GPU-parallelized model fitting for realtime non-Gaussian diffusion NMR parametric imaging

Marco Palombo[1,2,3], *Dianwen Zhang*[4], *Xiang Zhu*[5], *Julien Valette*[2,3], *Alessandro Gozzi*[6], *Angelo Bifone*[6], *Andrea Messina*[7], *Gianluca Lamanna*[8,9], *Silvia Capuani*[1,7]

[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]ITG, Beckman Institute, UIUC, Urbana, Illinois, United States
[5]College of Economics and Management, CAU, Beijing, China
[6]IIT, Center for Neuroscience and Cognitive Systems @ UniTn, Rovereto, Italy
[7]Physics Department, "Sapienza" University of Rome, Rome, Italy
[8]INFN, Pisa Section, Pisa, Italy,
[9]INFN, Frascati Section, Frascati (Rome), Italy

DOI: http://dx.doi.org/10.3204/DESY-PROC-2014-05/36

The application of graphics processing units (GPUs) for diffusion-weighted Nuclear-Magnetic-Resonance (DW-NMR) images reconstruction by using non-Gaussian diffusion models is presented. The image processing based on non-Gaussian models (Kurtosis and Stretched Exponential) currently are time consuming for any application in realtime diagnostics. Non-Gaussian diffusion imaging processing was implemented on the massively parallel architecture of GPUs, by employing a scalable parallel Levenberg-Marquardt algorithm (*GPU-LMFit*) optimized for the Nvidia CUDA platform. Our results demonstrate that it is possible to reduce the time for massive image processing from some hours to some seconds, finally enabling automated parametric non-Gaussian DW-NMR analyses in realtime.

1 Introduction

In this contribution we focused on the application of graphics processing units (GPUs) accelerated computing in reconstruction of diffusion weighted Nuclear-Magnetic-Resonance (DW-NMR) images by using non-Gaussian diffusion models, such as the diffusional kurtosis imaging (DKI) [1] and the stretched exponential model imaging (STREMI) [2], which allow to increase the sensitivity and specificity of the DW-NMR maps in healthy [3, 4, 5] and pathological subjects [6, 7]. However, the post-processing of DW-NMR images based on these models currently requires too long times for any realtime diagnostics. Typically, for the elaboration of diffusion maps, 10^6-10^7 voxels have to be managed. For each voxel a typical algorithm used to obtain non-Gaussian maps calculates at least three parameters by non-linear functions optimization. This is computationally demanding and takes some hours on recent multi-core processors(i.e. CPU Intel Xeon E5 and E7) to obtain a whole brain map. The aim of this work is to implement non-Gaussian diffusion imaging processing on the massively parallel architecture of GPUs and

optimize different aspects to enable on-line imaging.
To achieve this goal, diffusion images were acquired in a fixed mouse brain and two different algorithm to reconstruct DKI and STREMI maps were implemented on GPU. Successively, diffusion images were processed and non-Gaussian diffusion maps were obtained by using both the conventional currently used algorithm working on CPU and the new algorithms, based on a highly parallelized Levenberg-Marquardt (LM) method on GPU.
More specifically, we are concerned with a model-based approach for extracting tissue structural information from DW-NMR imaging data. Non-linear relations describing the DW-NMR signal attenuation have to be fitted to experimental dataset voxel-by-voxel by using the LM algorithm.
Certain features of the proposed implementation make it a good candidate for a GPU-based design: a) Independence between voxels across the three-dimensional brain volume allows voxel-based parallelization, b) Within each voxel, certain computation steps of data analysis are intrinsically iterative and independent, allowing further parallelization (i.e. fitting parameters estimation), c) Relatively simple mathematical operations are needed and these can be handled effectively by the GPU instruction set and d) Memory requirements are moderate during each step of the algorithm.

