Cardiac MRI Compressed Sensing Image Reconstruction with a Graphics Processing Unit

Majid Sabbagh*
msabbagh@ece.neu.edu

Martin Uecker†‡
martin.uecker@med.uni-goettingen.de

Andrew J. Powell§¶
andrew.powell@cardio.chboston.org

Miriam Leeser*
mel@coe.neu.edu

Mehdi H. Moghari§¶
mehdi.hedjazimoghari@cardio.chboston.org

*Electrical and Computer Engineering Department, Northeastern University, Boston, MA
†Institute for Diagnostic and Interventional Radiology, University Medical Center Göttingen, Göttingen, Germany
‡German Centre for Cardiovascular Research (DZHK), Göttingen, Germany
§Department of Cardiology, Boston Children's Hospital, Boston, MA
¶Department of Pediatrics, Harvard Medical School, Boston, MA

***Abstract*—Compressed sensing (CS) magnetic resonance imaging (MRI) reconstruction reduces the scan time by undersampling the data but increases the image reconstruction time because a non-linear optimization problem must be iteratively solved to reconstruct the images. The growing demand for reducing the examination time in cardiac MRI led us to investigate opportunities to accelerate this non-linear optimization problem to facilitate the migration of CS into the clinical environment. Using 3D steady-state free precession MRI images from 5 patients, we compared the speed and output quality of CS reconstruction using central processing unit (CPU), CPU with OpenMP parallelization, and graphics processing unit (GPU) platforms. Mean reconstruction time was 13.1 $\pm$ 3.8 minutes for the CPU, 11.6 $\pm$ 3.6 minutes for the CPU with OpenMP parallelization, and 2.5 $\pm$ 0.3 minutes for the CPU with OpenMP plus GPU. GPU and CPU reconstructed image quality as assessed by image subtraction were comparable. Additional developments needed for implementation of rapid CS image reconstruction in the clinical environment are discussed.**

I. Introduction

Cardiac magnetic resonance imaging (MRI) has become an essential part of procedural planning and monitoring of children and adults with congenital heart disease [1][2]. An important MRI method for acquiring high-resolution anatomic datasets of the entire thorax is the electrocardiogram and respiratory-gated three-dimensional steady-state free precession (3D-SSFP) sequence [3][4]. A notable limitation of this sequence is its relatively long imaging time, typically 10-15 minutes for 1.2 mm^3 isotropic resolution [5]. A patient's heart rate, breathing pattern, and position may change during such a long scan time, potentially leading to non-diagnostic image quality. Acceleration techniques such as compressed sensing (CS) may be used to reduce the imaging time and minimize the negative effects of variations in heart rate, breathing pattern, and gross motion of patients on image quality [6]. Although CS significantly reduces the scan time, it prolongs the image reconstruction time, since it requires execution of a computationally intensive non-linear optimization algorithm that iteratively estimates the whole image from undersampled data. Therefore, it is important to accelerate the CS reconstruction process so that CS can be used in the clinical setting.

Different approaches have been reported for accelerating CS reconstruction on many types of 1- or 2-dimensional data including Barzilai-Borwein and Split Bergman formulations on a single GPU [7][8] and multiple GPUs [9]. GPU implementation was also used to accelerate CS reconstruction on 2D cine MRI datasets [10]. Although these efforts were successful in speeding up CS reconstruction, they are only suitable for 1D or 2D datasets and lack a general and expandable structure to be suitable for 3D CS reconstruction.

Different methods have also been proposed to accelerate CS reconstructions for 3D MRI and computed tomography datasets using GPU and Intel's many integrated core architectures [11][12][13][14]. These acceleration techniques, however, were not applied to 3D datasets from patients. Thus, the aim of this study was to accelerate the ℓ1-ESPIRiT CS reconstruction algorithm [15] for 3D MRI patient datasets using *GPUs*, and compared the processing time and image quality with CS reconstructions using *CPUs*.

The rest of the paper is organized as follows: Section II provides brief information about CS image reconstruction and describes our experimental setup. Following that, Section III summarizes our results. Section IV analyzes our experiments and results and touches on future work. Finally, Section V concludes the paper.

