GPU imaging analysis for ultra-fast non-Gaussian diffusion mapping

Marco Palombo[1,2], Dianwen Zhang[3], Chen Zhu[4], Julien Valette[1], Alessandro Gozzi[5], Angelo Bifone[5], Andrea Messina[6], Gianluca Lamanna[7], and Silvia Capuani[6,8]
[1]CEA/DSV/I2BM/MIRCen, Fontenay-aux-Roses, France, France, [2]IPCF-UOS Roma, Phys. Dpt., Sapienza University, Rome, Rome, Italy, [3]ITG, Beckman Institute, UIUC, Urbana, Illinois, United States, [4]College of Economics & Management, CAU, Beijing, China, [5]IIT, Center for Neuroscience and Cognitive Systems @ UniTn, Rovereto, Italy, [6]Physics Dpt., Sapienza University, Rome, Italy, [7]INFN, Pisa Section, Pisa, Italy, [8]IPCF-UOS Roma, Phys. Dept., Sapienza University, Rome, Italy

PURPOSE. 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 mapsin detecting several pathological conditions[3,4]. However, the post-processing of DW-NMR images based on these models currently requires too long times for any application in real-time diagnostics. Typically, for the elaboration of these images, 10^6-10^7voxels have to be managed. For each voxel the algorithm calculates at least 3 parameters by non-linear functions optimization. This is computational demanding and takes some hours on recent multi-core processors(i.e. CPU Intel Xeon E5 and E7) to obtain a 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 online imaging.

METHODS. *Code 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 Levenberg-Marquardt (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 *GPU-LMFit* function[5]. *GPU-LMFit* uses a scalable parallel LM algorithm optimized for using the Nvidia CUDA platform. Here we used two versions of the *GPU-LMFit* function: a single-precision (S) and a double-precision (D) version. 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. Two different multi-core central processing unit (CPU) configurations and three different Nvidia GPUs were used for the analysis and the cross-comparison of CPU and GPU performance (see **Table1**). In particular, *lsqcurvefit* function with Parallel Computing Toolbox was used to test multi-core CPU performance. *DW-NMR data acquisition*: an *in vivo* and an *ex vivo* healthy mouse brain, fixed in paraformaldehyde and stored in PBS, was scanned at 7.0 T (BRUKER Biospec). An imaging version of PGSTE sequence was performed with 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}were acquired with10 b-values ranging from 100 to 8000s/mm^2 along 30 no-coplanar directions plus 5 b=0s/mm^2. *DW-NMR data analysis:* parametric maps of kurtosis metrics, i.e. K_{app}-maps, were obtained by fitting on a voxel-by-voxel basis the following relationto the DW image signal intensities (for $b \leq 3000 s/mm^2$): $S(b)/S(0)=\exp(-bD_{app}+1/6b^2D_{app}^2K_{app})$, where D_{app} and K_{app} are the apparent diffusion coefficient and kurtosis, estimated in the direction parallel to the orientation of diffusion sensitizing gradients, respectively. STREMI parametric maps, i.e. γ-maps, were obtained by similar procedure, using the following relation: $S(b)/S(0)=\exp[-(D_{app}b)^{\gamma}]$, where γ is the stretching parameter, being between 0 and 1.Finally, the non-Gaussian diffusion parametric maps Mγ and MK were computed by averaging across the 30 directions in the corresponding γ and K_{app}-maps, respectively. The analyzed set of DW-NMR images was of ~ 100-200 Mb. The total number of fittings to be performed was in the range of$(0.5\text{-}5)\times10^6$.

RESULTS & DISCUSSION. Non-Gaussian diffusion parametric maps Mγ and MK, obtained by using *lsqcurvefit* on multiple CPU threads and *GPU-LMFit* on GPU, are displayed in **Figure1**. The specific performances of each CPU configuration and GPU employed are reported in **Table1**, and the cross-comparison between *lsqcurvefit* and *GPU-LMFit* results is reported in **Figure2**. From **Figures 1** and **2** it is evident that the GPU approach for STREMI is in perfect agreement with conventional CPU one. On the contrary, for DKI, *GPU-LMFit* slightly overestimates MK values with respect to *lsqcurvefit*. However,it is important to note that MK-maps obtained with *GPU-LMFit* show a better contrast-to-noise ratio than the *lsqcurvefit* ones. Finally, from **Table1** it is possible to appreciate the relative speed-up obtainable using *GPU-LMFit*, which for medium level GPUs is ~100x but for high level ones can reach a factor ~1000x. This means that the GPU implementation reported here allowsto reduce the time for massive image processing from some hours to some seconds (see **Table1**).

CONCLUSION. The proposed implementation of LM algorithm on GPU makes it excellent for extensive GPU-based applications such as massive MRI processing. Our results show that the GPU application proposed here can further improve the efficiency of the conventional LM model fittings, finally enabling automated parametric non-Gaussian DW-NMR analysesin real-time.