2 Theory

Non-Gaussian DW-NMR models. Describing water molecules displacement with a Gaussian Motion Propagator, NMR signal decay recorded using a diffusion-sensitized sequence may be expressed as a b-value function according to the following equation [8]:

$$S(b) = S(0)\exp(-bD) \tag{1}$$

where D is the apparent diffusion coefficient 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. In 3D space, observing diffusion displacement in a generic direction and working in an anisotropic environment, D is no more a scalar, but a tensor (DT) and the coupling of non diagonal terms in DT has to be taken into account. Nevertheless, due to proprieties of Fourier Transform and Gaussian Propagator, the mono-exponential form of the signal decay can be conserved by introducing the so called b-factor [9]:

$$\ln\left(\frac{S(b)}{S(0)}\right) = -\sum_{i,j=1}^{3} b_{i,j}D_{i,j} \tag{2}$$

where $b_{i,j} = (\Gamma\delta)^2\Delta_{eff}\int_0^{t_{seq}} dt(\int_0^t dt' g(t')g(t')^T)_{i,j}$ is the correspondent term of b-factor matrix to the relative term of DT. Due to its propriety, DT is diagonalizable to obtain scalar invariant quantities as MD and FA and the reference frame in which DT is diagonal.
The formalism exposed is based on the assumption that molecular diffusion occurs in a homogeneous environment, implying a linear relationship between mean square displacement and diffusion time. However, in a complex system with pores and traps on many length scales, this simple relation is lost and Eqs.1 and 2 are no longer valid [10, 11]. We refer to these cases as "non-Gaussian" diffusion process. In the last decade several different approaches have been

introduced in NMR field to describe DW-NMR signal decay in case of non-Gaussian diffusion. Here we treat two of the most employed ones: (i) the DKI [1] and (ii) the STREMI [2, 10].
(i) DKI is based on the assumption that DW-NMR signal can be described by the following relation [1]:

$$\frac{S(b)}{S(0)} = \left\{ \left[\exp\left(-bD_{app} + b^2 A_{app} \right) \right]^2 + \eta^2 \right\}^{1/2} \tag{3}$$

where η is the background noise, and $A_{app} = \frac{1}{6} D_{app}^2 K_{app}$, with D_{app} and K_{app} the apparent diffusion coefficient and the apparent diffusional kurtosis, estimated in the direction parallel to the orientation of diffusion sensitizing gradients, respectively.
(ii) The STREMI assumes the following relation to describe the signal decay [2, 10]:

$$\ln(\frac{S(b)}{S(0)}) = -\sum_{i=1}^{3} (b_i^*)^{\gamma_i} A_i \tag{4}$$

where A_i is the generalized diffusion coefficient estimated along the direction identified by the ith eigenvector of DT, named $\vec{\epsilon}_i$; γ_i is the stretching exponent estimated along the ith direction of DT reference frame (being between 0 and 1); and b_i^* is the projection of bvalue along $\vec{\epsilon}_i$ (with components ϵ_{1i}, ϵ_{2i}, ϵ_{3i}, in the laboratory reference frame), i.e. $b_i^* = \vec{b} \cdot \vec{\epsilon}_i$.

3 Materials and Methods

DW-NMR data acquisition. An *ex vivo* healthy mouse brain, fixed in paraformaldehyde and stored in PBS [12], was scanned at 7.0T (BRUKER Biospec). An imaging version of PGSTE sequence was performed with $TE/TR = 25.77/4000ms$, $\Delta/\delta = 40/2ms$, number of averages $NA = 14$; 16 axial slices with thickness $STH = 0.75mm$, field of view $FOV = 6cm$, matrix 128x128 with in plane resolution of $470\mu m^2$ were acquired with 10 b-values ranging from 100 to $8000s/mm^2$ along 30 no-coplanar directions plus 5 $b = 0s/mm^2$

DW-NMR data analysis. Both DKI parametric maps (K_{app}-maps) and STREMI parametric maps (γ-maps) were estimated in each direction parallel to the orientation of diffusion sensitizing gradient, but K_{app}-maps were obtained by fitting on a voxel-by-voxel basis Eq.(3) to the DW image signal intensities (for $b \leq 3000s/mm^2$), while γ-maps, were obtained by similar procedure, using Eq.(4).
Finally, the non-Gaussian diffusion parametric maps MK and $M\gamma$ were computed by averaging across the 30 directions in the corresponding K_{app}- and γ-maps, respectively.

GPU implementation. A modern GPU device can have a number of "multiprocessors" (MP), each of which executes in parallel with the others. Using the Nvidia compute unified device architecture (CUDA), multiple thread blocks (and thus multiple fittings) can execute concurrently with many parallel threads on one multi-processor. Here we implemented an efficient and robust fitting algorithm, based on a highly parallelized LM method on GPU. The LM algorithm is based on an iterative numerical optimization procedure that minimizes the sum of squared model residuals. The function we used to perform LM fittings on GPU device is the single precision *GPU-LMFit* function, introduced and described in details elsewhere [13]. *GPU-LMFit* uses a scalable parallel LM algorithm optimized for using the Nvidia CUDA platform.

