Accelerating phase modulation for correcting EPI geometry distortion by modern GPGPU parallel computation.

Y-H. Yang[1], T-Y. Huang[2], F-N. Wang[3], and N-K. Chen[4]
[1]National Taiwan University of Science and Technology, Taipei, Taiwan, [2]National Taiwan University of Science and Technology, [3]Department of Biomedical Engineering and Environmental Sciences, National Tsing Hua university, [4]Brain Imaging and Analysis Center, Duke University Medical Center

Introduction

Reduction of computation / reconstruction time is critical for MRI, particularly when the information is needed immediately after the acquisition, such as for real-time image guided therapy, among others. Even though most of the MRI data can be reconstructed immediately with FFT, many advanced algorithms still require long computation time, making them impractical for real-time applications. For example, the 2D phase-modulation can be used to effectively remove EPI distortions based on the field inhomogeneity information [1,2]. However, the calculation complexity of phase-modulation is Ny-fold larger than the regular 2D image reconstruction (where Ny is the phase-encoding step), and the correction for dynamic EPI data (e.g. fMRI or perfusion) may take minutes even using the state-of-the-art computer hardware. It is thus extremely difficult to extend the phase modulation to 3D, or to further combine phase-modulation with parallel imaging, due to its large computation cost. Recently, the parallel computing using general-purpose computation on graphics processing units (GPGPU) has proven capable of accelerating the scientific computation through parallelizing the algorithm. In this study, we evaluate the performance of the GPGPU technique in phase-modulation calculation, in terms of the reduction of the the computation time.

Theory

The modern GPU equipped with massive parallel computing units, which are designed mainly for high-resolution shading especially in 3D gaming environment. The computation cores can actually be used for general arithmetic jobs. For the phase-modulation method, each phase-encoding line is modulated with a phase accumulation calculated from the field map (eq.1).

$$S(m\Delta k_x, n\Delta k_y, l\Delta T) = \iint \rho(x,y)\cdot \exp(im\Delta k_x)\cdot \exp(in\Delta k_y)\cdot \varphi(x,y,l\Delta T)\,dxdy \quad \text{(eq.1)}$$

where $\rho(x,y)$ is the spin density, Δk_x and Δk_y are the incremental in k-space, ΔT is the interecho time interval and $\varphi(x,y,l\Delta T)$ is the phase error term of each line *l*. Notice that each k-space point can be calculated independently and thus the method is suitable for parallel computing algorithm.

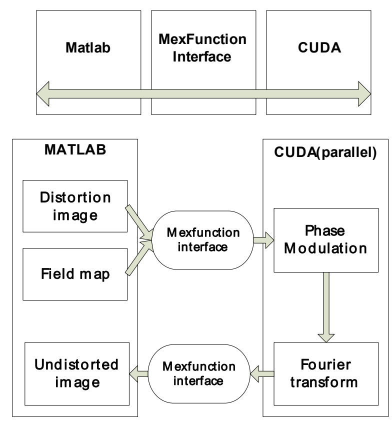

Figure 1: The communication interface and program flow chart of our GPU accelerated algorithm.

Material and Methods

Our program was implemented on GeForce GTX295 (NVIDIA, USA, processor cores per GPU:240 RAM per GPU:896MB) using compute unified device architecture (CUDA) programming model. To be compatible with other image reconstruction Matlab scripts developed in-house (Mathworks, USA), the program was further incorporated with a communication interface with MATLAB. Our implementation consisted of the following procedures. First, the k-space data, the field map, and the echo-spacing were transformed to C language-based memory space with the MATLAB interface “mexFunction”. Second, the data were duplicated into the DRAM of graphic card. Third, the corrected k-space data were calculated with the parallel computing threads. Each k-space point was calculated with one thread performing phase-modulation and thus the threads could be executed independently. Finally, the corrected k-space data and the reconstructed images (with inverse-Fourier-transform) were transferred back to MATLAB work space for further image processing. The developed algorithm was installed into our CUDA workstation (CPU: 4-core Intel i7, GTX296 GPU: 4).

