

PowerGrid: A open source library for accelerated iterative magnetic resonance image reconstruction

Alex Cerjanic^{1,2}, Joseph L Holtrop^{1,2}, Giang Chau Ngo¹, Brent Leback³, Galen Arnold⁴, Mark Van Moer⁴, Genevieve LaBelle^{2,5}, Jeffrey A Fessler⁶, and Bradley P Sutton^{1,2}

¹Bioengineering, University of Illinois at Urbana-Champaign, Urbana, IL, United States, ²Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL, United States, ³PGI Compilers & tools; an NVIDIA brand, Portland, OR, United States, ⁴National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL, United States, ⁵Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, United States, ⁶Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, United States

Synopsis

PowerGrid is an accelerated, open source, freely available toolkit for iterative reconstruction supporting non-Cartesian trajectories. Using high level compiler directives, GPU accelerated Fourier transform operators were implemented in a high level syntax designed to correlate with the popular Image Reconstruction Toolbox (IRT). A speed-up of up to 8.96x over the unaccelerated IRT reconstruction was obtained using an NVIDIA Tesla K40c accelerator.

Introduction

Iterative methods for image reconstruction problems in MRI are essential for advanced techniques including: compressed sensing, low rank reconstruction, SENSE reconstruction¹, non-Cartesian trajectories^{2,3}, field correction for off-resonance distortion², and joint-reconstruction based parameter estimation⁴. Iterative reconstructions can enable high resolution imaging with high SNR, such as submillimeter isotropic diffusion weighted acquisitions on a 3T clinical scanner⁵, by taking advantage of parallel imaging and physics-based field correction. However, all of these advanced reconstruction techniques typically have high computational demands and may not be available to clinicians and researchers.

Fostering robust research and exchange in MR reconstruction techniques requires sharing open and extensible code for others to refine, extend, and improve. Previous work with graphics processing units (GPUs) as hardware accelerators for image reconstruction, such as the IMPATIENT toolkit⁶⁻⁸, produced very high speedup factors over CPU reconstructions for non-Cartesian field corrected SENSE reconstructions. However, its implementation is unfriendly for MR physicists to extend and adapt, particularly to accommodate additional problem formulations such as motion correction models or alternative regularization schemes. In this work, we utilize compiler directives (available through OpenACC 2.0) to target GPU accelerators with intuitive, high level C/C++ syntax to build extensible reconstruction software and tools. The result, PowerGrid, is an open source toolkit that makes accelerators and iterative reconstruction as accessible to MR physicists as the commonly-adapted Image Reconstruction Toolbox (IRT)⁹.

Implementation:

Building upon a set of hardware-accelerated operators such as the discrete Fourier transform (DFT) and the nonuniform fast Fourier transform (NUFFT) via gridding, objects implementing reconstruction techniques were coded in C++ to take advantage of the accelerated transforms. Figure 1 shows an example of how these objects maintain similar syntax and structure to those based in MATLAB from the IRT⁹ including a demonstration of a complete gradient descent solver in 6 lines of PowerGrid/C++ code with intuitive syntax using freely available libraries¹⁰. Table 2 shows a list of features implemented as part of PowerGrid.

The NUFFT and the DFT were written in C++ for CPUs and annotated for use with GPU accelerators (PGI/NVIDIA OpenACC Toolkit 15.7) as shown in Figure 2. The OpenACC compiler directives enable the use of CPU based C code on accelerators such as the GPU with suggestions to the compiler about parallelism and data movement.

Methods:

Test data sets were acquired on a Siemens 3 T Trio scanner using spiral readout acquisitions and either a 32-channel or 12-channel head coil. The 12-channel data was compressed to 4 virtual coils using Siemens hardware coil compression. Reconstructions were performed on a quad-core Intel Xeon E5620 Linux workstation, 12 GB RAM, with a Tesla K40 GPU and Ubuntu Linux 14.04.