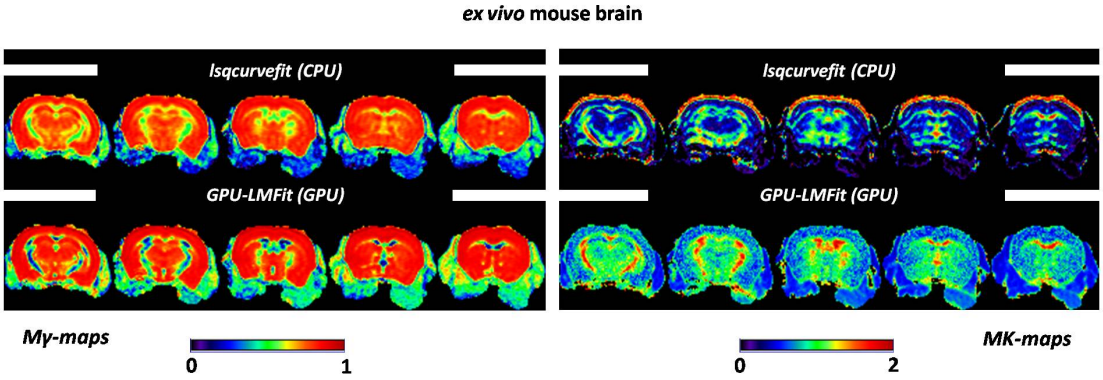

Figure 1: Comparison between CPU and GPU non-Gaussian diffusion maps reconstruction. Reported maps were obtained by using the *lsqcurvefit* on Intel Xeon E5- E5430 CPU and *GPU-LMFit* on Nvidia Quadro K2000 GPU (see Table 1 for details).

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, and a Nvidia GPU Quadro K2000, with 2Gb of dedicated memory, supporting 1024 threads per block and a maximum number of 64 registers per thread, 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.
Each analyzed DW-NMR image dataset is of $\sim$ 150 Mb, and requires that the total number of fittings to be performed to create a single image of the non-Gaussian parametric map is in the range of $(0.5 - 5)\text{x}10^6$. In the kernel function, we choose to distribute the computation in 4096 CUDA blocks, each of which has 8 threads concurrently executing to compute a fitting. Therefore, the kernel function was called multiple times in the program to complete all fittings for an image, and each call to this function requires 7-15 Mb global memory on GPU.

4 Results and Discussion

Non-Gaussian diffusion parametric maps $M\gamma$ and MK, obtained by using *lsqcurvefit* on multiple CPU threads and *GPU-LMFit on GPU*, are displayed in Figure 1. The specific performances of the CPU and GPU employed are reported in Table 1, while the cross-comparison between *lsqcurvefit* and *GPU-LMFit* results is reported in Figure 2.

The Figures 1 and 2 show that the GPU approach for STREMI is in agreement with conventional CPU one. Conversely, for DKI, *GPU-LMFit* slightly overestimates MK values with respect to *lsqcurvefit*. However, it is important to note that MK-maps obtained by *GPU-LMFit*

