Customized CPU Accelerated CS-based MRI Reconstruction Platform

Kyunghyun Sung[1,2], Di Wu[3], Fei Han[1,2], Ziwu Zhou[1,2], Peng Hu[1,2], Holden Wu[1,2], Alex Bui[1,2], and Jason Cong[3]
[1]*Radiological Sciences, University of California, Los Angeles, Los Angeles, CA, United States,* [2]*Bioengineering, University of California, Los Angeles, Los Angeles, CA, United States,* [3]*Computer Science, University of California, Los Angeles, Los Angeles, CA, United States*

Purpose: Compressed sensing (CS) senses less and computes more [1,2]. By exploiting known structures of otherwise unknown solutions, it uses fewer measurements than traditional methods, allowing for shorter imaging time in MRI; however, it also requires computation to compensate for the reduced measurements, including longer reconstruction runtimes. CS has successfully shown its promise in MRI, but new algorithmic and acceleration approaches are needed to translate CS-based MRI methods into clinical practice. In this work, we propose to develop our new CS algorithm into a CS-based MRI reconstruction platform using customized CPU accelerated implementation.

Compressed Sensing Algorithm: CLEAR (Calibrationless Locally low-rank EncourAging Reconstruction) [3] promotes a low-rank structure in the small image patches and allows reconstructing a multi-coil image x (size: $N_x \times N_y \times N_c$) by solving the following convex optimization without needs of separate or auto-calibration scans:

$$\min_x \; |y - \Phi x|_2^2 + \lambda \sum_{(i,j)\in P} \left| R_{i,j} \cdot x \right|_*$$

where Φ is the undersampled Fourier transform, y is acquired multi-coil k-space data, P contains the central pixel indices of image patches covering all the pixels, and $R_{i,j}$ extracts the small image patch ($B_x \times B_y \times N_c$; $B_x << N_x$ and $B_y << N_y$) and create a matrix ($B_x B_y \times N_c$) at the pixel index (i j). We have extended the CLEAR method by randomly adjusting P to minimize the patch-related artifacts, and have used iterative soft-thresholding [4] to minimize the computational cost and the dependency on the regularization parameter λ.

CPU Accelerated Implementation: C/C++ implementation was performed with Intel Math Kernel Library (MKL) to accelerate the CLEAR algorithm using multi-core CPU. The input data was assumed to be 4D ($N_x \times N_y \times N_z \times N_c$), and the 4D data set was divided into 3D ($N_y \times N_z \times N_c$) for each x location. We have expolored three levels of parallelism: 1) all 3D data for each x location are independent 2) the singular value decomposition (SVD) is independent for each image patch ($R_{i,j}$ x), and 3) parallelism exists across image pixels in computation kernels such as SVD and fast Fourier transform (FFT). Among these levels, we choose to parallelize the reconstruction of 3D data for each x location using OpenMP with the following advantages:

- The core computations (SVD and FFT) are relatively small and can be fit in CPU last two level caches, minimizing the communication overhead across CPU cores and sockets.
- Parallelizing the outermost loop (3D data for each x location) minimizes the thread-scheduling overhead of OpenMP.
- 3D data may have different convergence rates, and OpenMP dynamically balances the workload for each thread and minimizes synchronization overhead.

Custom Image Reconstruction Platform: Figure 1 shows the schematic outline of the proposed reconstruction framework, and the actual installation of the reconstruction computer (CPU: 4core/4GHz, Memory: 32GB) to a research-only intraoperative MRI scanner (3.0T Siemens Trio system). The total cost of the reconstruction computer is less than $1,000. Two TCP/IP communication modules were inserted to the vendor provided reconstruction pipeline to bypass the FFT module (FFT, partial Fourier, GRAPPA/SENSE, and coil combination). Other reconstruction modules including k-space filters and post-reconstruction image filters are maintained to minimize the complexity of the custom-built non-linear image reconstruction algorithm.

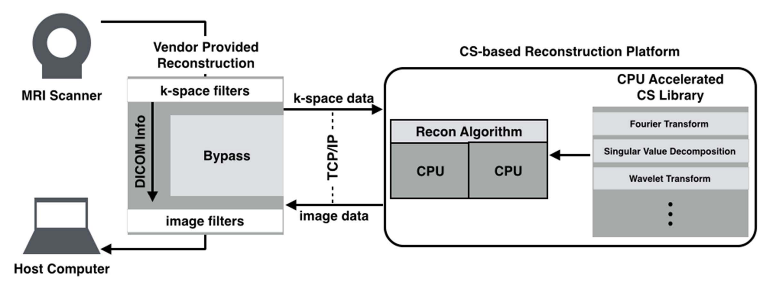

Fig 1: CS-based reconstruction platform using customized CPU accelerated implementation.

Evaluation and Results: We evaluated the proposed CS-based reconstruction platform by prospectively acquiring undersampled k-space data (7x acceleration) and assessing the actual reconstruction runtime (from end of the scan to end of image transmission to the console).

All reconstructions were successfully completed and not compromised by the CPU-acceleration. The reconstructed images were available at the console 1-3 minutes after scan finished. Table 1 summarizes the total reconstruction runtime with different configurations. Our CPU accelerated implementation was able to achieve 120x speed-up compared with Matlab baseline, and combining CPU and GPU can further reduce the reconstruction runtime (projected values).

Data Matrix Size (Nx × Ny × Nz × Nc)	CLEAR (MATLAB)	CLEAR (CPU)	CLEAR (CPU/GPU; Projected)
512×300×140×12	5 hour	2.5 min	38 sec
256×256×60×12	1 hour 38 min	50 sec	14 sec

Table 1: *Comparison of reconstruction runtime for MRI data*

Conclusion: The proposed reconstruction framework using customized CPU accelerated implementation significantly increased the computational power at a very low cost and enables in-line CS-MRI reconstruction within acceptable time.

References: [1] Donoho, IEEE TIT, 2006. [2] Candes et al., Inverse Problems, 2007. [3] Trzasko, et al., IEEE 2011. [4] Daubechies et al., Comm Pure Appl. Math 2004.