759 Final Project Proposal

Motivation: Elastography is a process of using Ultrasound to measure the elasticity of a material from an applied force. This has many important uses, especially in medical imaging. In our Ultrasound Lab in the Medical Physics Department, we work on calculating the elasticity for cardiac and liver applications. This done by calculating displacement from a given force in the tissue. The gradient of the displacement is then taken to give a relative strain measurement of the tissue. This strain is an important aspect for clinicians to deduce the tissue properties for assessing current treatments or ailments.

Current methods for calculating the displacement include the Multilevel using 2D Normalized Cross Correlation; the displacement estimation method developed in our lab. The Multilevel algorithm operates as a pyramid, where initial displacements are computed starting on a coarse grid using large kernels, followed by computations on finer and finer spatial grids to obtain high SNR estimates with high spatial resolution. This method is very computationally expensive and time consuming, so it currently is done post-procedure due the time constraints of procedures. We aim to create a GPU implementation of this displacement estimation method in hopes of speeding up processing time, leading to a more real-time implementation.

Problem Statement: We aim to create a GPU implementation for one grid level of the Multilevel using 2D Normalized Cross Correlation to speed up displacement estimation computational time. This single level can easily be repeated in the future for the other levels to create the pyramidal scheme that the Multilevel ensues.

Proposed Implementation Approach: Bottleneck in the displacement estimation using radio-frequency (RF) ultrasound signal is the calculation of two-dimensional normalized cross-correlation (NCC). NCC is calculated between a fixed-sized window from pre and post RF data at different shift locations. Then the shift corresponding to the maximum value of NCC is assigned as the displacement between the two frames. This process is repeated over the whole image to generate the final displacement map. Main focus of our project is to leverage the computational capability of GPU to speed up the NCC calculation process. Typical size of RF frame is 6000⨯350 while corresponding displacement image is around 300⨯250 pixels. Estimating one pixel in the displacement map involves number of NCC calculation at different shift of predetermined fixed size which provides us with an opportunity of parallelization. We want to assign one cuda block to calculate on pixel displacement image resulting into 300⨯250 number of blocks in GPU. Within each block, appropriate number of threads will be assigned which will calculate the NCC at each shift. Maximum NCC calculation for each block will be done a single thread after all other threads are done calculating NCC. Final displacement image will be brought back to CPU and further post-processing will done sequential manner. The implementation is done using MATLAB. Custom cuda kernel will written in C++ and compiled with nvcc. Mex file will have written to interface the cuda kernel with MATLAB so that it can be called from MATLAB. Comparison of accuracy will be done with current sequential implementation in MATLAB.

Future Accomplishments: The GPU implementation will be compared against CPU implementation both for accuracy and speed up. In future work the GPU implementation for the single level of the Multilevel algorithm can be repeated for other levels to create a fast, comprehensive displacement estimation algorithm. This algorithm may then replace our current slow CPU implementation to increase our post-processing time. Even further speed ups can be made after this project to the point that this method can be implemented real-time for clinicians.

Deliverables:

1. Establish an interface from MATLAB to C/CUDA.
2. CUDA implementation of the displacement estimation algorithm for one level. The goal is to have two iterations of the algorithm. Version 1 will attempt to parallelize the algorithm with a strict focus on accuracy. Version 2 will attempt to get further speedup by exploring strategies such as coalesced/shared memory accesses.
3. Elastogram similarity comparison using simulated/actual data from applications in ultrasound imaging.
4. Performance analysis showing the effect of algorithm parameters such as window size on speedup between our implementation and the original MATLAB implementation.
5. Presentation of results to research group and advisor.
6. Project report containing the proposal and a summary of the work that was done with results no later than December 21st at 11:59 PM.
7. Upload code to GitHub that can directly verify the results from the final project report.

Team members: (Number reflects deliverable number above.)

- 2nd year PhD candidates in the Electrical and Computer Engineering Department.
- Research assistants in the Varghese Ultrasound Lab advised by Dr. Tomy Varghese.

Robert Pohlman: 2, 4, 7
Rashid Al Mukaddim: 1, 2, 5
Michael Turney: 2, 3, 6

Participate in Rescale sponsored Final Project competition: No, we will not be participating in Rescale Final Project competition.

Link to your Final Project Repo: https://github.com/mturney2/Final-Project-Code