Accelerating Compressed-Sensing-Based DCE-MR Image Reconstruction with GPU

Jiangsheng Yu[1], Yiqun Xue[2], and Hee Kwon Song[2]
[1]Toshiba Medical Research Institute USA, Cleveland, Ohio, United States, [2]University of Pennsylvania, Philadelphia, Pennsylvania, United States

Introduction: Dynamic contrast-enhanced (DCE)-MRI is a powerful tool to evaluate tumor vascular response and for the assessment of antiangiogenic and antivascular therapeutics. High temporal resolution is needed for accurate determination of tumor enhancement kinetics. Radial acquisition with golden-angle view angle increment followed by temporal filtering (KWIC) was previously proposed for DCE-MRI to achieve high temporal resolution without losing image quality. Alternatively, temporally constrained reconstruction based on compressed sensing (CS) has recently been developed for dynamic MR imaging to obtain both high temporal and spatial resolutions, in which a high undersampling factor is achieved by utilizing the temporal sparsity between the neighboring time frames[1]. The accuracy of the CS reconstruction was investigated and demonstrated that the CS method potentially yields more accurate and precise perfusion measures over other undersampling reconstruction methods such as KWIC [2]. However, the intensive computation overhead in CS reconstruction has limited the application for clinical data processing where large data sets are generated from multi-slice and multi-channel acquisitions. In this work a parallelized implementation of CS reconstruction for DCE-MRI using graphics processing unit (GPU) is presented and compared with that of the sequential C++ implementation.

Methods:
CS-based DCE-MR image reconstruction aims to minimize the following cost function

$$\arg\min_{m} \left\| F_u m - y \right\|_2^{\,2} + \lambda \sum_{i=1}^{N} \left\| D_t(m_i) \right\|_1 \qquad (1)$$

where m is the reconstructed image, N is the total number of time frames, F_u denotes the undersampled backward gridding operator, y is the measured k-space data, and the coefficient λ weighs the temporal sparsity of the enhancement relative to the data fidelity term. D_t is the image intensity difference between adjacent time frames. $\| \|_1$ and $\| \|_2$ denote L1 and L2 norms. The time-frame -difference operator in the sparsity term utilizes the fact the subtraction of the neighboring time frames removes the background signal and thus reflects the signal changes due to contrast agent injection.
Conjugate gradient algorithm with backtracking line search is used to iteratively reconstruct the DCE-MRI dynamic series. A profiling of the sequential C++ implementation shows that the gridding transforms back and forth between radial and Cartesian k-spaces (and thus called the forward and backward griddings) are the most time consuming part of the CS reconstruction. As shown in Figure 1(a), depending on the distance and the convolution kernel, the radial k-space point distributes to its neighboring Cartesian points in the forward gridding, and vice versa in the backward gridding. In the GPU implementation, both forward and backward griddings are radial-point driven, i.e., a thread is assigned to each radial point operation. A synchronization problem arises in the forward gridding when two threads try to access the same memory indexed by the Cartesian point (Figure 1(b)). Although it is possible to perform atomic operation to sequentialize the memory access, this process is time-consuming (except some high-end GPUs [3]) and offsets the GPU parallel processing advantages. We utilize a simple yet effective method previously reported for parallel gridding, in which the radial points of each view are divided into multiple groups whose Cartesian neighboring points are non-overlapping [4].

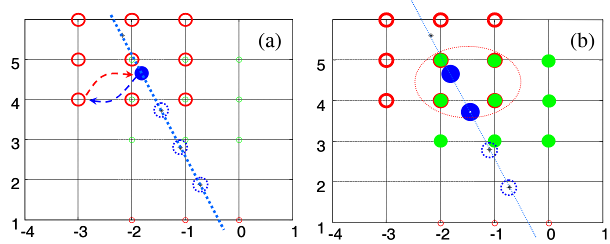

Fig. 1 (a) Forward (blue arrow) and backward (red arrow) gridding between radial and Cartesian k-spaces. The radial sampling point contributes to the neighboring Cartesian points in forward gridding, and vice versa in backward gridding (b) Synchronization problem in forward gridding when each radial point is assigned a thread in GPU parallel computation.

Results and Discussion:
Figure 2 shows the DCE-MRI images reconstructed with CS algorithm for a 3D clinical data set acquired using 3D hybrid radial acquisition with the following parameters: TR=3.38ms, TE=1.6ms, flip angle=25°, FOV=30x30cm^2, 26 slices, 5 channels, 6144 radial views with golden-angle increment, 192 readout points. The CS reconstruction uses 25 views per time frame, resulting a temporal resolution of ~2.2s. To reconstruct 100 time frames of the dynamic series with a 5x5 gridding kernel on a laptop computer with a Intel Core i7-2640M (@2.80GHz 2.80 GHz) CPU and a GeForce GT 640M 96 cores (@1.5GHz) GPU, it took ~2.0hrs and ~30hrs for the parallelized GPU program and C++ sequential program, respectively, resulting an acceleration factor of ~15. A profiling of the GPU kernel functions shows that further parallelization and optimization of conjugate searching algorithm need to be performed to achieve a higher acceleration factor.
The sparsity term in Eq.[1] includes the signal differences between all the neighboring time frames. The approach prevents online image reconstruction as data processing has to wait until the end of data acquisition. A piece-wise based CS reconstruction is under investigation to compare the image quality and total reconstruction time.

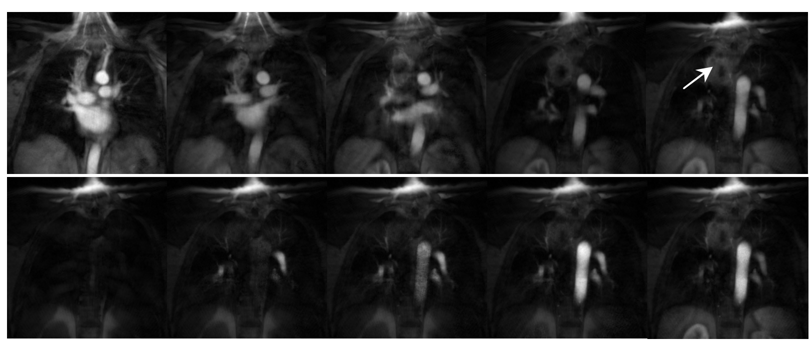

Fig. 2 GPU accelerated CS image reconstruction for a clinical DCE-MRI data set. Five representative slices are shown in the first row. The second row demonstrates the dynamic signal changes at time points [0, 2, 7, 11, 20] seconds after the injection of the contrast agent for the fifth slice. The tumor is marked with the arrow.

Conclusion: Current work presents a parallelized GPU implementation to accelerate the CS-based image reconstruction in radial DCE-MRI. The forward and backward gridding operations, which are the most-time consuming part of the conjugate gradient searching, is addressed with a radial-point driven parallelization approach by assigning a thread for each radial point operation. To avoid the synchronization problem in forward gridding, the radial points of each view are divided into multiple groups whose Cartesian neighboring points are non-overlapping. A comparison with the C++ sequential implementation shows an acceleration factor of ~15 can be achieved on a moderately GPU-powered laptop computer.

Acknowledgments to Dr. Song's grants: American Cancer Society RSG-08-118-01-CCE; NIH P41-RR02305; NIH R01-CA125226; NIH UL1RR024134.

Reference: (1) Adluru et al. Magn Reson Med 2007; 56: 1027-1036. (2) Xue et al. ISMRM 2012:1966. (3) Nam et. al., Magn Reson Med 2012, 5, Mar (online). (4) Yu, et al ISMRM 2012:2553.