Accelerating Compressed Sensing MRI Reconstruction with GPU Computing

D. S. Smith[1,2], J. C. Gore[1,2], and E. B. Welch[1,2]
[1]Radiology and Radiological Sciences, Vanderbilt University, Nashville, TN, United States, [2]Institute of Imaging Science, Vanderbilt University, Nashville, TN, United States

INTRODUCTION

GPU computing refers to using graphics processing units (GPUs) to perform numerically intensive scientific computations. High-end video cards can contain hundreds of separate floating-point units, allowing for massively parallel computations at a fraction of the cost of CPU-based supercomputers on a per Gigaflop basis. The power of GPU computing is already being realized in advanced medical image reconstruction (1-4). Compressed sensing (CS) MRI (5) is an iterative MRI reconstruction technique, so it is more computationally intensive than traditional inverse Fourier reconstruction. One barrier to the routine adoption of CS MRI is the perceived delay between acquisition and reconstruction of useful images. Compressed sensing solvers work almost entirely with vector and image arithmetic, making them an excellent candidate for acceleration through GPU parallelization. Here we illustrate how GPUs can be used to achieve significant increases in CS reconstructions of large MRI data sets at moderate cost.

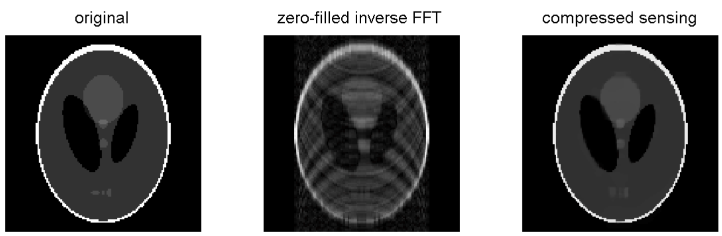

Figure 1: Example reconstructed image obtained in benchmark.

MATERIALS AND METHODS

The reconstructor platform uses a six-core, 2.67 GHz Xeon X5650 with 12 MB of L3 cache, 12 GB DDR3 RAM, and an NVIDIA Tesla C2050 GPU card. Our software platform was MATLAB R2010b (Natick, MA) combined with Accelereyes (Atlanta, GA) Jacket 1.5.0, CUDA Toolkit 3.1, and CUDA developer driver 260.24. The open-source split Bregman code of Goldstein & Osher (6) was chosen as the CS solver and modified to use Jacket. A baseline single-precision, matrix multiply benchmark was performed to measure peak numerical performance of the CPU and GPU on our particular system. To remove dependencies on the multithreading performance of MATLAB, the CPU benchmark was run without multithreading. The CS benchmark was a partial Fourier reconstruction of a Shepp-Logan phantom subject to a random 4x undersampling (75% of k-space randomly deleted). Fig. 1 shows a sample reconstruction. This reconstruction was repeated for image dimensions ranging from 32 to 4096 on both the CPU and the GPU, and timings and reconstruction error were measured.

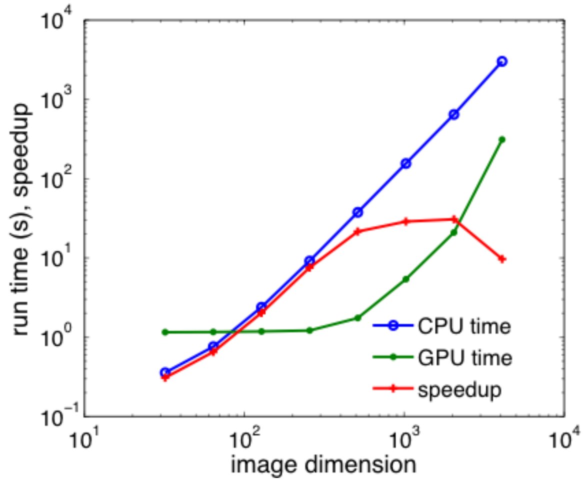

Figure 2: Results of CS reconstruction timing experiment.

RESULTS

The baseline matrix multiply benchmark produced a peak Gigaflop rating of the system of 23.4 Gflops for the CPU and 587 Gflops for the GPU. This suggests that speedups of 25x or more of the CS reconstruction could be achieved. The results of our CS reconstruction benchmark are shown in Fig. 2. Table 1 shows the performance of the system in terms of voxels reconstructed per second as a function of image dimension. The CPU and GPU reconstructed identical images to within the numerical uncertainty.

Dimension	32	64	128	256	512	1024	2048	4096
CPU time (s)	0.357	0.759	2.40	9.15	37.6	155	645	3010
GPU time (s)	1.15	1.17	1.18	1.21	1.75	5.40	21.0	311
CPU rate (vox/s)	2900	5400	6800	7200	7000	6700	6500	5600
GPU rate (vox/s)	890	3500	14000	54000	150000	190000	200000	54000
Speedup	0.31	0.65	2.0	7.5	22	29	31	9.7

Table 1: Reconstruction speeds on the CPU and GPU.

DISCUSSION

We have shown that GPUs dramatically accelerate CS MRI reconstruction of large images. For our system, images of dimensions between 512 and 2048 are able to realize the maximum theoretical speedup of the GPU. The gain for smaller images is progressively hampered by communication overhead. The largest images cause inefficient memory access patterns on the GPU, reducing the potential gains. The optimal image dimension is fortunately similar to that of high-resolution MRI data, so GPU computing seems to be ideally suited for fast CS MRI reconstruction.

REFERENCES [1] Hansen et al.; *Mag Reson Med* 2008;59:463-468, [2] Sorensen et al.; *IEEE Trans Med Imag* 2008;27:538-547, [3] Stone et al.; *J. Parallel Distrib. Comp.* 2008;68:1307-1318, [4] Jia et al.; *Med Phys* 2010;37:1757-1760, [5] Lustig et al.; *Mag Reson Med* 2007;58:1182-1195, [6] Goldstein, T., and Osher, S.; *SIAM J Imag Sci* 2009;2:323-343

ACKNOWLEDGEMENTS NIBIB T32 EB001628. RO1 EB00461.