II. Methods

A. Compressed Sensing (CS) Image Reconstruction

CS reconstruction is based on an iterative non-linear optimization algorithm which aims to estimate and reconstruct images from very few randomly sampled image data. This algorithm is based on 2 principles: a) the samples should be randomly acquired, and b) for image reconstruction, the images should be sparse in a certain domain. This algorithm then reconstructs the images by iteratively optimizing a non-linear cost function ensuring that the output images are consistent with the acquired samples and sparse in a given domain after each iteration.

The ℓ1-ESPIRiT CS reconstruction algorithm that we utilized in this study has 4 main stages. First, the undersampled data is modulated by a complex exponential in the frequency domain (FFTMOD). Then, the data from receiver coils (up to 25 coils for our datasets) are projected onto 8 orthogonal channels in the coil compression stage (CC). In the ESPIRiT calibration stage (ECALIB), the coil sensitivities of the 8 compressed channels are estimated. Finally, in the parallel imaging compressed sensing reconstruction stage (PICS), the CS reconstruction is performed.

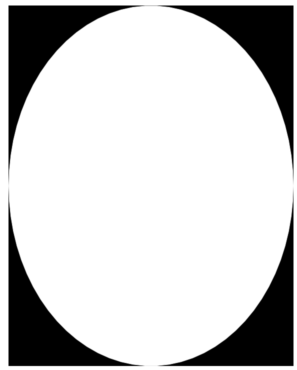
To implement ℓ1-ESPIRiT CS reconstruction on the CPU and GPU, we used the Berkeley Advanced Reconstruction Toolbox (BART) [16]. BART is an open-source framework for computational MRI. It has a programming library which includes common operations on multi-dimensional arrays, such as Fourier and wavelet transforms. Furthermore, BART has generic implementations of iterative optimization algorithms and a toolbox of command line programs which provide direct access to basic operations and efficient implementations of many calibration and reconstruction algorithms for CS.

The CPU node that we used had dual Intel E5 2650 CPUs @ 2.00 GHz and 128 GByte of RAM and the GPU node had NVIDIA Tesla K20m GPU with 2496 computing cores and 4.8 GB of global memory with the processor and RAM configuration similar to the CPU node. For CS reconstruction of the 5 datasets on the GPU, a mean of 591 $\pm$ 101 MB of GPU memory was utilized.

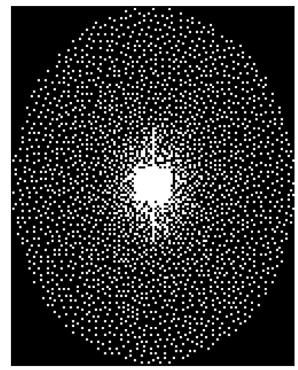
B. Experimental Setup

To compare ℓ1-ESPIRiT CS reconstruction on a CPU and GPU, 5 patients (3 females, age 22.6 $\pm$ 11.1 years) with congenital heart disease referred for cardiac MRI exams were recruited. For each patient, a 3D-SSFP sequence was acquired with the following parameters: field of view about $386\times230\times120$ mm^3, voxel size 1.5 mm^3 reconstructed to 0.75 mm^3, flip angle 90°, echo time 2.4 ms, repetition time 4.7 ms, and bandwidth 542 Hz. The datasets acquired from patients had slightly different dimensions because the patients had different thorax sizes. The 3D-SSFP data were randomly undersampled using a variable density Poisson disc sampling pattern for about a 6-fold reduction of imaging time (Figure 1). After completing the scan, the data was reconstructed using the ℓ1-ESPIRiT CS algorithm on 1) a CPU without any parallelization, 2) a CPU with 24 thread OpenMP parallelization, and 3) a CPU with OpenMP + a GPU, all with 100 iterations. For the CPU with OpenMP + GPU reconstructions, only the PICS stage was accelerated on a GPU; the FFTMOD, CC, and ECALIB stages were performed on a CPU with OpenMP parallelization.

