Using graphics processing units to achieve real-time bone surface extraction from volumetric medical ultrasound image data using local phase features

AMIR-KHALILI A¹, ABUGHARBIEH R¹, HODGSON AJ²

¹Biomedical Image and Signal Computing Lab, Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada
²Centre for Hip Health and Mobility, Mechanical Engineering, University of British Columbia, Vancouver, BC, Canada

ahodgson@mech.ubc.ca

Introduction: We present a novel method for real-time bone surface extraction from clinical 3D B-mode ultrasound data in the context of orthopedic fracture reduction surgery (e.g. hip, distal radius and femur). Our method detects local phase features based on 3D directional (log-Gabor) filters. In previous work [1], we have demonstrated the effectiveness of phase symmetry (PS) features for segmentation and localization of bone fractures in 3D ultrasound. In a recent extension to that work, we demonstrated a drastic improvement in the computational efficiency of PS extraction [2] by taking advantage of the inherent parallelizability of the multi-scale linear filters, as was previously done in the context of real-time multimodal image registration by [3]. The purpose of this paper is to summarize our recent computational study and outline our planned future developments.

Methods: Currently available implementations of 3D local phase image feature extraction techniques based on log-Gabor filters generate the filters ‘on-the-fly’. In order to speed up the implementation, we split the algorithm into two parts; one that can be computed off-line (pre-computed or templated at compile-time), and another that is on-line (computed live as images are acquired). Since the process of generating the filters is quite time-consuming and can be performed independently of the images, we generate all our filters off-line and store them in a filter-bank. Only filters requested for computation are loaded onto the GPU on demand. These filters may be requested either by manual parameter setup or automatically using our automatic parameterization method as described in [4].

To enable quantitative validation, we implemented our image-processing pipeline on four platforms: (1) using MATLAB R2011b on a dual processor host machine with two Xeon x5472 CPUs @ 3GHz (Intel Corp., Mountain View, California, US) with 8GB of RAM (note that some functions in MATLAB such as the FFT routine have been optimized to take advantage of all eight cores of the Xeon machine), (2) using a C++ algorithm on the same machine as in (1) (though this algorithm lacked one significant step in our processing pipeline, so its runtime will be underestimated), (3) an entry level GPU (Nvidia NVS 5400M mobile graphic card with 1GB of memory), and (4) a professional GPU (Nvidia Tesla c2050).

Results: We ran the four implementations 20 times each on 128x128x128 3D ultrasound scans of the iliac crest in live subjects and repeated the processing for 15 combinations of filter parameters. On average, the C++ implementation took 1.93s per volume, the Matlab implementation 1.28s, and the GPU implementation on the Tesla card 0.08s. The NVS card produced results approximately 3-4 times faster than the Matlab implementation, although because the memory on this card was limited, we could not load the full set of filters tested in (4). Overall, our implementation on the Tesla GPU proved to be between 15 to 25 times faster than state-of-the-art methods [2].
**Discussion:** Implementing our automatic US processing algorithm on a professional grade GPU produced dramatic computational improvements enabling full 3D datasets to be processed in an average time of under 100ms which, if proven in a clinical system, would allow for near-real-time computation. We are currently implementing our algorithm on an open research sonography platform (OpenSonix, Ultrasound Medical Corporation, Richmond, BC, Canada). High powered graphic cards can easily be integrated into the open architecture of this system, thus enabling GPU computation on its diagnostic medical and research ultrasound devices [5][6]. This implies that our algorithm, developed using the CUDA parallel programming environment (Nvidia Corporation Santa Clara, CA, USA), can be modified to acquire data from the host sonography machine with minimum latency. This is advantageous compared to methods that process raw radio frequency data or access different ultrasound modalities over a TCP/IP connection [7] and process them on a separate device. We intend to use this platform within a surgical environment for accurate detection of fractures and as an integral part of our developing of a computer aided surgery pipeline [8] where we use PS features to register intra-operative ultrasound to pre-operative computed tomography (CT) images. We anticipate a further speedup of approximately three times as the next generation of GPU cards is released as Nvidia is reporting such gains with their new Kepler architecture.

**References**


