Fast diffusion-guided QSM using Graphical Processing Units

Owen L Kaluza[1], Amanda C L Ng[2,3], David K Wright[4,5], Leigh A Johnston[5,6], John Grundy[7], and David G Barnes[2]

[1]Monash e-Research Centre, Monash University, Clayton, Victoria, Australia, [2]Monash Biomedical Imaging, Monash University, Clayton, Victoria, Australia, [3]Department of Electrical & Electronic Engineering, The University of Melbourne, Parkville, Victoria, Australia, [4]Centre for Neuroscience, The University of Melbourne, Parkville, Victoria, Australia, [5]Florey Institute of Neuroscience and Mental Health, Parkville, Victoria, Australia, [6]NeuroEngineering Laboratory, Dept. Electrical & Electronic Engineering, The University of Melbourne, Parkville, Victoria, Australia, [7]Centre for Complex Software Systems and Services, Swinburne University of Technology, Hawthorn, Victoria, Australia

Target audience. Practitioners of quantitative susceptibility mapping; computational imaging scientists using graphical processing units to accelerate algorithms.

Purpose. Diffusion-weighted (DW) magnetic resonance (MR) images can be used to provide per-voxel geometric models for improved quantitative susceptibility mapping (QSM). The approach studied here, diffusion-guided QSM (dQSM),[1] treats the magnetic susceptibility effect of each image voxel as isotropic (spheres[2]) or axial (cylinders[3]) depending on the fractional anisotropy (FA) in corresponding DW images. The computation of the matrix formulation of the problem is prohibitively expensive on central processing unit (CPU) cores. Acceleration of the algorithm by utilizing graphics processing unit (GPU) cores is necessary to achieve image computation times practical for research use today, and for clinical application in the near future.

Methods. *Formal problem statement.* In dQSM, $\Delta B = -\gamma.\mathrm{TE}.\phi^{-1}$, where γ is the gyromagnetic ratio of water, TE is echo time and ϕ is phase in the (complex) T_2* gradient-echo (GRE) image.[1] The susceptibility map, Δχ, is related to Δ*B* according to $\Delta B(\boldsymbol{r}) = \sum F(\boldsymbol{r}', \boldsymbol{r}-\boldsymbol{r}')\,\Delta\chi$, where $F(\boldsymbol{r}', \boldsymbol{r}-\boldsymbol{r}') = F_s(\boldsymbol{r}-\boldsymbol{r}')$ for spherically-modeled voxels (FA < 0.2) and $F(\boldsymbol{r}', \boldsymbol{r}-\boldsymbol{r}') = F_c(\boldsymbol{r}-\boldsymbol{r}')$ for cylindrically-modeled voxels (FA ≥ 0.2). Δχ is solved by minimising $\kappa\,\|\mathbf{Ax}-\mathbf{b}\|_2^2 + (1-\kappa)\,\|\mathbf{Lx}\|_2^2$ where $\mathbf{Ax}-\mathbf{b}$ is the matrix-vector representation of $\sum F(\boldsymbol{r}', \boldsymbol{r}-\boldsymbol{r}')\,\Delta\chi - \Delta B(\boldsymbol{r})$ and **L** is a second-order derivative.

Minimisation on the GPU. The Landweber Iteration (LI) is used to compute the minimisation. The LI corresponds to an *interact* kernel,[4] predisposing it to an efficient vectorised implementation. The matrix A is *extremely* large, of the order 1 terabyte (TB) for small animal brain imaging and 10-20 TB for human brain imaging. Ordinarily this would necessitate implementation on a large-memory supercomputer. However, the critical observation in this work is that A can be computed *on-demand* from a relatively small input data set (< 10-100 MB), and the LI solution for dQSM imaging is therefore *feasible for adaptation to the GPU.* Single-precision calculations are sufficient (given the typical dynamic range of MR imaging data), and memory access patterns can be arranged to be entirely sequential, thus a *highly-efficient GPU implementation is possible*. We have implemented a vectorised version of the LI solution in OpenCL, including provision for multi-GPU deployment, with the demarcation between fine and coarse parallelism defined by the existing code targeted to the BlueGene/Q[1]. Despite its relative immaturity compared to NVIDIA's CUDA framework, OpenCL offers the same capability for acceleration, and operates on a wider range of hardware platforms. Our initial GPU implementation took place as follows: (1) analysis of the algorithm complexity and memory access patterns to identify loops to be parallelised, and to determine instantaneous memory requirements; (2) allocation and initialisation of GPU device memory buffers for the problem elements (A, B, x, ...); and (3) implementation of OpenCL kernels to compute the LI using massive parallelism. The LI computation comprises two expensive $O(N^2)$ calculations in nested loops. The loop over the output data buffer was vectorised (i.e. allocated to parallel threads on the GPU), leaving a single loop within the kernel. This approach minimizes algorithm complexity, avoids race conditions in output, and enables maximally-coalesced input memory access.

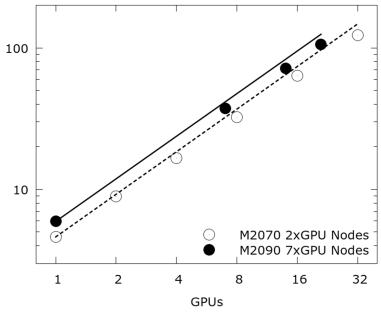

Figure 1. Measured speed-up (relative to 12 Westmere 2.66GHz cores) for gSTAR and MASSIVE. Solid (dashed) line indicates ideal scaling for gSTAR (MASSIVE).