The execution times of ℓ1-ESPIRiT CS reconstruction for CPU, CPU with OpenMP parallelization, and CPU with OpenMP + GPU implementations were evaluated by coarse grain profiling of the 4 main stages in the algorithm (i.e., FFTMOD, CC, ECALIB, and PICS). To ensure that the results were statistically robust, the reconstructions were repeated 10 consecutive times for each dataset in all 3 implementations, and the mean execution time was reported. Then, the bottlenecks in the execution times were determined using the profiling results. Image quality using the 3 different reconstructions was quantitatively compared by calculating the mean difference error between the reconstructed images. When image quality is similar, the mean difference error should be close to zero. A two-tailed paired Student's t-test was used to compare the execution times and the mean difference error. A p-value$\leq$0.05 was considered statistically significant.



a) Fully sampled data



b) Undersampled data

Figure 1: (a) Pattern of fully sampled dataset compared with (b) pattern of variable density Poisson disc undersampled data to achieve about 6 fold reduction in 3D-SSFP acquisition time.

The Boston Children's Hospital Committee on Clinical Investigation approved this study, and written informed consent was obtained from the patients.

III. Results

The acquisitions and reconstructions were successfully completed on all 5 patients. The acquisition time for the 3D-SSFP datasets with about 6$\times$ undersampling had a mean of 3.4 $\pm$ 1.0 minutes.

Table I and Figures 2, 3, and 4 show the CS reconstruction times for CPU, CPU with OpenMP, and CPU with OpenMP + GPU, respectively. The mean processing time to reconstruct the five 3D patient datasets was 13.08 $\pm$ 3.82 minutes on the CPU, 11.56 $\pm$ 3.59 minutes on the CPU with OpenMP, and 2.52 $\pm$ 0.30 minutes on the CPU with OpenMP + GPU. The pairwise differences between the execution times on CPU, CPU with OpenMP, and CPU with OpenMP + GPU for all datasets were significant as all p-value$<$0.01.

Figure 5 shows 3D-SSFP images acquired from a patient before and after image reconstruction using a CPU and a GPU, and the difference between the images reconstructed on CPU and GPU. For all 5 patient datasets, the mean difference between the CPU and GPU reconstructed images was 2.2e-06 and the mean maximum error was 6.9e-04.

Table I: Execution times of PICS on CPU and GPU as well as total execution times of $\ell 1$-ESPIRiT CS algorithm on CPU, CPU with OpenMP parallelization, and CPU with OpenMP parallelization plus GPU for PICS for 3D datasets in 5 patients.

		PICS execution time (minutes)		Total execution time (minutes)		
	Data Size	CPU	GPU	CPU	CPU with OpenMP	CPU with OpenMP + GPU
Patient 1	256×128×113	6.36	0.98	7.83	6.50	2.11
Patient 2	256×145×96	8.97	0.97	10.52	9.46	2.30
Patient 3	256×141×128	12.19	1.33	13.98	12.30	2.66
Patient 4	256×170×121	14.14	1.50	16.00	14.28	2.76
Patient 5	256×157×124	15.17	1.47	16.96	15.28	2.81

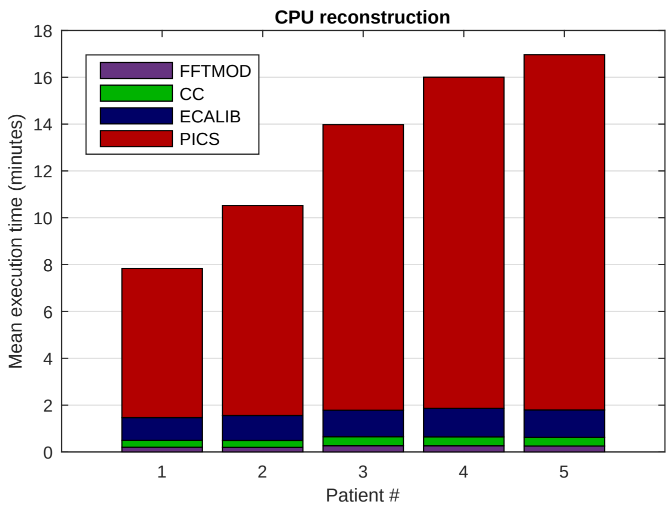


Figure 2: Execution times of the 4 main stages in the $\ell 1$-ESPIRiT CS reconstruction algorithm on the CPU for 3D datasets in 5 patients (ordered according to increasing data size).