Results:

Table 2 compares the IRT/MATLAB reconstructions on CPU only versus GPU enabled reconstructions utilizing PowerGrid. For the smaller 2D dataset (192x192x1x32 coils, R=2), the speed up factor was much more modest (1.69x) due to the dominance of CPU-GPU transfers and CPU-based linear algebra vs. the small Fourier transforms

transforms on GPU. For larger datasets, such as 3D datasets, the NUFFT on GPU via gridding performed better than the IRT/MATLAB and the DFT on GPU.

For the smaller 3D test dataset (64x64x16x4 coils, R=1), PowerGrid had a speed up factor of 8.96x via GPU acceleration versus the fastest MATLAB implementation in the IRT. The larger 3D test dataset (120x120x16x32 coils, R=1), Figure 3, had a slower speedup of (2.23x) over the IRT implementation due to the increased number of coils as the SENSE operator is not currently GPU accelerated and additional coils results in additional CPU-GPU transfers and overhead.

Finally, the modular nature of PowerGrid was demonstrated with a nonlinear motion induced phase correction algorithm¹¹ derived from the SENSE operator. The phase corrected example sped up by 4.0x over the IRT.

Conclusion:

PowerGrid provides access to GPU acceleration for iterative reconstructions that is both adaptable and extensible for the implementation of additional reconstruction algorithms. By maintaining an object and operator based syntax, implementing novel algorithms in C++ are simplified in terms of programming time and required expertise. In our initial observations, reasonably well-experienced IRT users can perform an initial port of their MATLAB-based code to C++ using PowerGrid in less than a day. The combination of an open-source, freely available (<http://mrfil.github.io/PowerGrid>) toolkit with field inhomogeneity correction and accelerated non-uniform Fourier transforms makes the practical use of longer readout, non-Cartesian trajectories more accessible and more computationally tractable.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. We acknowledge support from the Blue Waters project by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. We also acknowledge the assistance of Fernanda Foertter, Oak Ridge Leadership Computing Facility, and the National Center for Supercomputing Applications

References

- [1] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: Sensitivity encoding for fast MRI," *Magnetic Resonance in Medicine*, vol. 42, no. 5, pp. 952–962, Nov. 1999.
- [2] B. P. Sutton, D. C. Noll, and J. A. Fessler, "Fast, iterative image reconstruction for MRI in the presence of field inhomogeneities," *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, pp. 178–188, Feb. 2003.
- [3] J. A. Fessler and B. P. Sutton, "Nonuniform fast fourier transforms using min-max interpolation," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 560–574, Feb. 2003.
- [4] Giang Chau Ngo and B. P. Sutton, "R2* mapping for robust brain function detection in the presence of magnetic field inhomogeneity," presented at the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2014, pp. 1537–1540.
- [5] Holtrop JL, Van AT & Sutton BP. Pushing the Resolution of 3D Spin Echo Diffusion Acquisition. Proceedings of the 20th Annual Meeting of ISMRM, Melbourne 2012:pg 1881.
- [6] J. Gai, N. Obeid, J. L. Holtrop, X.-L. Wu, F. Lam, M. Fu, J. P. Haldar, W.-M. W. Hwu, Z.-P. Liang, and B. P. Sutton, "More IMPATIENT: A gridding-accelerated Toeplitz-based strategy for non-Cartesian high-resolution 3D MRI on GPUs," *Journal of Parallel and Distributed Computing*, vol. 73, no. 5, pp. 686–697, May 2013.
- [7] W. M. W. Hwu, D. Nandakumar, J. Haldar, I. C. Atkinson, B. Sutton, Z.-P. Liang, and K. R. Thulborn, "Accelerating MR image reconstruction on GPUs," presented at the IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009. ISBI '09, 2009, pp. 1283–1286.
- [8] S. S. Stone, J. P. Haldar, S. C. Tsao, W. M. W. Hwu, B. P. Sutton, and Z. P. Liang, "Accelerating advanced MRI reconstructions on GPUs," *Journal of Parallel and Distributed Computing*, vol. 68, no. 10, pp. 1307–1318, Oct. 2008.