REFERENCES: [1]Jensen JH, et al. *MagnReson Med* 2005; 53(6):1432-1440.[2]De Santis S, et al. *MagnReson Med* 2011; 65(4):1043-1052.[3]Grinberg F et al. *PloS one* 2014; 9(2):e89225.[4]Kamagata Koji et al. *Neuroradiology* 2014; 56(3): 251-258.[5]Zhu X, Zhang D *PloSone* 2013; 8(10):e76665.

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

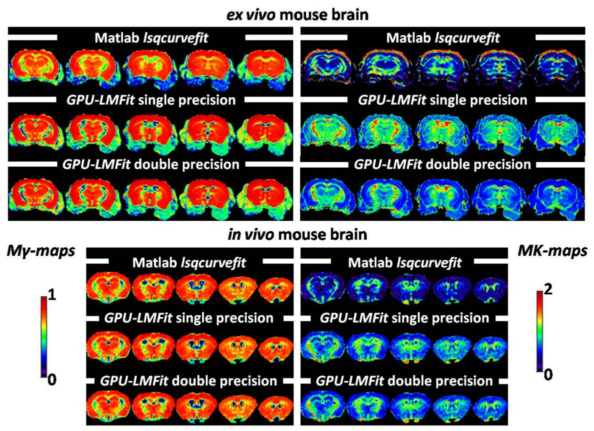
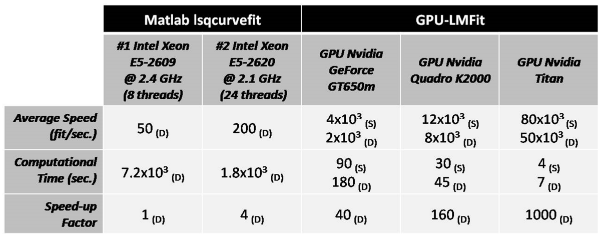

ex vivo mouse brain
Matlab *lsqcurvefit*
Matlab *lsqcurvefit*
GPU-LMFit single precision
GPU-LMFit single precision
GPU-LMFit double precision
GPU-LMFit double precision
in vivo mouse brain
Mγ-maps
MK-maps
Matlab *lsqcurvefit*
Matlab *lsqcurvefit*
1
0
2
0
GPU-LMFit single precision
GPU-LMFit single precision
GPU-LMFit double precision
GPU-LMFit double precision

Figure 1 Comparison between CPU and GPU non-Gaussian diffusion maps reconstruction. Reported maps were obtained by using the *lsqcurvefit* on Intel Xeon E5-2620 CPUs and *GPU-LMFit*on NvidiaQuadro K2000 GPU (see **Table1** for details).



	Matlab lsqcurvefit		GPU-LMFit		
	#1 Intel Xeon E5-2609 @ 2.4 GHz (8 threads)	*#2 Intel Xeon E5-2620 @ 2.1 GHz (24 threads)*	*GPU Nvidia GeForce GT650m*	*GPU Nvidia Quadro K2000*	*GPU Nvidia Titan*
Average Speed (fit/sec.)	50 $_{(D)}$	200 $_{(D)}$	4×10^3 $_{(S)}$ 2×10^3 $_{(D)}$	12×10^3 $_{(S)}$ 8×10^3 $_{(D)}$	80×10^3 $_{(S)}$ 50×10^3 $_{(D)}$
Computational Time (sec.)	7.2×10^3 $_{(D)}$	1.8×10^3 $_{(D)}$	90 $_{(S)}$ 180 $_{(D)}$	30 $_{(S)}$ 45 $_{(D)}$	4 $_{(S)}$ 7 $_{(D)}$
Speed-up Factor	1 $_{(D)}$	4 $_{(D)}$	40 $_{(D)}$	160 $_{(D)}$	1000 $_{(D)}$

Table 1 Comparison of CPU and GPU performance in non-Gaussian diffusion maps reconstruction. Different kind of hardware configurations were investigated and specified in the second row. Subscripts (S) and (D) indicate single- and double-precision, respectively.

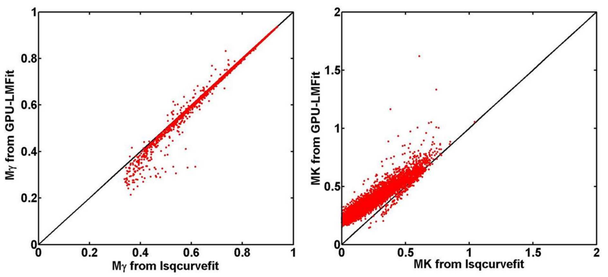


Figure 2 Cross-comparison between *lsqcurvefit* and *GPU-LMFit* results. *GPU-LMFit*(D) derived Mγ (left) and MK (right) values as a function of *lsqcurvefit* (D) derived ones, for the voxels of parametric maps within mouse brain (for both *in vivo* and *ex vivo*). The straight line represents perfect correspondence.