Multi-GPU Implementation for Iterative MR Image Reconstruction with Field Correction

Y. Zhuo[1], X-L. Wu[2], J. P. Haldar[2], W-M. W. Hwu[2], Z-P. Liang[2], and B. P. Sutton[1]

[1]Bioengineering, University of Illinois at Urbana-Champaign, Urbana, IL, United States, [2]Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, United States

INTRODUCTION

Many advanced MRI image acquisition and reconstruction methods see limited application due to high computational cost in MRI. For instance, iterative reconstruction algorithms (e.g. non-Cartesian k-space trajectory, or magnetic field inhomogeneity compensation) can improve image quality but suffer from low reconstruction speed. General-purpose computing on graphics processing units (GPU) have demonstrated significant performance speedups and cost reductions in science and engineering applications. In fact, GPU can offer significant speedup due to MRI parallelized-data structure, e.g. multi-shots, multi-coil, multi-slice, multi-time-point, etc. We propose an implementation of iterative MR image reconstruction with magnetic field inhomogeneity compensation on multi-GPUs. The MR image model is based on non-Cartesian trajectory (i.e. spiral) in k-space, and can compensate for both geometric distortion and some signal loss induced by susceptibility gradients.

THEORY

Field inhomogeneity near air/tissue interfaces at the ventral brain (i.e. orbitofrontal cortex) leads to susceptibility artifacts, including geometric distortions and signal loss [1-2]. We have previously reported an imaging model including the physics of magnetic field inhomogeneity map and its 1st derivate, which can compensate for both types of susceptibility artifacts [3].

Imaging model: The MR samples d acquired in k-space are described in Eq.(1). $\rho(\mathbf{r})$ is the image estimate at spatial location $\mathbf{r}$; $\mathbf{k}(t_m)$ is the non-Cartesian trajectory in k-space; $\omega(\mathbf{r})=\omega(x,y,z)$ denotes the magnetic field inhomogeneity, as shown in Eq(2), including off resonance constant ω_n (in Hz), within-plane ($G_{x,n}$, $G_{y,n}$) and through-plane ($G_{z,n}$) susceptibility gradients (in Hz/cm). The model is discretized using a 3D rectangle basis function $\varphi_n(x,y,z)$. Eq(3) shows $\mathbf{\Phi}_n(\mathbf{k}(t_m))$, the Fourier Transform (after discretization) of the basis function at location $\mathbf{k}_m$, combined with the field inhomogeneity map and its gradients with Z-shimming gradients.

$$d\left(\mathbf{k}\left(t_m\right)\right)=\int \rho(\mathbf{r}) e^{-i2\pi\omega(\mathbf{r})t_m} e^{-2\pi\mathbf{k}(t_m)\mathbf{r}} \mathrm{d}\mathbf{r} \quad (1)$$

$$\omega(x,y,z)=\sum_{n=0}^{N-1}\left(\omega_n+G_{X,n}\left(x-x_n\right)+G_{Y,n}\left(y-y_n\right)+G_{Z,n}\left(z-z_n\right)\right)\cdot\varphi_n\left(x,y,z\right), \quad (2)$$

$$\mathbf{\Phi}_n\left(\mathbf{k}\left(t_m\right)\right)=sinc\left(\left(k_x\left(t_m\right)+G_{x,n}t_m\right)\Delta_x\right)\cdot sinc\left(\left(k_y\left(t_m\right)+G_{y,n}t_m\right)\Delta_y\right)\cdot sinc\left(\left(k_z\left(t_m\right)+G_{z,n}t_m\right)\Delta_z\right) \quad (3)$$

Iterative CG solver: For image reconstruction, the gridding method can only compensate for the geometric distortion, but not signal loss (which is induced by the susceptibility gradients). Alternatively, our iterative algorithm can model the physics of field inhomogeneity and its gradients. In this work, we use a conjugate gradient iterative algorithm for reconstruction including correction for magnetic field inhomogeneity as described earlier. Therefore signal loss as well as signal distortion can be efficiently reduced and the resulting image quality is improved. We expected to gain a speedup by an order of magnitude for multi-GPU implementation.

