

Accelerating pixel-by-pixel non-linear curve fitting using parallel computation on graphic processing units: Application to pulmonary perfusion mapping.

W-M. Tseng¹, T-Y. Huang¹, Y-R. Lin², and M-T. Wu³

¹Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ²Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ³Department of Radiology, Kaohsiung Veterans General Hospital, Kao-Hsiung, Taiwan

Introduction

Pixel-by-pixel curve fitting is generally applied on quantitative MRI studies, example includes but certainly not limited to, T1 mapping with inversion recovery sequences, dynamic contrast enhancement (DCE) perfusion mapping with gamma-variable fitting[1,2] or pharmacokinetic modeling (e.g. Brix model). The non-linear curve fitting is a time-consuming task since it generally requires lots of iterations to find optimal parameters. Therefore, reconstructing quantitative MRI maps with non-linear curve fitting may require minutes even with state-of-the-art workstations. In this article, parallel computation with graphic computation units (GPU) is proposed to reduce the calculation time of pixel-by-pixel gamma-fitting on pulmonary perfusion datasets.

Material and Methods

The main idea of GPU-based pixel-by-pixel fitting is to distribute the fitting computation of each pixel into a GPU thread and then use massively parallel GPU kernels for computing. Since each pixel can be calculated independently, this method is suitable for parallel computing algorithms. In this study, we implement Levenberg-Marquardt algorithm, which solves nonlinear least-squares problems, [3] into a GPU system with CUDA (NVIDIA, USA). The GPU is programmed to run in SIMD mode (single-instruction-multiple-data). This enables computing curve-fitting (i.e. single instruction) of multiple pixels (i.e. multiple data) in parallel. For the DCE pulmonary perfusion studies, the fitting model of the program is a gamma-variable function, as given in eq.1:

$$s(t) = K \cdot (t - t_0)^\alpha \cdot e^{-(t-t_0)/\beta} \quad (\text{eq.1})$$

where k , α and β are arbitrary parameters, and t is the imaging time, t_0 is the first-bolus arrival time. Most MRI analysis tools in our group work in MATLAB environment (Mathwork, USA). Therefore, the GPU-based parallel curve-fitting is programmed to be compatible with MATLAB through a communication interface "mexFunction". The work-flow is illustrated in Fig. 1. First, the pulmonary DCE images were transformed to the C-language environment. The images were then duplicated into DRAM of GPU. The curve-fitting jobs of all pixels were distributed into GPU threads. After accomplishing curve-fitting of all pixels, the parameters of gamma function (e.g. k , α and β) were transferred back to MATLAB environment for further reconstruction of the perfusion maps. The experiment was conducted with a 1.5T MRI scanner (GE Signa CVi). A volunteer pulmonary DCE dataset was acquired by using an inversion-recovery-prepared segmented EPI technique with cardiac gating. Imaging parameters were TI/TR/TE=180/6.5/1.2 ms, ETL=4, matrix size=256x256 and slice thickness=8mm with 7 coronal slices acquired. A bolus of 0.05 mmol/Kg Gd-DTPA (Magnevist, Schering, Germany) was injected using an MR-compatible power injector at a speed of 3ml/sec. The DCE data were processed with the propose GPU parallel curve-fitting on a personal computer equipped with an Intel Core-i7 920 CPU and four NVIDIA GTX295 GPUs (240 CUDA kernels per GPU).

Result

Figure 2 shows pulmonary-blood-volume (PBV) maps obtained by CPU-based (see Fig.2a) and GPU-based (see Fig.2b) gamma fitting, respectively. CPU-based and GPU-based methods produced the same PBV maps. The total computation times needed to reconstruct PBV maps of all 7 slices are listed in Table 1. GPU-based parallel fitting effectively reduce the PBV-map reconstruction time. By using multiple GPUs, the reconstruction time is reduced to ~37 seconds.

Discussion and Conclusions

This study attempts to implement a parallel pixel-by-pixel fitting toolbox which is compatible to MATLAB environment. The performance of this toolbox was evaluated by a pulmonary DCE perfusion dataset. PBV maps were reconstructed by performing pixel-by-pixel gamma-fitting. GPU-based fitting reduced the total computation time from ~20 minutes to ~0.5 minute. This work can thus effectively speed up the clinical evaluation of pulmonary perfusion. Since this toolbox was programmed with general Levenberg-Marquardt algorithm, it can be applied to other curve-fitting applications, such as T1 mapping or perfusion mapping with pharmacokinetic modeling. In conclusion, the GPU-based pixel-by-pixel fitting is a promising method to accelerate nonlinear curve-fitting problems of MRI applications.

Reference

- [1] Hatabu H., et al. Magn Reson Med;1999;42:1033–1038.
- [2] Levin DL., et al. Magn Reson Med;2001;46:166–171.
- [3] More, J.J., Numerical Analysis, ed. G. A. Watson, pp. 105–116, 1977.

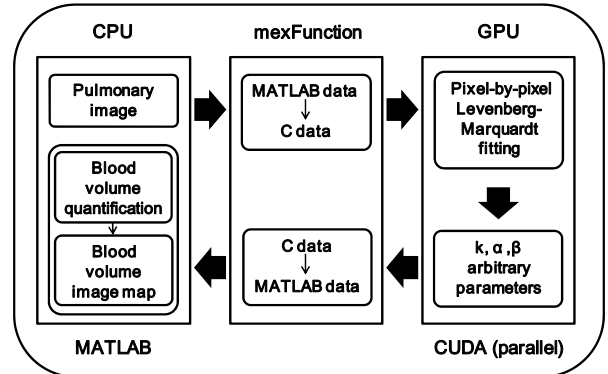


Figure 1: The communication interface and program flow chart of our GPU accelerated algorithm.

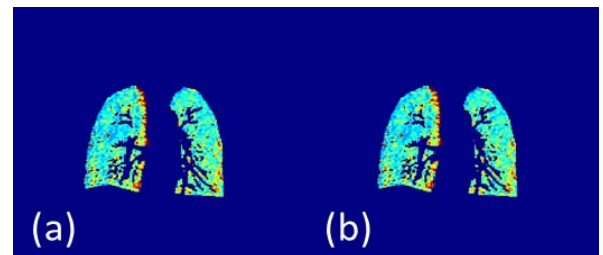


Figure 2: PBV maps computed by (a) CPU-based gamma-fitting (b) GPU-based gamma-fitting

Table 1 The time comparison of CPU and GPU

Processor	Computation Time (second)
CPU	1315.95
GPUx1	127.07
GPUx4	37.03