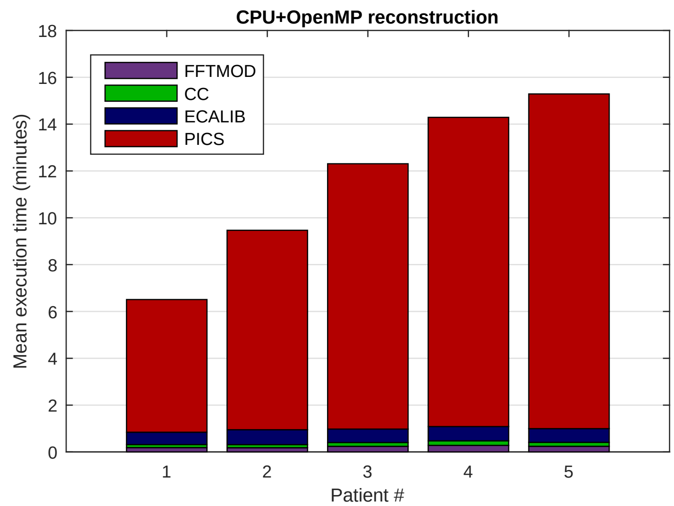


Figure 3: Execution times of the 4 main stages in the $\ell 1$-ESPIRiT CS reconstruction algorithm on the CPU with OpenMP parallelization for 3D datasets in 5 patients (ordered according to increasing data size).

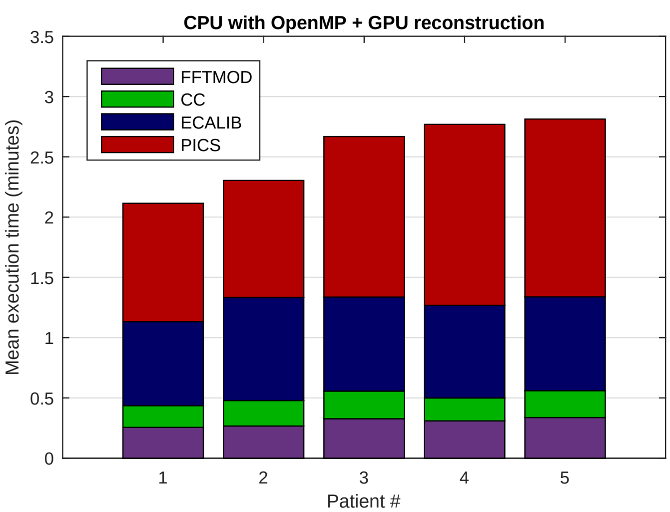
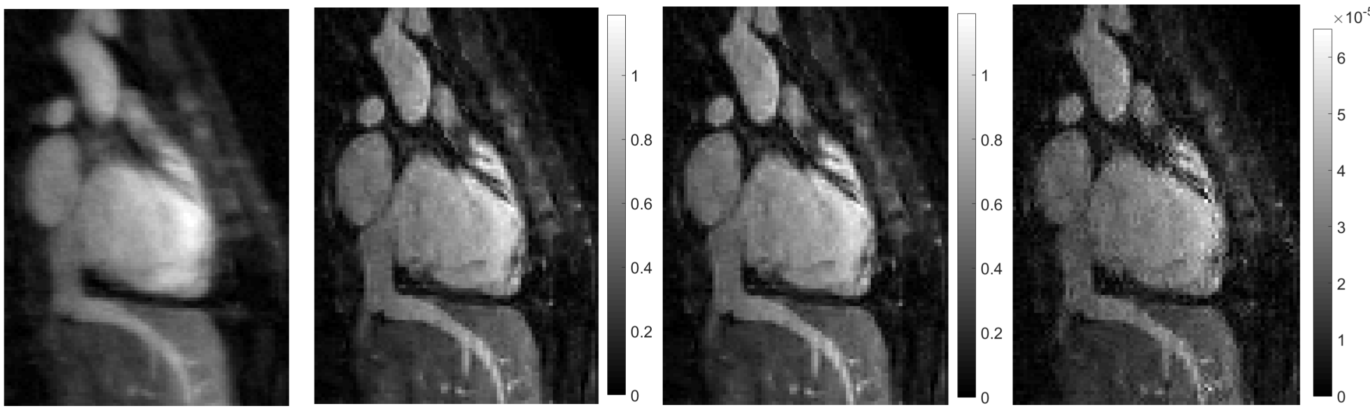


Figure 4: Execution times of the 4 main stages in the $\ell 1$-ESPIRiT CS reconstruction algorithm on the CPU with OpenMP + GPU for 3D datasets in 5 patients (ordered according to increasing data size). Only PICS was implemented on GPU while the other 3 stages were performed on CPU with OpenMP parallelization.

