GPU Implementation for Non-Cartesian Parallel MRI

H. Shahzad\textsuperscript{1,2} and H. Omer\textsuperscript{1}

\textsuperscript{1}: COMSATS Institute of IT, Islamabad, Pakistan, \textsuperscript{2}: National Centre for Physics, Islamabad, Pakistan

Introduction:

MRI scanners conventionally use Cartesian trajectories to acquire data in k-space which is line-by-line acquisition on a rectangular grid. Non-Cartesian trajectories provide an efficient coverage of the k-space and contain benign aliasing artifacts because of under-sampling. Non-Cartesian trajectories are computationally more demanding and require additional operations such as gridding and de-gridding.

Theory:

SENSE:

Sensitivity Encoding (SENSE) is a Parallel MRI (PMRI) reconstruction algorithm which uses coil sensitivity information to reconstruct fully sampled image from the acquired under-sampled data. Pruessmann et al. proposed a method called Iterative SENSE (CG-SENSE) reconstruction for non-Cartesian trajectories which uses Conjugate Gradient (CG) algorithm to reconstruct the MR Image from the non-Cartesian under-sampled data. In CG-SENSE, gridding and de-gridding operations (along with FFT) are required to reconstruct MR image for the non-Cartesian trajectories. The stability of CG-SENSE has been verified using simulations with spiral, radial and random trajectories in the Pruessmann work and it has been shown that CG-SENSE algorithm enables an efficient image reconstruction with arbitrary k-space sampling patterns. In CPU implementation of CG-SENSE, gridding and de-gridding operations are performed iteratively which increases the computational time. Gridding and de-gridding operations contain inherent parallelism, therefore it is possible to implement them exploiting the inherent parallelism present by using parallel computing language.

Graphical Processing Units (GPUs):

Graphical Processing Units (GPUs) provide fast and efficient implementation of MRI reconstruction algorithms using parallel computing language (Compute Unified Device Architecture (CUDA)). Several research groups have already published their results using GPUs for MR image reconstruction in recent years. GPUs can be used to exploit the inherent parallelism present in gridding and de-gridding operations in CG-SENSE algorithm. In this work we have implemented gridding and de-gridding operations on GPU and the results show that the proposed GPU implementation is approximately 10 times faster than the conventional CPU implementation.

Methods and Results:

For the GPU implementation, the software platform used in this work is MSDN 2010 and NVIDIA CUDA toolkit v6.5 (integrated in MSDN 2010). CUDA code is executed on GPU model NVIDIA GeForce GTX 780 with 2304 CUDA cores and 3072 MB GDDR memory. The performance of gridding and de-gridding implementation on GPU is compared with its C language implementation for CPU core i7 with 2.9 GHz clock speed and 8 GB RAM. The conventional CPU implementation is very similar to the GPU implementation. The number of operations is exactly the same in both cases. Figure 1 shows the scope of this work. The CPU implementation executes the code sequentially using ‘For’ loops whereas GPU implementation breaks up the tasks and launches the threads to execute the tasks in parallel. In GPU, threads are launched as per number of operations required in gridding and de-gridding to attain the highest level of parallelism. Each thread is capable of selecting the pixel values from the acquired multiple coil MRI data and maps the acquired data into experimental k-space trajectory (spiral in our case) in case of de-gridding operation. For gridding operation, each thread performs similar functionality as in case of de-gridding but it maps the data to Cartesian grid from the experimental trajectory (spiral trajectory).

The temporary variables for thread usage are stored in registers instead of global memory to reduce the reconstruction time. CUDA events are used to measure the elapsed time. To monitor the reconstructed image quality for both the CPU and GPU implementations, we have used artifact power (AP) as a quantifying parameter. AP provides the “Square Difference Error” between the reconstructed image and the reference image. AP should be minimum (ideally zero) for good reconstruction results. The MRI data sets (phantom and human head data sets) used in this work were obtained at St. Mary's Hospital, London using 1.5 T GE Scanner. Table 1 shows a comparison of the CPU and GPU implementations of gridding and de-gridding in terms of time consumption in CG-SENSE. Figures 2 and 3 show the reconstructed phantom image and human head image using both CPU and GPU implementations of gridding and de-gridding operations in the CG-SENSE algorithm. Artifact power is mentioned at the bottom of the reconstructed images. It is clear from the Figure 2 and 3 that GPU implementation does not affect the image quality of the reconstructed image as indicated by the artifact power values. The results show that GPU implementation attains 10 times reduction in execution time (due to the parallel computing on CUDA cores) as compared to the CPU implementation while maintaining the overall image quality.

Conclusion:

Gridding and de-gridding are the main time consuming operations in the CG-SENSE algorithm. GPU implementation of these operations is presented in this work which shows more than 10 times performance improvement in terms of computation time without affecting the overall image quality.