[9] J. A. Fessler, "Image Reconstruction Toolbox." Available at: <http://web.eecs.umich.edu/~fessler/code/>

[10] C. Sanderson, "Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments.," NICTA, 2010. Available at: <http://arma.sourceforge.net>

[11] Liu C, Moseley ME & Bammer R. Simultaneous phase correction and SENSE reconstruction for navigated multi-shot DWI with non-cartesian k-space sampling. Magnetic resonance in medicine 2005;54:1412-1422.

Figures



Figure 1: Code listings comparing the implementation of a SENSE operator for both non-Cartesian and Cartesian reconstruction illustrating the forward and adjoint operators in Iterative Reconstruction Toolbox (IRT)/MATLAB and PowerGrid/C++ syntax. Also included is an example of a gradient descent reconstruction based SENSE reconstruction.

PowerGrid Features	Test System
<ul style="list-style-type: none">CPU and GPU Accelerated Fourier Transform OperatorsField-Corrected Iterative Fourier Transform (SIFT)Field-Corrected Iterative Fourier Transform (SIFT)Field-Corrected Iterative Fourier Transform (SIFT)Runtime Temporal InterpolationRuntime Temporal InterpolationNon-Cartesian SENSE Operator2D and 3D Reconstruction ParametersQuadratic Regularization PenaltyFast Gradient Descent MethodsNon-Rectangular Masked Phase CorrectionUtilitiesImageIO/ITK InterfaceMathematical Operator InterfaceOpenACC Accelerator Support	<ul style="list-style-type: none">Ubuntu Linux 14.04Intel Xeon E5620 2.43 GHz12GB RAMPGI OpenACC Toolkit 15.7NVIDIA K40c Tesla Accelerator (12GB)Image Reconstruction ToolkitMATLAB R2014a

Table 1: Table of features in PowerGrid and detailed test system setup for results in this abstract.

Test Cases	Run Times (s)		
	IRT/MATLAB	PowerGrid GPU NUFFT	PowerGrid GPU DFT
2D (192 x 192 x 32 coils)	323.44	109.03	72.96
2D (256 x 256 x 32 coils)	169.17	197.88	663.40
3D (64 x 64 x 16 x 4 coils)	341.81	38.14	112.94
3D (120 x 120 x 16 x 32 coils)	1705.01	762.07	*
Phase Corrected 3D (120 x 120 x 4 x 32 coils)	1242.35	1437.57	310.74

* Denotes reconstruction did not complete in less than one hour

Table 2: Table summarizing run times for the iterative reconstructions between MATLAB implementations and PowerGrid on GPU. Note that for small transforms, such as the 2D case, CPU-GPU transfer overhead dominates the runtime, limiting potential speedup via GPU. For larger transforms, such as the 3D case, larger speedups are achievable.

```
... need to call fft2 on the FFT for gradient to work so we need to use fft2
...
//checking here is optional to test compiler flags with independent and collapse pragmas
#pragma omp taskwait(1) independent
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        for (int k = 0; k < N; k++) {
            // ...
        }
    }
}
// ...
}
```

Figure 2: Example of OpenACC accelerated routine from griddingSupport.hpp implemented with compiler pragma directives. OpenACC directives express parallelism and data movement to allow the compiler to translate code from CPU to GPU.

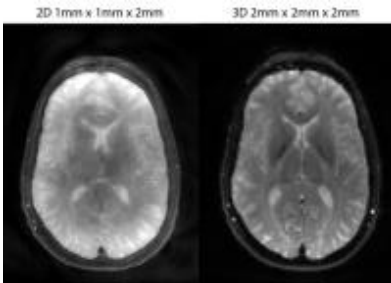


Figure 3: Examples of GPU accelerated, non-Cartesian, field corrected reconstructions from PowerGrid. The left image corresponds to the 2D 256x256x32 coils test case (Spin Echo, FA 90, TE 60ms/TR 2s), and the right image to the 3D 120x120x16x32 coils case (Spin Echo, FA 90, TE 84ms/TR 2s) from Table 2.

