Preliminary results about non-Gaussian diffusion NMR imaging reconstruction on GPU

Marco Palombo

Introduction

A Nvidia GPU device can have a number of multiprocessors (MP), each of which executes in parallel with the others. On one multi-processor, multiple CUDA blocks (and thus multiple fittings) can execute concurrently. One Levenberg-Marquardt (LM) fitting is executed in one CUDA block, with many parallel CUDA threads. The function we used to perform LM fittings on GPU device is the GPU-LMFit function [1]. GPU-LMFit uses a scalable parallel LM algorithm that Zhang and co-workers developed and optimized for using NVIDIA CUDA. Each fitting is processed by many GPU threads in parallel. In addition, a number of fittings using GPU-LMFit can execute concurrently in a single GPU device, and this makes an ideal method for efficient automated massive parametric imaging techniques, where the fitting procedure has to been repeated independently for each image pixel to fit the measured data in the pixel to a numerical model to resolve the parameters that are of interest. Another important feature of the architecture of GPU-LMFit is that **it can allow multiple GPUs applications**, where the measured experimental data in the host computer memory is separately passed to the global memories of multiple GPUs, and then the host program launches the kernel functions on each GPU device. Therefore, **the multiple GPUs application can further improve the efficiency of the LM model fittings** with GPU-LMFit, and enable real time automated massive parametric NMR analyses.

Results

Data: in vitro healthy mouse brain, fixed in paraformaldehyde and stored in PBS, scanned at 7.0 T (BRUKER Biospec) . An imaging version of PGSTE sequence was performed withh TE/TR = 25.77/4000 ms, Δ/δ= 40/2 ms, NA = 14; 16 axial slices with STH= 0.75mm, FOV=6cm, matrix 128x128 with in plane resolution of 470μm^2, 11 b-values ranging from 100 to 9000s/mm^2 along 30 no-coplanar directions plus 5 b=0s/mm2 images was acquired. Therefore, taking into account that mouse brain cover less than a half of the full image, the **maximum number of fittings** to be performed is about **$32x32x30x16 \sim 5x10^5$**.
A Nvidia Quadro K2000 with 2Gb of dedicated memory, which supports 1024 threads per block, with a maximum number of 64 registers per thread, was used for the analysis. Choosing how to distribute these threads in blocks of size Q affects performance. Together win Zhang, we choose to distribute the computation in **8 blocks** with **grid dimension (4096, 1, 1)**.
The analyzed **set of DW-NMR images** was of **121.521 Mb**. The **total used global memory** on GPU was between **7 and 15 Mb**, depending on the number of voxels per analyzed slice, which can vary a lot.
The average **speed** was about **12000 fits/second**, with respect to about **50 fits/second** of a multi-core CPU Intel Xeon E5430 @2.66GHz (4 cores, 8 threads). This means a **speed-up factor** of about **240**.
In terms of processing time, to obtain a full non-Gaussian diffusion map of the mouse brain, **GPU computing** takes about **40 seconds** against about **2.7 hours** of **multi-core CPU computing**.

[1] Zhu, X., & Zhang, D. (2013). Efficient Parallel Levenberg-Marquardt Model Fitting towards Real-Time Automated Parametric Imaging Microscopy. *PloS one*, *8*(10), e76665.