Interactive Visualization of 4D Coronary MRI with Prolonged Acquisition Window: GPU-Accelerated Flexible Gridding Reconstruction for Lag-Free Performance

Keigo Kawaji[1], Sebastien Roujol[1], Warren J Manning[1,2], and Reza Nezafat[1]

[1]*Medicine, Beth Israel Deaconess Medical Center and Harvard Medical School, Boston, MA, United States,* [2]*Radiology, Beth Israel Deaconess Medical Center and Harvard Medical School, Boston, MA, United States*

TARGET AUDIENCE: Scientists and clinicians interested in real-time workflow and visualization of non-Cartesian MR data.

INTRODUCTION: Free-breathing 3D coronary MRI sequences are commonly acquired at the quiescent period of the cardiac cycle obtained from a prior breath-held scout CINE scan. However, the derived trigger delay timing may be inaccurate due to patient's R-R interval fluctuations between these scans, or due to breath-holding induced heart rate variability. We have recently proposed an approach that acquires a larger acquisition window and allows retrospective reconstruction of any desired temporal sub-window with minimal coronary motion [1]. This method uses a 3D stack-of-stars (SOS) sampling scheme with golden-angle-derived interleaving, and requires a dedicated visualization platform to allow interactive selection of any desired reconstruction window by the operator. However, 3D SOS reconstruction needs a computationally expensive gridding operation, which may bottleneck the visualization step by causing lagging performance that hinders the interactive clinical assessment. In this study, we propose and evaluate a graphical user interface (GUI)-based, graphic processing unit (GPU)-accelerated reconstruction and visualization workflow. Our technique performs rapid gridding, and displays data from any operator-selected temporal subset of the time-sorted 4D k-space data in real-time to fit within a clinical workflow for coronary MR assessment.

PROPOSED WORKFLOW: The acquired 3D radial SOS k-space data was processed using an in-house platform developed in MATLAB, and was integrated with GPU processing for rapid non-Cartesian gridding and real-time visualization. The raw k-space data is first indexed into the following coordinate system [k_x, k_y, k_z, t], and passed to the GPU upon operator instruction. The provided GUI (Figure 1) forwards all selected k-space points and coordinates from the selected slice and temporal window to the GPU reconstruction engine. The 2D image is then processed in the following steps: 1) The input parameters (namely the complex values and its coordinates) from MATLAB are mounted onto the GPU memory. 2) The gridding operation is parallelized and performed on a per-point basis, where gridded data from each GPU kernel is accumulated onto the final Cartesian grid using a dedicated GPU atomic addition operation [3,4]. 3) Image reconstruction is performed as a sequence of four sub-steps: a) zero-padding, b) inverse FFT, c) Cropping to the correct FOV, and 4) Roll-off correction. 4) The reconstructed image is mounted back to the CPU memory, and is returned to the MATLAB GUI.

EXPERIMENTAL EVALUATION: All experiments were performed using coronary data acquired using a balanced SSFP sequence with radial SOS sampling, where the rotation angle was customized to account for uniform distribution over any subset within the acquisition window. For the GPU operation in this study, a NVIDIA GeForce GTX 480 hardware was used, where all GPU operations were implemented using the CUDA platform. The performance of each GPU reconstruction substep was evaluated using different input k-space parameters typically acquired in high-resolution coronary MRI, and the measured times were averaged over 100 operations. For comparison, the reconstruction was also performed and evaluated using the NUFFT package [2] provided in MATLAB, which requires an initial preparation step that generates a look-up table.

RESULTS: Fig 1 shows the timing of each of the reconstruction operation from the operator instructions to visualization. The total time for the combined transfer between MATLAB and GPU was 0.2 seconds. Table 1 shows the average computation times at different reconstruction window sizes. The GPU implementation required a fixed 0.2 seconds for MATLAB-GPU interaction, 0.2-0.35 seconds for gridding - for a total of 0.4-0.6 seconds for visualization of the desired 2D image from any desired temporal window and slice. For CPU gridding with NUFFT, an initial preparation step of the appropriate k-space look-up table took 1.5-10.5 seconds before performing the gridding and visualization steps in a single step (1.2-1.8 seconds), which required ~3 times than the GPU implementation.