Validation and timing. The method to ensure the accuracy of the GPU-based LI solution and improve its performance was: (1) comparison of the output susceptibility map Δχ with those from the reference CPU implementation at coincident iterations; (2) accurate (instrumented) measurement of execution times (to completion for validation; to 100 iterations for reporting and optimisation purposes); and (3) improvement of kernel and driver code by repeated validation and execution timing following the application of code optimisations. ΔB maps derived from numerical phantom and ex-vivo mouse brain data[1] were processed with dQSM implemented on: (1) the IBM Blue Gene/Q supercomputer (based on 16-core PowerPC 1.6GHz CPUs); (2) dual-GPU nodes of the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) cluster (dual 6-core Intel Westmere 2.66GHz CPUs and 2 NVIDIA Tesla M2070 GPUs per node); (3) 7-GPU nodes of the GPU Supercomputer for Theoretical Astrophysics Research (gSTAR) cluster (dual 6-core Westmere 2.66GHz CPUs and 7 Tesla M2090 GPUs per node); (4) 2-GPU nodes of the swinSTAR cluster (dual 8-core Intel SandyBridge 2.2GHz CPUs and 1 Tesla K10 dual-GPU per node).

Results. The reference implementation on the Blue Gene/Q computes the susceptibility map for the mouse brain in 16 hrs using 4096 cores (256 nodes) - 52 seconds per iteration (including all overhead); 12 cores of a dual Westmere node compute the map at 640s per iteration; and a single M2070 GPU computes the LI solution at a rate of 140s per iteration. Considering realistic deployment configurations, 21 Tesla M2090 GPUs deployed in three high-density servers can compute the susceptibility map in just 112 minutes; and 16 Kepler K10 GPUs deployed in 16 low-density servers are projected (based on single K10 timings) to compute the map in 72 minutes. Figure 1 shows the scalability of our solution. Figure 2 summarises the measured completion time for a range of realistic hardware configurations.

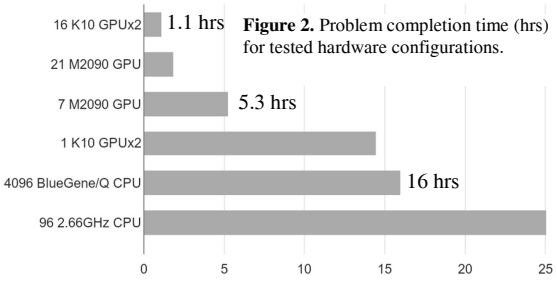

Figure 2. Problem completion time (hrs) for tested hardware configurations.

Discussion. Diffusion-guided QSM is drastically faster on GPU than CPU. Contemporary compute GPUs such as the Tesla M2070 (K10) deliver speed-ups of order 8-10 (11-14) times over contemporary 6-core CPUs such as the Intel Westmere, 50-60 (70-85) times over a single core thereof. Our implementation is demonstrably scalable, i.e. dQSM images can be computed more rapidly with more hardware. The dQSM problem is suited to the GPU for the outstanding reason that the elements of the matrix A in the LI formulation can be computed on-demand; without this property the problem would be intractable on GPUs. Scalability with respect to image size is not reported here, however for human imaging, voxel number will be approximately 4-fold more, and further acceleration of the computation of dQSM images will be needed to achieve clinically-useful post-processing times. The current implementation of the LI is exact - i.e. the susceptibility in each voxel affects every other voxel. By introducing physically reasonable constraints on the spatial support of $F(\boldsymbol{r}', \boldsymbol{r}-\boldsymbol{r}')$ we expect significant additional acceleration of the algorithm.

Conclusion. We have dramatically accelerated the dQSM method, to the extent that its use in research imaging facilities is immediately practicable on quite modest computational hardware. Further speed-up of the GPU implementation should be possible, which, together with algorithmic improvements, and the growth in compute capability of GPUs, is expected to enable clinically-relevant post-processing times (less than 30 minutes on modest hardware).

Acknowledgements. This research was supported under the Australian Research Council's *Discovery Projects* funding scheme (DP120102653). We acknowledge access to the MASSIVE and gSTAR/swinSTAR facilities.

References: 1. ISMRM 2013 abstract 2394: Diffusion-guided quantitative susceptibility mapping. **2.** Liu T, Liu J, de Rochefort L, et al. Morphology enabled dipole inversion (MEDI) from a single-angle acquisition: Comparison with COSMOS in human brain imaging. *Magnetic Resonance in Medicine*. 2011;66(3):777–783. **3.** Lee J, Shmueli K, Kang B-T, et al. The contribution of myelin to magnetic susceptibility-weighted contrasts in high-field MRI of the brain. *NeuroImage*. 2012;59(4):3967–3975. **4.** Barsdell, B.R., Barnes, D.G., Fluke, C.J. Analysing astronomy algorithms for GPUs and beyond. *Monthly Notices of the Astronomical Society*, 2010:408(3):1936-1944.