Estimation of GPU acceleration in NMR medical imaging reconstruction for realtime applications

Marco Palombo[1,2,3], *Matteo Bauce*[4], *Andrea Messina*[4], *Gianluca Lamanna*[5,6], *Silvia Capuani*[1,4]

[1]IPCF-UOS Roma, Physics Department, "Sapienza" University of Rome, Rome, Italy
[2]CEA/DSV/I2BM/MIRCen, Fontenay-aux-Roses, France
[3]CEA-CNRS URA 2210, Fontenay-aux-Roses, France, France
[4]Physics Department, "Sapienza" University of Rome, Rome, Italy
[5]INFN, Pisa Section, Pisa, Italy,
[6]INFN, Frascati Section, Frascati (Rome), Italy

DOI: http://dx.doi.org/10.3204/DESY-PROC-2014-05/37

Introduction. Recently new Nuclear-Magnetic-Resonance (NMR) methods have been developed to measure physical parameters that are directly correlated with properties of biological tissues, such as the Diffusional Kurtosis Imaging (DKI) [1] and the Stretched Exponential Model Imaging (STREMI) [2]. Such methods are based on non-Gaussian diffusion measurements. These techniques are particularly interesting because they allow to increase the sensitivity and specificity of the NMR imaging for healthy and pathological cerebral conditions [3, 4, 5, 6].
The DKI is essentially based on the assumption that DW-NMR signal can be described by the following relation [1]:

$$\frac{S(b)}{S(0)} = \left\{ \left[\exp\left(-bD_{app} + b^2 A_{app} \right) \right]^2 + \eta^2 \right\}^{1/2} \tag{1}$$

where η is the background noise, $A_{app} = \frac{1}{6} D_{app}^2 K_{app}$, with D_{app} and K_{app} respectively the apparent diffusion coefficient and the apparent diffusional kurtosis, estimated in the direction parallel to the orientation of diffusion sensitizing gradients, and $b = (\Gamma \delta g)^2 \Delta_{eff}$ with Γ the nuclear spin gyromagnetic ratio, g the diffusion sensitizing gradient strength, δ the diffusion sensitizing gradient duration and Δ_{eff} the effective diffusion time, depending on the particular diffusion sensitized sequence used.
On the contrary, the STREMI assumes the following relation to describe the signal decay [2, 7]:

$$\frac{S(b)}{S(0)} = \exp\left[-(b)^{\gamma} A \right] \tag{2}$$

where A is the generalized diffusion coefficient and γ is the stretching exponent, being between 0 and 1.
Currently, the post-processing of NMR images based on these new techniques requires a too long time (2 hours on a multi-core Intel Xeon E5430 CPU) for any use in realtime diagnostics. In this contribution we focused on the application of graphics processing units (GPUs) accelerated computing to improve the speed in reconstruction of diffusion weighted nuclear magnetic

resonance (DW-NMR) images by using non-Gaussian diffusion models. The aim of this work is to use model-related numerical simulations of DW-NMR signal in realistic conditions, to estimate the possible speed-up factor achievable by using GPU computing for non-Gaussian diffusion mapping.