IV. Discussion

Using MRI 3D-SSFP datasets from patients with congenital heart disease, we evaluated the execution time of the $\ell 1$-ESPIRiT CS reconstruction algorithm using a CPU, a CPU with OpenMP parallelization, and a CPU with OpenMP parallelization + a GPU. The CPU with OpenMP decreased the execution time of $\ell 1$-ESPIRiT CS reconstruction algorithm on average 11% compared with CPU without OpenMP. The CPU with OpenMP + GPU reduced the execution time of the CS algorithm on average 80% compared to the CPU without OpenMP and did not compromise the image quality. Implementing only PICS stage on a GPU greatly sped up the CS reconstruction about 5.5× compared to the CPU with OpenMP.

The total execution time of CS reconstruction on the CPU, CPU with OpenMP, and CPU with OpenMP + GPU varied for the datasets with different sizes. However, this variation was smaller on the CPU with OpenMP + GPU compared to the



a) Raw image b) CS reconstructed image on CPU c) CS reconstructed image on GPU d) Difference image (b-c)

Figure 5: An example of 3D-SSFP images in sagittal view acquired from a patient with congenital heart disease: (a) raw image before CS reconstruction, (b) ℓ1-ESPIRiT CS reconstructed image on CPU, (c) ℓ1-ESPIRiT CS reconstructed image on GPU, and (d) the absolute difference between the reconstructed images on CPU and GPU. The mean difference error between the CPU and GPU reconstructed images was 1.37e-05 with the maximum error of 6.49e-05.

CPU implementation (0.3 minutes vs. 3.6 minutes). This could partially stem from the fact that the GPU computation limits are not yet reached and the speedup is only bounded by the data transfer bandwidth and latency to the GPU. After accelerating the PICS stage using a GPU, the execution time of PICS became comparable to the ECALIB processing time. Hence, the calibration stage (ECALIB) could be the next component to investigate for the possibilities of parallelism using GPUs.

The execution times of the FFTMOD, CC, and ECALIB stages for the CPU implementation were longer than their execution times for the CPU with OpenMP and CPU with OpenMP + GPU implementations, because of the OpenMP parallelization that was used in these stages for the CPU with OpenMP and CPU with OpenMP + GPU implementations.

The mean difference between the images reconstructed on CPU and GPU were on the order of 1e-05, and the images were very similar. This small mean difference is potentially due to the fact that different order of operations could lead to different results in floating-point arithmetic. In our study, the PICS stage was implemented using sequential coding on the CPU and parallel coding on the GPU, which yielded to different order of operations and therefore unequal results [17].

Limitations and Future Work

The number of subjects in our study was small. We are planning to apply the CS MRI algorithm on more patients in clinical settings. Currently, only the PICS stage was implemented on a GPU while other stages were executed on a CPU with OpenMP parallelization. These stages can be also implemented on GPUs for further reduction of CS reconstruction time. We also aim to enhance the library components that are used in BART. In addition to custom designed library components, BART uses third party implementations of linear algebra package (LAPACK) and GNU scientific library for linear algebraic and matrix operations, such as Cholesky decomposition, and CUDA library for GPU implementation of fast Fourier transform. We plan to upgrade these libraries to latest versions or replace them with more optimized ones to improve reconstruction performance. This may require changes to the structure of BART to support the new versions of these libraries. A single GPU node was used in our study for accelerating the ℓ1-ESPIRiT reconstruction algorithm. A multi-GPU implementation using message passing interface and CUDA can be used for further improvement in the reconstruction time for 3D or higher dimensional datasets.

