A GPU Implementation of Compressed Sensing Reconstruction of 3D Radial (Kooshball) Acquisition for High-Resolution Cardiac MRI

S. Nam[1,2], T. A. Basha[2], M. Akçakaya[2], C. Stehning[3], W. J. Manning[2], V. Tarokh[1], and R. Nezafat[2]
[1]SEAS, Harvard University, Cambridge, MA, United States, [2]Dept. of Medicine (Cardiovascular Division), Beth Israel Deaconess Medical Center, Harvard Medical School, Boston, MA, United States, [3]Philips Research, Hamburg, Germany

Introduction: 3D non-Cartesian sampling trajectories allow high isotropic spatial resolution, better depiction of cardiac anatomy and ease of image prescription in whole heart cardiac MRI. 3D acquisition using stack of spirals or radials with Cartesian sampling in k_z and 3D radial sampling (kooshball) have been previously used to image cardiac anatomy [1-4]. 3D kooshball imaging sequence allows respiratory self-gating, excellent motion and flow insensitivity and improved scan efficiency. However, these imaging techniques suffer from long scan and reconstruction time. While highly under-sampled 3D kooshball is more forgiving in terms of artifacts than Cartesian acquisition, the streaking artifacts could impact the clinical interpretation. Compressed sensing (CS) reconstruction [5] has the potential to remove these artifacts; however even a feasibility study of this approach with 3D non-Cartesian sampling has been limited mainly due to high computational burden associated with iterative CS reconstructions. In this study, we sought to implement and evaluate an iterative 3D CS reconstruction method for high-resolution kooshball whole heart imaging on Graphics Processing Units (GPU). The resulting speedup was compared with a sequential C++ implementation on CPU.

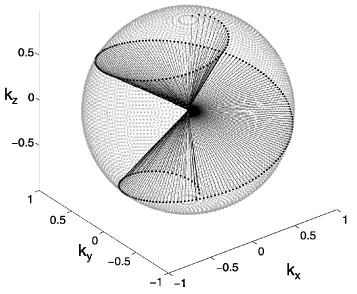

Fig. 1: *One interleave from the kooshball trajectory.*

Materials And Methods: The kooshball trajectory [6] has N_i interleaves and each interleave consists of N_p projections. Each interleave is the rotated version of the first interleave around the k_z-axis. **Figure 1** demonstrates the end points of the projection lines in one interleave. The acquired MR signals can be formulated in an encoding matrix format as $\mathbf{y} = \mathbf{A}\mathbf{x}$, where **y** denotes the acquired k-space samples, **x** denotes the desired image, and **A** denotes the encoding matrix. **A** can be regarded as taking the reverse steps of conventional gridding reconstruction without sampling density compensation: 3D FFT is first performed on the image and then the Cartesian k-space samples are re-gridded onto the 3D radial sample points. An image **x** is reconstructed iteratively from the undersampled k-space data by solving the minimization problem of $\arg\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$, where the l_1 norm of the image coefficient is used for regularization [7].

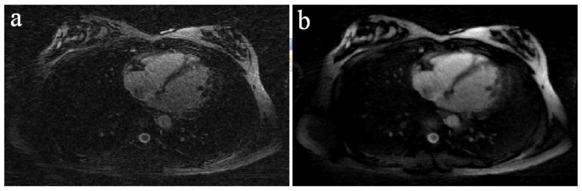

Fig. 2: *A slice from an axial view of the heart reconstructed with (a) conventional gridding, (b) iterative CS method. CS with l_1 minimization reduces the streaking artifacts with penalty of further image blurring.*

3D CS reconstruction algorithm for kooshball was implemented in CUDA environment (Intel Core2 Quad Q9400, 8.0 GB memory, Nvidia GeForce GTX 480, Windows 7 64bit OS). Main computational overhead of the iterative method comes from the multiplications of the large encoding matrix **A** and its conjugate $\mathbf{A}^H$, which consist of FFT/IFFT and convolution interpolation (gridding). As opposed to the case of stack-of-stars trajectory, the gridding cannot be performed separately in each k_x-k_y plane and this imposes a computational challenge because it is harder to localize the k-space data for parallelization. cuFFT and cuBLAS packages are used for FFT/IFFT and other arithmetic operations. In order to compare the reconstruction image quality, conventional gridding reconstructed image with density compensation was implemented in CUDA environment. The CS reconstruction was also implemented in standard C++ environment using FFTw package [8] for comparison of the reconstruction time. The thresholding parameter λ was manually selected to get the best image quality. 100 iterations were used in reconstruction.
Whole heart acquisition was performed using an SSFP sequence with N_i= 10 interleaves, N_p = 900 projections per interleave and N_s = 344 sample points per each projection which gives 1/13 of sampling density compared with Nyquist-sampled 3D Cartesian data. All images were acquired using 1.5T Philips (Achieva) with 5-channel cardiac coil array. The following parameters were used: TR/TE/α=4.2/2.1/60°, FOV=270×270×270mm^3, resolution=1.5×1.5×1.5mm^3. The acquisition time was approximately 3:28 minutes. The reconstructed image size is 344×344×344.

Result: **Figure 2** shows example slices from 3D images of the heart reconstructed using a) 3D conventional gridding and b) iterative CS reconstruction. The iterative reconstruction improves the image quality, suppressing the high frequency noise and streaking artifacts, however it results in further image smoothing due to l_1 minimization. **Table 1** shows the average time required for performing the main operations in the iterative reconstruction with CUDA and C++ implementations. The CUDA implementation yields 58× speedup compared to the C++ implementation.

Conclusion: GPU implementation significantly reduces the reconstruction time (58×) of 3D kooshball acquisition with CS, allowing clinical feasibility studies. This speed up facilitates further investigation in improving reconstruction algorithm for 3D kooshball and potentially other 3D non-Cartesian trajectories. Our current GPU technology does not allow for parallel coil reconstruction due to limited available GPU memory.

	FFT	IFFT	Gridding	Re-gridding	Thresholding	Misc.	Total
CUDA (sec)	0.2398	0.2395	0.5938	0.4211	0.0122	0.1262	**1.6326**
C++ (sec)	9.6546	9.6326	35.9027	34.2584	1.6052	3.8415	**94.8950**
speedup	40.3	40.2	60.5	81.4	131.6	N/A	**58.1**

Table 1. *Average time required for performing main operations in CS iterative reconstruction with CUDA and C implementations.*

Acknowledgements: The authors wish to acknowledge grant support from NIH R01EB008743-01A2, AHA SDG-0730339N, and Catalyst (Harvard Clinical and Translational Science Center).

References: [1] Dietrich O, proc. 18th ISMRM, p. 2887, 2010. [2] Wu HH, proc. 18th ISMRM, p. 2878, 2010. [3] Spincemaille P, proc. 18th ISMRM, p. 2899, 2010. [4] Stehning C, MRM, pp. 476-480, 2005. [5] Lustig M, MRM, pp. 1182-1195, 2007. [6] Wong ST, MRM, pp. 778-784, 1994. [7] Wright SJ, IEEE TSP, pp. 2479-2493, 2009. [8] http://www.fftw.org.