Methods. *Synthetic DW-NMR data generation.* Brain regions characterized by highly coherent axonal bundles, with different geometrical organization were considered to simulate DW-NMR signal from water molecules diffusing within a realistic medium. In human brain such a region can be identified by the Corpus Callosum (CC) that, according to the estimated axonal diameter distributions within it, can be mainly subdivided into two regions: the first, characterized by denser and smaller axons, includes the genu and splenium (GS-CC); the other, characterized by less dense and bigger axons, includes the body (B-CC).
To simulate DW-signal, a Monte-Carlo (MC) simulation was implemented in C++. A total of 10^4 point like spins were randomly placed in a 2D plane of 0.5x0.5 mm^2, resembling a voxel of the DW-NMR image dataset. A random walk at a rate $\Delta t \sim 2.5\text{x}10^{-5}$ s per step, with bulk diffusivity (D_0) set to $1.4\text{x}10^{-3}$ mm^2/s and particle step size $\Delta x = (4D_0\Delta t)^{1/2}$ was performed between randomly packed axons. Axonal diameter distribution and density were numerically reproduced within each voxel of the synthetic image. Specifically, the chosen mean axon diameter $\pm SD$ and axon density percentage were: 2 ± 0.5 μm, ~ 0.50, for G-CC and S-CC; 6 ± 1.5 μm, ~ 0.35 for B-CC. Assuming axons as infinitely long coaxial cylinders, the DW-NMR signal decay for each voxel of the synthetic image due to a Pulsed Field Gradient Stimulated Echo sequence (PGSTE) was simulated through spin phase accumulation. The DW-NMR signal for the whole image was obtained by averaging the signal from each voxel. The parameters of the sequence were chosen to be similar to those of experiments we performed and report elsewhere [5]. We simulated different synthetic images, at different resolutions: 32x32, 64x64, 128x128, 256x256, 512x512, 1024x1024 and 2048x2048, to test the performance of CPU and GPU in analyzing images at different resolutions using a voxel-by-voxel fitting approach.
Synthetic DW-NMR data analysis. DKI and STREMI metrics were estimated by fitting on a voxel-by-voxel basis Eqs.(1) and (2) to the synthetic DW image signal intensities, respectively. This procedure was performed for all the image resolution investigated.
Here we used an efficient and robust fitting algorithm, named *GPU-LMFit*, based on a highly parallelized Levenberg-Marquardt (LM) method on GPU, introduced and described in details elsewhere [8]. *GPU-LMFit* uses a scalable parallel LM algorithm optimized for using the Nvidia CUDA platform. The code kernel calls *GPU-LMFit* to perform the LM algorithm on each CUDA block, which is mapped to a single voxel. Because the processing of different voxels is totally independent, the CUDA blocks do not need to synchronize, and the kernel launches as many blocks as voxels contained in a particular slice to speed up performance. The code was optimized to be fully integrated within Matlab (The Mathworks, Natick, MA, USA) scripts. A multi-core central processing unit (CPU) Intel Xeon E5430 processor at 2.66GHz with 8 thread, an Nvidia GPU GeForce GT650m and an Nvidia GPU Titan were used for the analysis and the cross-comparison of CPU and GPU performance. In particular, *lsqcurvefit* function with Parallel Computing Toolbox was used to test multi-core CPU performance.

Results and Discussion. An example of two simulated voxels with realistic geometry and local magnetic field inhomogeneities; the resulting DW-NMR signal and the fitted curves is reported in Figure 1-a), b) and c). The results of the performance test, obtained for each

synthetic image resolution, is instead reported in Figure 1-d).

Numerical results reported in Figure 1-d) suggest that for typical clinical images, whose resolution ranges from 64x64 to 256x256, an expected speed-up factor of $\sim$ 100x (for the Nvidia GeForce GT650m) and $\sim$ 1000x (for the Nvidia Titan) with respect to CPU performance is achievable by using massive parallel GPU computing to perform non-linear fitting of non-Gaussian diffusion models to DW-NMR dataset. Despite the results presented here are based on a simplistic simulation, where several effects including noise are neglected, they are in good agreement with experimental results. In real experiments, noise effects is not negligible, specially at high b-values, and can decrease the performance of both the CPU and GPU algorithms used. Indeed, high noise fluctuations in experimental data may introduce many spurious local minima in the likelihood function to be minimized in fitting routine. This implies that conventional fitting pipeline, based on LM algorithm, often fail in finding the global minimum, becoming strongly dependent on the fitting parameters initialization. Further investigations are therefore in progress in order to optimize the LM based fitting algorithms to make them less sensitive to noise fluctuations.

Conclusion. In this contribution we focused on the application of GPUs in the reconstruction of DW-NMR images based on non-Gaussian diffusion models. By using model-related numerical simulations, the performances of LM based fitting algorithm on CPU and GPU were tested for synthetic images at different resolutions and on two different GPUs: a low and a high-level Nvidia GPU. Our numerical results suggest that the implementation of LM algorithm on GPU makes it excellent for extensive GPU-based applications such as massive MRI processing, further improving the efficiency of the conventional LM model fittings on CPU. Specifically, an expected speed-up factor of $\sim$ 100x (for the Nvidia GeForce GT650m) and $\sim$ 1000x (for the Nvidia Titan) with respect to CPU (Intel Xeon E5430) performance is achievable by using massive parallel GPU computing. These results strongly suggest the GPU computing as a powerful tool for enabling automated parametric non-Gaussian DW-NMR analysis in realtime.

Acknowledgements

This work was partially supported by MIUR Futuro in Ricerca 2012 grant N. RBFR12JF2Z.

References

[1] J.H. Jensen *et al.*, Magn. Reson. Med. **53 (6)** 1432 (2005).

[2] S. De Santis, *et al.*, Magn. Reson. Med. **65 (4)** 1043 (2010).

[3] M. Palombo, *et al.*, J. Magn. Reson. **216** 28 (2012).

[4] J. GadElkarim, *et al.*, IEEE J. Emerg. Sel. Top. Circ. Syst. **3** 432 (2013).

[5] M. Palombo, *et al.*, Magn. Reson. Med.(2014), DOI: 10.1002/mrm.25308.

[6] S. Capuani, *et al.*, Magn. Reson. Imag. **31** 359 (2013).