METHODS

The reconstruction is implemented on a NVIDIA Tesla S1070 which containing 4 GT200 GPUs. Tesla S1070 yields 4 TFLOPS of peak performance and 408 GB/s memory access bandwidth, and 240 computing cores per processor. The reconstruction code is written using the NVIDIA CUDA 3.0, which supports the single-program, multiple-data (SPMD) parallel execution. The CPU used for comparison is a dual-core 2.4 GHz AMD Opterons with 8 GB of memory on operating system Fedora 10.

Here, we extend our previous work [4-5] to field-compensated model described in theory. In this work, we use up to 4 GPUs to parallelize the computation of multiple slices of 3 different matrix sizes (64x64, 128x128, 256x256). Note that in our implementation, the data (including k-space trajectory, field map and its gradients) were stored on the GPU memory for computation of the whole conjugate gradient algorithm, therefore reducing the time-consuming data transfer and memory allocation. For multi-GPU support, a scheduler is implemented to start a single slice at a time on each GPU, while keeping a balanced load for all GPUs.

Table.1 Performance comparisons between CPU and multi-GPUs.

	CPU	GPU			
		1 GPU	2 GPU	3 GPU	4 GPU
Time (s)	448	1.20	1.11	0.91	0.80
Speedup	\	373x	403x	492x	**560x**

Table.2 Performance comparisons between CPU and 4 GPUs for different matrix sizes.

	64x64x4		128x128x4		256x256x4	
	CPU	GPU	CPU	GPU	CPU	GPU
Time (s)	448	0.8	717	2.1	2864	2.6
Speedup	\	560x	\	341x	\	**1102x**

RESULTS and DISCUSSION

The performance comparison is shown in Table 1-2. For multi-GPUs, the speedup is 373x for 1 GPUs and 560x for 4 GPUs (with 64x64x4 matrix size). We also test different matrix sizes on 4 GPUs. For matrix size higher than 64x64 in each slice, tiling technique is implemented when using constant memory to speedup the memory operation. Therefore, the speedup reduces from matrix size 64x64 to 128x128 in each slice by considering the time for tiling. The speedup reaches **1102x (three orders of magnitude)** for matrix size of 256x256x4. In summary, the computation time is significantly reduced by at least two orders of magnitude for multi-GPUs implementation of our iterative MR image reconstruction algorithm, and the speedup increases along with the number of working GPUs. The resulting images looks almost identical (Fig.1) between CPU and GPU, while the reconstruction is done with (a,c) or without (b,d) present of field map and its gradients. The differences (e,f) between GPU and CPU are negligible (range of 10^{-4}) compared with resulting image intensity (around 1) in (a-d).

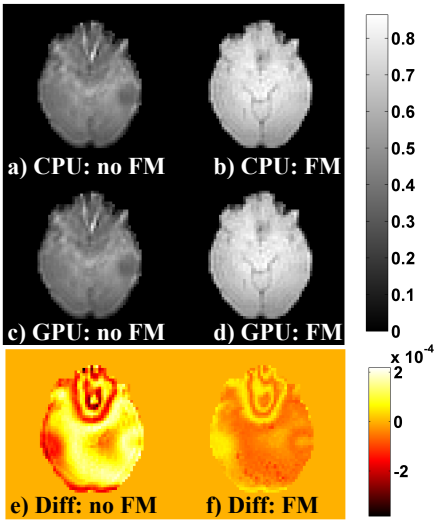

Fig. 1 CPU and 4 GPUs implementation comparisons. a-b) CPU: without and with FM compensation, c-d) GPU: without and with FM compensation, e-f) Difference between CPU and GPU (very small): without and with FM compensation.

CONCLUSION

We proposed a fast multi-GPU-based implementation of our MR imaging model reconstruction algorithm with susceptibility artifacts compensation. The speedup is up to two orders of magnitudes and increases along with number of GPUs. The significant speed improvement reduces computation times down to a clinically acceptable scale.

REFERENCES [1] B.P. Sutton, et al. TMI, 22(2):178-188, 2003. [2] J.A. Fessler, et al. TSP. 51(2):560-574, 2003. [3] Y. Zhuo, et al. EMBC, 2009 Sep; 5721-5724. [4] S.S Stone, et al, JPDDC. 68:1307-1318, 2008. [5] WmW. Hwu, et al, ISBI, 2009, June, 1283-1286.