gpuNUFFT - An open source GPU library for 3D regridding with direct Matlab interface

Florian Knoll[1], Andreas Schwarzl[2], Clemens Diwoky[2], and Daniel K Sodickson[1]
[1]*Bernard & Irene Schwartz Center for Biomedical Imaging, Department of Radiology, NYU School of Medicine, New York, New York, United States,* [2]*Institute of Medical Engineering, Graz University of Technology, Graz, Styria, Austria*

Target Audience: Researchers and clinicians interested in 3D non-Cartesian image reconstruction.

Purpose: 3D non-Cartesian trajectories are very attractive for iterative image reconstruction [1], and especially in compressed sensing [2], because of incoherent aliasing in the case of undersampling. However, image reconstruction is still challenging due to prohibitively expensive computation times. The computational bottlenecks are usually gridding and inverse gridding, which have to be evaluated in each iteration. Parallel implementations using graphics processing units (GPUs) have evolved as a practical, inexpensive and computationally effective way to tackle this problem [3]. There are already efficient and powerful software packages available within the research community [4-8] but in many cases they are restricted to the case of 2D trajectories, and they are usually integrated in larger software packages for individual reconstruction frameworks of the authors and also use their own conventions for data I/O. This makes re-use of modules from these packages challenging, especially when considering that the large majority of code for image reconstruction is written in Matlab (The MathWorks, Natick, MA). The goal of this work is to introduce gpuNUFFT, a new open-source 3d regridding GPU library with a built-in Matlab interface that is straightforward to include in all implementations of iterative image reconstruction.

Methods: Code design and implementation: While the efficiency of the implementation was an important factor during the code design process for gpuNUFFT, the major focus was on ease of integration into existing reconstruction frameworks. As the Matlab NUFFT toolbox by Fessler et al. [9] is the de-facto standard for non-Cartesian image reconstruction, an interface was designed that allows the generation of operators in exactly the same way. The same conventions concerning the format of k-space trajectory, density compensation and parameters of the Kaiser-Bessel kernel were used. A platform independent CMAKE based compilation framework was generated that also takes care of the compilation of mexfiles that serve as the interface to Matlab.
The core of gpuNUFFT, the actual GPU implementation of regridding and inverse gridding, was realized using NVIDIAs Compute Unified Device Architecture (CUDA) [10]. The design was inspired by the threaded multicore CPU implementation in [11]. 3D k-space is divided into smaller subelements and calculations are performed in parallel both within and over these subelements. Memory latencies are reduced by using the GPUs shared memory as a cache within those subelements.

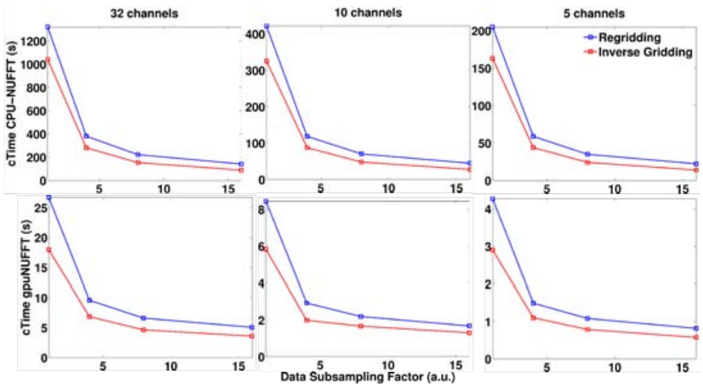

Figure 1: Computation times of regridding and inverse gridding using the CPU reference implementation [10] (top) and gpuNUFFT (bottom). Experiments are shown for different numbers of 3D radial projections as defined by the subsampling factors (1,4,8,16) and different numbers of receive channels (32: left, 10: middle, 5: right).

Evaluation of reconstruction times: MR measurements of the brain of a healthy volunteer were taken with a 3D radial balanced UTE sequence described in [12] with sequence parameters TR=1.62ms, TE=0.1ms, FOV=210mm, matrix=160×160×160, FA=18°. A fully sampled scan consisted of 25582 radial 3D projections with scan time t=47s. To evaluate the performance of gpuNUFFT with respect to different data sizes, accelerated acquisitions with acceleration factors R=4 (6395 projections, t=12s), R=8 (3197 projections, t=6s) and R=16 (1582 projections, t=3s) were acquired as well. All scans were performed on a clinical 3T MR system (MAGNETOM Skyra, Siemens AG, Erlangen) with a 32-channel head coil array. In addition, reconstructions were also performed after SVD coil compression to 10 and 5 virtual channels. Computations were performed on a conventional desktop workstation (Intel Xeon X5647 2.93GHz, 8 cores, 12GB RAM, NVIDIA GTX680 GPU with 4GB of memory), using Matlab R2012b (64bit) on Linux Kernel 3.5.0-41. gpuNUFFT was compiled with CUDA 5.0.