V. Conclusions

To the best of our knowledge, this is the first study which evaluates ℓ1-ESPIRiT CS reconstruction algorithm on a GPU for 3D cardiac MRI datasets with different sizes acquired from patients. We show that utilization of a GPU reduces the CS image reconstruction time to less than 3 minutes without compromising image quality. We show that by using the CS acceleration technique in MRI to shorten the imaging time and parallel programming to reduce the corresponding CS image reconstruction time, the total MRI processing time can be significantly minimized. This advance should facilitate the use of CS MRI in clinical settings. Future work will explore further acceleration of CS reconstruction with a goal of less than one minute processing time per 3D cardiac MRI dataset using heterogeneous computing platforms.

References

[1] P. Kilner *et al.*, “Recommendations for cardiovascular magnetic resonance in adults with congenital heart disease from the respective working groups of the European Society of Cardiology,” *European Heart Journal*, vol. 31, no. 7, pp. 794–805, 2010.

[2] R. C. Hendel *et al.*, “ACCF/ACR/SCCT/SCMR/ASNC/NASCI/SCAI/SIR 2006 appropriateness criteria for cardiac computed tomography and cardiac magnetic resonance imaging,” *Journal of the American College*

of Cardiology, vol. 48, no. 7, pp. 1475–1497, 2006.

[3] T. S. Sørensen *et al.*, "Virtual cardiotomy based on 3-D MRI for preoperative planning in congenital heart disease," *Pediatric Radiology*, vol. 38, no. 12, pp. 1314–1322, 2008.

[4] ——, "Operator-independent isotropic three-dimensional magnetic resonance imaging for morphology in congenital heart disease: A validation study," *Circulation - American Heart Association*, vol. 110, no. 2, pp. 163–169, 2004.

[5] M. Fenchel *et al.*, "Three-dimensional morphological magnetic resonance imaging in infants and children with congenital heart disease," *Pediatric Radiology*, vol. 36, no. 12, pp. 1265–1272, 2006.

[6] M. H. Moghari *et al.*, "Three-dimensional heart locator and compressed sensing for whole-heart MR angiography," *Magnetic Resonance in Medicine*, 2015 (in press).

[7] J. C. Park *et al.*, "Fast compressed sensing-based CBCT reconstruction using Barzilai-Borwein formulation for application to on-line IGRT," *Medical Physics*, vol. 39, no. 3, pp. 1207–1217, 2012.

[8] D. S. Smith *et al.*, "Real-time compressive sensing MRI reconstruction using GPU computing and Split Bregman methods," *International Journal of Biomedical Imaging*, vol. 2012, no. 864827, pp. 1–6, 2012.

[9] T. Quan *et al.*, "Multi-GPU reconstruction of dynamic compressed sensing MRI," in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, vol. 9351, 2015, pp. 484–492.

[10] M. S. Hansen *et al.*, "Cartesian SENSE and k-t SENSE reconstruction using commodity graphics hardware," *Magnetic Resonance in Medicine*, vol. 59, no. 3, pp. 463–468, 2008.

[11] B. E. Nett *et al.*, "GPU implementation of prior image constrained compressed sensing (PICCS)," *Proc. SPIE*, vol. 7622, p. 762239, 2010.

[12] D. Kim *et al.*, "High-performance 3D compressive sensing MRI reconstruction," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, 2010, pp. 3321–3324.

[13] S. S. Stone *et al.*, "Accelerating advanced MRI reconstructions on GPUs," *Journal of Parallel and Distributed Computing*, vol. 68, no. 10, pp. 1307–1318, 2008.

[14] S. Nam *et al.*, "Compressed sensing reconstruction for whole-heart imaging with 3D radial trajectories: A GPU implementation," *Magnetic Resonance in Medicine*, vol. 69, no. 1, pp. 91–102, 2013.

[15] M. Uecker *et al.*, "ESPIRiT an eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA," *Magnetic Resonance in Medicine*, vol. 71, no. 3, pp. 990–1001, 2014.

[16] ——, "Berkeley advanced reconstruction toolbox," *ISMRM*, vol. 23, p. 2486, 2015.

[17] Y. Gu *et al.*, "Behavioral non-portability in scientific numeric computing," in *Euro-Par 2015: Parallel Processing*. Springer, 2015, pp. 558–569.