Significant speed-up in total processing time is observed compared to the NUFFT-based preparation. When the GPU-based gridding parameters were matched to the default input parameters for NUFFT-based gridding (kernel shape parameter W from 5.5 to 6.0 and oversampling factor from 1.25 to 2.0), the average computation time using the GPU increased from 0.53 sec to 0.86 sec; significantly less than the equivalent NUFFT-based gridding step that may take as much as 13 seconds (10.5 for prep + 1.8 for gridding), which is unsuitable for a flexible and retrospective reconstruction workflow for interactive visualization.

DISCUSSIONS AND CONCLUSION: A GUI tool that provides flexible GPU-based offline radial k-space reconstruction is developed, and enables on-the-fly assessment of any desired temporal window to retrospectively select a motion-free 3D volume from the 4D coronary k-space data.

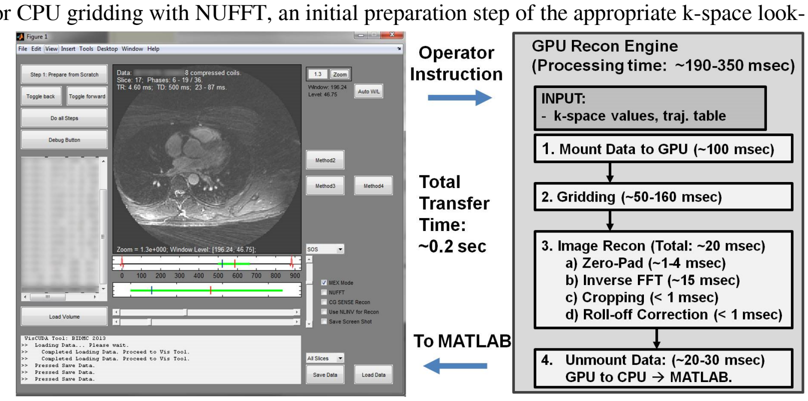

Figure 1. Screenshot of GUI (left side) and workflow of GPU reconstruction (right side). Computation times are reported for each step, requiring a total of 190-350 milliseconds for the GPU operation to perform. An additional 0.2 ms is required for total latency to transfer the input and resulting image data back and forth between the GUI and GPU.

Table 1 Average Processing Times GPU vs CPU Methods

Test Methodology (#Samples × #Spokes × #Coils) #k-space points to grid	GPU-TRANSFER	GPU-GRID	GPU-TOTAL	CPU-PREP	CPU-GRID /TOTAL
	Total Computation Time (seconds)				
1.3 mm resolution Recon					
RecWin Size = 7; (464 × 98 × 8) = 363776 pts	0.18 ± 0.01	0.19 ± 0.02	0.37 ± 0.04	1.5 ± 0.1	1.2 ± 0.1
RecWin Size = 16; (464 × 224 × 8) = 831488 pts	0.18 ± 0.01	0.23 ± 0.02	0.41 ± 0.04	3.5 ± 0.4	1.4 ± 0.1
RecWin Size = 48; (464 × 672 × 8) = 2494464 pts	0.18 ± 0.01	0.35 ± 0.03	0.53 ± 0.04	10.5 ± 1.1	1.8 ± 0.2
Submillimeter resolution Recon					
RecWin Size = 48; (600 × 864 × 8) = 4147200 pts	0.19 ± 0.01	0.78 ± 0.08	0.96 ± 0.10	17.9 ± 1.8	2.8 ± 0.29

ACKNOWLEDGEMENT: The authors acknowledge grant support from Samsung Electronics and NIH R01EB008743-01A2.

REFERENCES: 1) Kawaji et al. Int MRA Workshop 2013. 2) Fessler et al. IEEE Trans Sg Proc. 2003. 3) Sorensen et al. IEEE Trans Med Im 2007. 4) Nam et al. MRM 2013.