The GPU-accelerated program for EPI distortion correction has been evaluated in two data sets. First, phantom data were acquired with EPI with the following parameters: TR=1000ms, TE:62~71ms, ΔTE=1ms, FOV=240mm, matrix=128×128, using a 3.0 Tesla MR system (Siemens, Tim-Trio, Erlangen, Germany). The field inhomogeneity was purposely enhanced by manually adjusting the current of shim coil. A conventional turbo-spin-echo image was also acquired, and used as a reference. Second, a high-resolution diffusion-tensor data-set, acquired in our previous PROPELLER-EPI study [3] conducted in MGH (Charlestown, MA), was processed with the conventional and GPU-accelerated phase-modulation, and the results were then quantitatively compared. Scan parameters for PROPELLER-EPI (in a 3.0 Tesla MR system; Siemens Magnetom Allegra) included TR/TE=1600/70ms, matrix=128×128, FOV=220mm slice: 16, diffusion gradient direction:6, blade:26. Note that the phase-modulation correction needed to be performed on each blade for PROPELLER-EPI data. Furthermore, we also implemented a CUDA-accelerated version of “imrotate”, a MATLAB command for image rotation (generally gain two-fold speed-up in PROPELLER reconstruction for matrix of 128×128, data not shown due to the length restriction in the abstract). Since our system was equipped with multi-core CPU and multi-GPU, we also compared the computation speed using different numbers of CPU-cores and GPU.

Result

Our method was verified on the phantom study (see Figure 2). The shape of the corrected EPI image matched well with the conventional scan (red profile). The acceleration factor of GPU over CPU was 2.89±0.07 (average value of ten times execution). Applying on the PROPELLER EPI data set, the parallel algorithm reduced the computation time from ~1750 seconds (single core CPU) to ~100 seconds (four GPUs). The detail of the comparison (CPU versus GPU) was listed in table 1.

Discussion and Conclusions

In our study, the GPU computation was used to accelerate the EPI distortion correction, making it practical for real-time applications. The results showed that the CUDA program effectively reduced the reconstruction time. The parallel computation reduced the total image reconstruction time of high-resolution diffusion-tensor images by ~1650 seconds (by 94.24%). The significant reduction in the reconstruction time should make the high-quality PROPELLER-EPI much more practical for clinical utilizations. Moreover, the accelerated distortion correction should also benefit the fMRI study, consisting of a large number of time-series data. Note that the “double-precision floating point” is only supported in high-end cards (GT200). Therefore, our program was implemented using “single-precision floating point” for better compatibility with existing systems. Our results show that the error caused by single-precision calculation was acceptable (less than 10^{-6} %). In conclusion, the GPU computing is a promising method to accelerate EPI distortion correction.

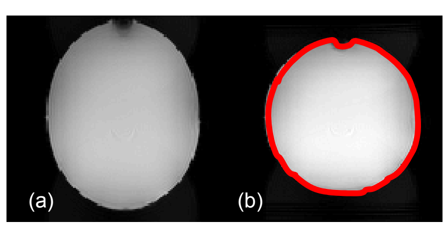

Figure 2: (a) the distorted EPI phantom (b) The corrected image reconstructed by proposed method. The red profile overlaid on top is the profile obtained from conventional turbo-spin-echo scan.

Table 1
CPU VS GPU speed comparison

CPU/GPU #	1	2	4
CPU	1754	1166	631(sec)
GPU	349	168	101(sec)

Table 1: Computation time comparison of CPU and GPU. The data-set for reconstruction was the PROPELLER EPI diffusion tensor imaging.

References

[1]Zeng Huairen et al. MRM(2002)48:137-146
[2]Chen NA et al. MRM (1999)41:1206-1213
[3] Wang FN al. MRM (2005)54:1232-1240