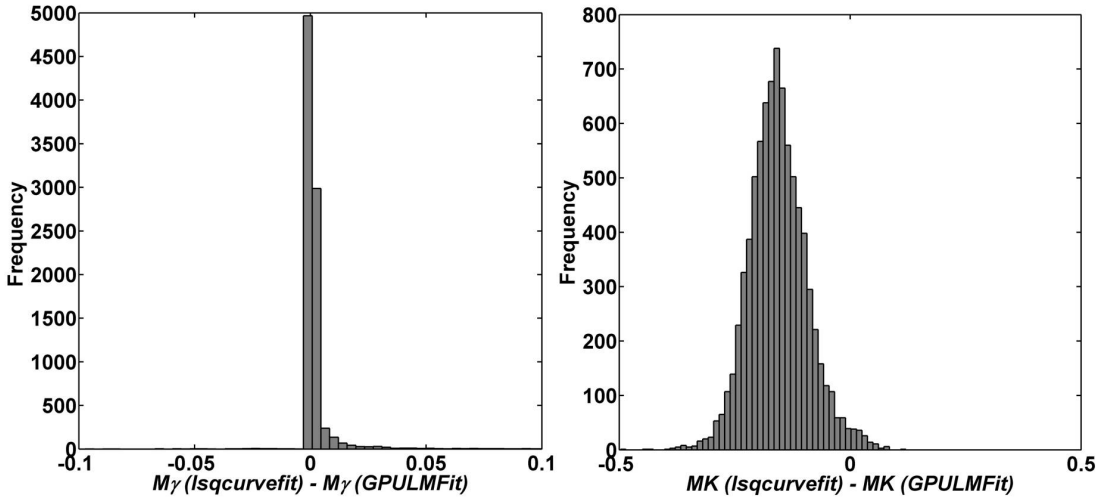

Figure 2: Cross-comparison between *lsqcurvefit* and *GPU-LMFit* results. The frequency histograms of the difference between $M\gamma$ (left) and MK (right) values derived with *GPU-LMFit* and *lsqcurvefit* are computed from all the voxels of the parametric maps comprised within the *ex vivo* mouse brain investigated.

show a contrast-to-noise ratio different from the *lsqcurvefit* ones (see Figure 1): they better discriminate between white and gray matter regions within mouse brain, even if they are more grainy and noisy than the *lsqcurvefit* ones. The grainy and noisy characteristic of these maps can be due to the single precision implementation of the *GPU-LMFit* function. Indeed, each K_{app}-map is obtained by fitting, voxel-by-voxel, Eq.3 to get the two constants D_{app} and A_{app}. Then, K_{app} is estimated by keeping the ratio: $K_{app} = \frac{6A_{app}}{D_{app}^2}$. It is therefore clear that using single or double precision operators in K_{app} estimation can make the difference. In particular, the *lsqcurvefit* is a double precision Matlab function, while the *GPU-LMFit* version used here is single precision. Therefore, further developments are actually in progress to release a double precision version of *GPU-LMFit*, in order to increase the quality of non-Gaussian diffusion maps obtained on GPU.

Finally, from Table 1 it is possible to appreciate the relative speed-up obtainable by using *GPU-LMFit*, which for a medium level GPU like the Quadro K2000 is $\sim$ 240x. In terms of computational time, this speed-up factor means that the GPU implementation reported here allows to reduce the time for massive image processing from some hours to some seconds (see

Performance Parameters	**CPU**	**GPU**
Average Speed (fit/sec.)	50	12000
Computational Time (sec.)	7200	30
Speed-up factor	1	240

Table 1: Comparison of CPU and GPU performance in non-Gaussian diffusion maps reconstruction.

Table 1). Moreover, numerical simulations were performed on high level Nvidia GPU, the Nvidia Titan, to test the additional speed-up factor obtainable by employing one of the most powerful GPU now available (results not reported here). Simulation results suggest that an additional speed-up factor of 6.6x with respect to the Nvidia Quadro K2000 GPU is achievable by using the Nvidia Titan GPU. This demonstrates that automated parametric non-Gaussian DW-NMR analysis in realtime is now really possible by using the GPU approach proposed in this work.

5 Conclusion

In this contribution we focused on the application of GPUs in the reconstruction of DW-NMR images based on non-Gaussian diffusion models. This application can benefit from the implementation on the massively parallel architecture of GPUs, optimizing different aspects and enabling online imaging. A pixel-wise approach by using a fast, accurate and robust parallel LM minimization optimizer, called *GPU-LMFit*, was implemented in CUDA and fully integrated in Matlab. Our results show that the GPU application proposed here can further improve the efficiency of the conventional LM model fittings, reducing the time for DW-NMR image-processing from hours to seconds, finally enabling automated parametric non-Gaussian DW-NMR analysis in realtime. Moreover, another important feature of the architecture of the proposed approach is that it can allow multiple GPUs applications [13], 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, another natural development of this work, behind the upgrade to the double precision version, is the multiple GPUs application in order to further improve the efficiency of the LM model fittings with *GPU-LMFit*.

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] F. Grinberg, *et al.*, PloS one **9(2)** e89225 (2014).
[8] E.O. Stejskal, J.E. Tanner, J. Chem. Phys. **42(1)** 288 (1965).
[9] P.J. Basser, *et al.*, Biophys. J. **66(1)** 259 (1994).
[10] M. Palombo, *et al.*, J. Chem. Phys. **135 (3)** 034504 (2011).
[11] M. Palombo, *et al.*, Nature Sci. Rep. **3** 2631 (2013).
[12] L. Dodero , *et al.*, PloS one **8(10)** e76655 (2013).
[13] X. Zhu , D. Zhang, PloS one **8(10)** e76665 (2013).