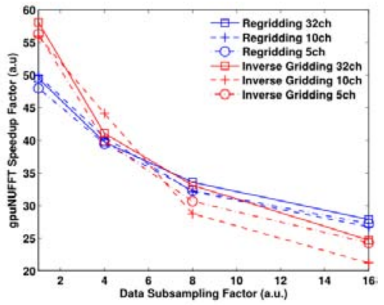

Figure 2: Speedups with gpuNUFFT corresponding to the computation times of the experiments in Figure 1.

Results: **Figure 1** shows a comparison of the computation times of the CPU NUFFT reference implementation from [10] and gpuNUFFT for gridding and inverse gridding with different combinations of undersampling factors and coil elements. The corresponding speedup factors that can be achieved with gpuNUFFT are shown in **Figure 2** and lie in the range between 20 and 60, with the general trend of higher GPU speedups for larger data sets. Regridding reconstruction results of a single transversal and a sagittally reformatted slice are shown in **Figure 3** for the fully sampled scan using all 32 channels. The scale of the difference images is 250 times higher for visualization purposes. Finally the gpuNUFFT forward and adjoint operators were tested in a Matlab implementation of CGSENSE [1]. Results are also shown in **Figure 3** for the case R=4 and 32 channels using 15 CG iterations. The corresponding total CGSENSE reconstruction time speedup factor with gpuNUFFT was 40, indicating the large amount of computation time required for gridding and inverse gridding.

Discussion and Conclusion: The goal of the distribution of the gpuNUFFT library, which will be available on our webpage (http://cai2r.net/resources/software) is to provide a tool for researchers who want to leverage the computing power of their GPUs to speed up their reconstructions while continuing to use already existing Matlab implementations tailored to their applications and without having to do any GPU computing themselves. In particular GPU regridding can be enabled with the change of just a single line of existing code (the generation of the NUFFT operator). The results from the experiments in this work led to speedups between factor of 20 and a factor of 60 in comparison to the well-established NUFFT toolbox by Fessler et al., with no visually perceptaible differences in the reconstructed images. In our experiments the largest speedups were observed for the largest datasets (largest number of coils and projections). The reason for this behavior is that in order to make efficient use of the massively parallel GPU architecture, the largest possible number of cores must be used at all times while minimizing the overhead of data transfer between Matlab and the GPU. Therefore the actual speedup factors that will be observed in different applications will depend on the type of trajectory, the matrix size and the percentage of total computation time required for regridding and inverse gridding in a particular implementation.

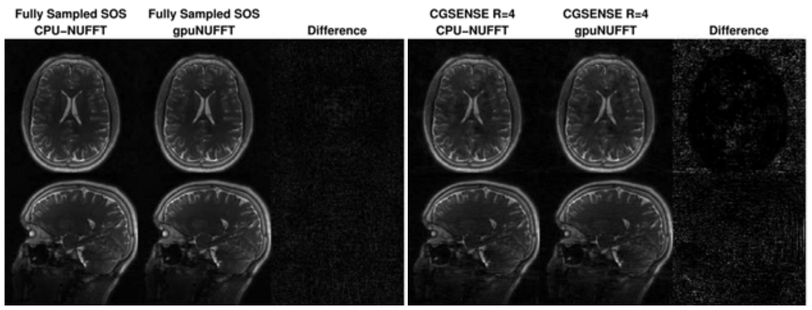

Figure 3: Left: Comparison of reconstruction quality of CPU reference implementation and gpuNUFFT for regridding of fully sampled data from 32 channels and SOS channel combination. Right: Comparison using iterative SENSE, 4 times undersampled data, 32 channels. Total computational speedup for CGSENSE gpuNUFFT was 40. The scale of the difference images is 250 times higher for visualization purposes.

References: [1] Pruessmann et al., MRM 46: 638-651 (2001), [2] Lustig et al., MRM 58: 1182-1195 (2007), [3] Sorensen et al., IEEE TMI 27: 538-547 (2008), [4] Hansen and Sorensen, MRM 69: 1768- 1776 (2013), [5] Murphy et al., Proc. ISMRM 2013: 2630, [6] Gai et al., Proc. ISMRM 2012: 2550, [7] Freiberger et al., CiSE 15: 34-44 (2013), [8] Uecker et al., MRM early view (DOI: 10.1002/mrm.24751) (2013), [9] Fessler and Sutton, IEEE TSP 51: 560-574 (2003), [10] NVIDIA Corporation, NVIDIA CUDA – Programming Guide v5.5 (July 2013), [11] Zwart et al., MRM 67: 701-710 (2012), [12] Diwoky et al., Proc ISMRM 2011: 327.