[7] M. Palombo, *et al.*, J. Chem. Phys. **135 (3)** 034504 (2011).

[8] X. Zhu , D. Zhang, PloS one **8(10)** e76665 (2013).

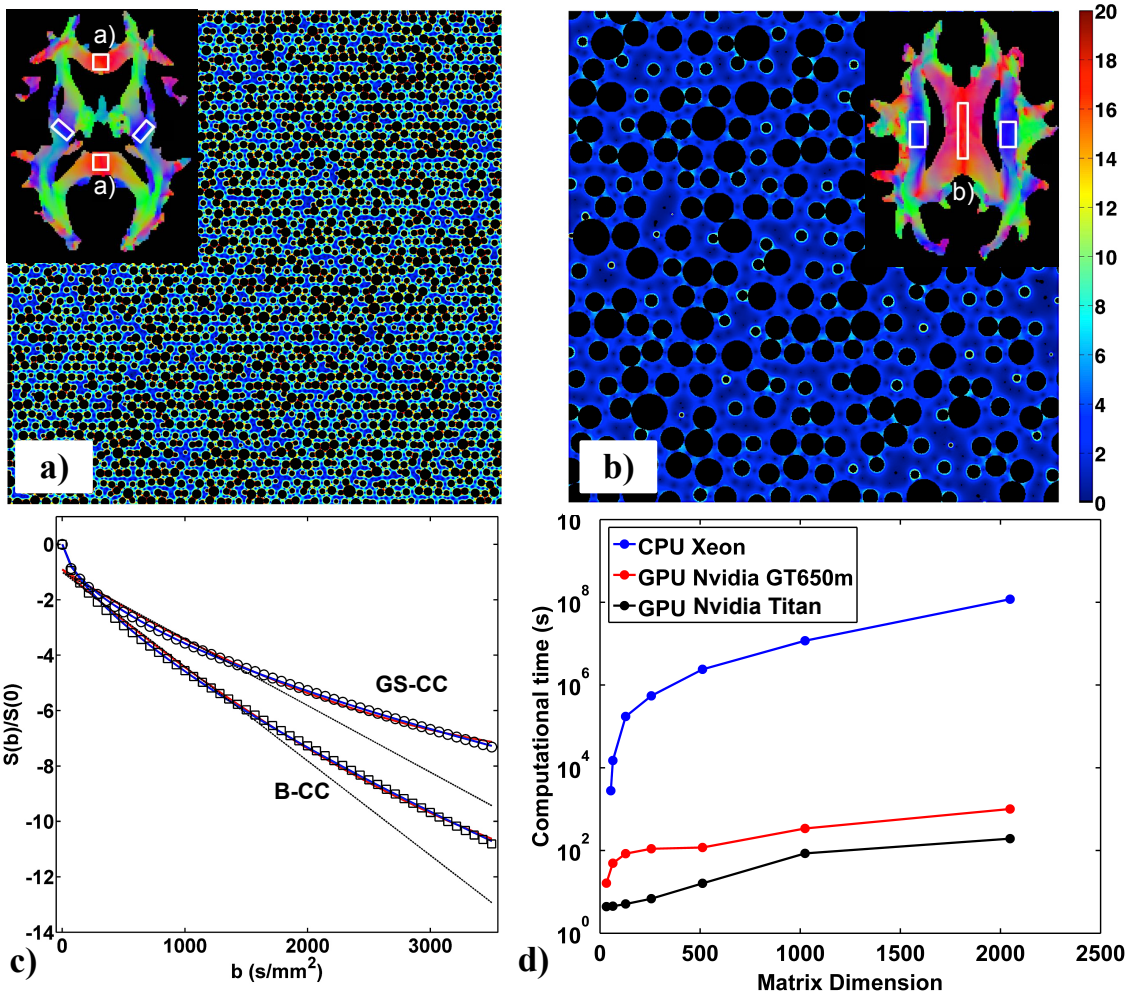

Figure 1: a-b) Local axonal geometry and magnetic field inhomogeneities strength spatial distribution computed with $\Delta\chi^{H_2O-TISSUE} = -0.010$ ppm (in IS) and static magnetic field of 3.0 T, for a representative voxel selected within GS-CC and B-CC, respectively, depicted in the WM maps reported in insets. c) S(b)/S(0), as a function of b, for the two representative voxels in GS-CC (circles) and B-CC (squares). Straight lines represent Eq.(1) (red) and (2) (blue), fitted to the simulated data. As comparison, dotted black lines represent the curves of equation $-bD_{app}$, with D_{app} values estimated by the fitting procedures to the data until $b \leq 2000$ s/mm^2. d) Performance test of the low-level, high-level Nvidia GPUs and high performance CPU described in the main text, for different values of image resolutions (matrix dimension).