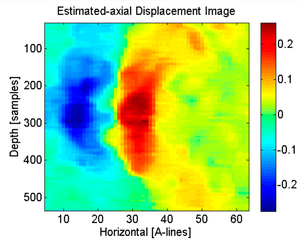


Fig. 6 The demonstration of shear wave propagation. The axial displacements are shown in the image where the shear wave propagates from the left to the right..

In Fig. 6, one axial displacement image tracked from a tissue-mimicking phantom is shown. The tissue mimicking phantom was an Elasticity QA phantom (Model 49A, CISR Inc., VA, USA). Shear wave was generated using a research Ultrasound scanner (V1, Verasonics Inc., WA, USA) and ultrasound data were subsequently acquired using the same scanner. The shear wave speed estimated based on the proposed GPU method was in a good agreement (< 2%) with the stiffness supplied by the vendor,

5. Conclusions

In this study, we implemented a GPU-accelerated subsample displacement estimation method. The experimental results indicate that the GPU-version has a great improvement in computation efficiency. Combining the high subsample estimation accuracy in both directions with real-time performance by GPU parallel computation, the new speckle tracking method has many potential applications in assessment of tissue motion beyond discussed in this poster. The proposed methods will be evaluated further in clinical trials, and then improved accordingly.

Acknowledgments

This study is supported in part by a NIH grant (R15-CA179409-01A1) and a equipment donation from Nividia. The first author has been awarded a post-doctoral fellowship from the China Scholarship Council.

References

[1] J. Jingfeng and J. H. Timothy, Physics in Medicine and Biology, vol. 60, p. 8347, 2015.
[2] J. Meunier and M. Bertrand, IEEE Transactions on Medical Imaging, vol. 14, pp. 293-300, 1995.
[3] J. A. Jensen, The Journal of the Acoustical Society of America, vol. 89, pp. 182-190, 1991.
[4] Y. Li and J. A. Zagzebski, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 46, pp. 690-699, 1999.
[5] A. Fitzgibbon, M. Pilu, and R. B. Fisher, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 21, pp. 476-480, 1999.
[6] T. J. Hall, Y. Zhu, and C. S. Spalding, Ultrasound in Medicine & Biology, vol. 29, pp. 427-35, 2003.