Towards real-time availability of 3-D temperature maps created with temporally constrained reconstruction

Nick Todd[1], Jaya Prakash[2], Henrik Odeen[3], Josh de Bever[4], Allison Payne[1], Phaneendra Yalavarthy[2], and Dennis L. Parker[1]
[1]*Radiology/UCAIR, University of Utah, Salt Lake CIty, UT, United States,* [2]*Supercomputer Education and Research Centre, Indian Institute of Science, India,* [3]*Physics, University of Utah, Salt Lake CIty, UT, United States,* [4]*Computer Science, University of Utah, Salt Lake CIty, UT, United States*

INTRODUCTION:

In order to improve MR scan time, many investigators use the approach of sub-sampling the k-space data and generating images with a constrained reconstruction (or compressed sensing) technique. While these techniques are greatly beneficial to dynamic imaging studies due to the improved temporal resolution they offer, they are not yet suitable for real time applications as most require long reconstruction times and perform the reconstruction in a batch fashion that uses all time frames at once. Here we present an approach to overcome these limitations, which we call real-time temporally constrained reconstruction (RT-TCR), and apply the method to undersampled dynamic 3-D MR temperature data of a high intensity focused ultrasound (HIFU) heating procedure.

METHODS:

Original TCR. The original TCR algorithm reconstructs images, *m*, from k-space data, *d*, by iteratively minimizing a cost function[1, 2]:

$$m = \underset{m'}{\arg\min}\left(\left\| WFm' - d \right\|_2^2 + \alpha \sum_i^N \left\| \nabla_t m'_i \right\|_2^2 \right) \qquad [1]$$

where *F* is the Fourier Transform, *W* is a binary function that represents which phase encoding lines have been acquired, *m'* is the image estimate, and α is a spatially varying free parameter. Using an entire 4-D data set of 192x108x30x77 voxels and 100 iterations, the algorithm takes 236 seconds to converge on a 12-core computer (Dual Intel Xeon Processor X5650, 2.66 GHz, 64 GB RAM).

RT-TCR. Several modifications have been made to the original TCR method in order to achieve real-time availability of the reconstructed images. First, the data is reduced along the fully sampled k_x and k_z directions such that the RT-TCR algorithm only operates on the part of the image where the temperature is changing rapidly. A sliding window reconstruction is used for the other areas. Second, the code has been rewritten for both parallel processing and a GPU implementation (NVIDIA Quadro 6000 with 448 cores). Third, when the data for the current time frame *t* is acquired, the reconstruction is performed using only frames [*t* – P, *t*], with P = 12 for these results. This reconstruction gives an initial estimate of the temperature at time *t*, and is also used to update the temperatures for times *t*-1, *t*-2, ... *t*-P/2. In this way, an initial estimate of the current temperature is obtained with no lag time (except for the RT-TCR computation time), and the temperatures for previous time frames are updated successively until they converge, with each update having some lag time.

Testing. A 4-D data set acquired during HIFU heating of an agar phantom was used for testing (3-D segmented EPI sequence, 6X undersampling, 1.5x1.5x3.0mm resolution; 192x108x30 matrix; TR/TE = 28/10 ms; EPI factor = 9; 1.7 sec/image). RT-TCR temperatures (with no lag time and with a 1-time-frame lag) are compared against sliding window and original TCR temperatures.

RESULTS & CONCLUSIONS:

Using a 10x108x10x13 voxel subset of the data, the RT-TCR algorithm creates images in 1 second using the 12-core computer (CPU) and 0.7 seconds using GPU. Figure 1 shows temperature plots from one voxel in the center of the HIFU hotspot. Figure 2 shows temperature maps from the original TCR images at four different times (black arrows in Figure 1) and the corresponding difference maps for sliding window and RT-TCR reconstructions. The RT-TCR temperatures with no lag deviate somewhat from the original TCR temperatures, especially where the 2nd derivative in time is high. Minimal errors remain after first RT-TCR update and almost no errors remain after the 2nd update (not shown). The results are a good step towards realizing real-time availability of TCR images that would provide high spatial and temporal resolution temperature maps with full 3-D coverage over a large volume, and GPU computing holds a promise in this realization.

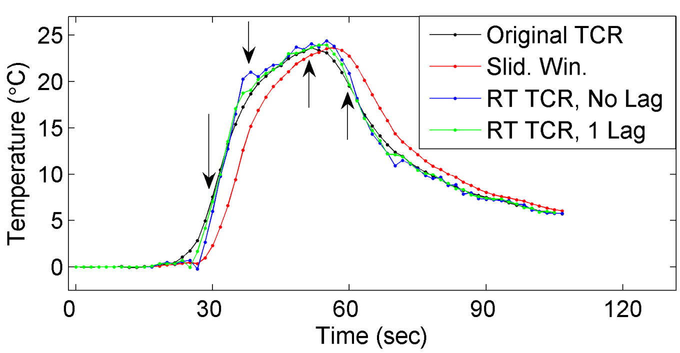

Figure 1. Temperature plots of the voxel in the center of the heated region. Original TCR temperatures are compared to sliding window and RT-TCR temperatures. Sliding window reconstruction shows significant temporal averaging effects. RT-TCR temperatures with no lag show fairly good agreement, albeit with some errors. RT-TCR temperatures with a 1-time-frame lag show very close agreement. Black arrows indicate the four time frames displayed in Figure 2.

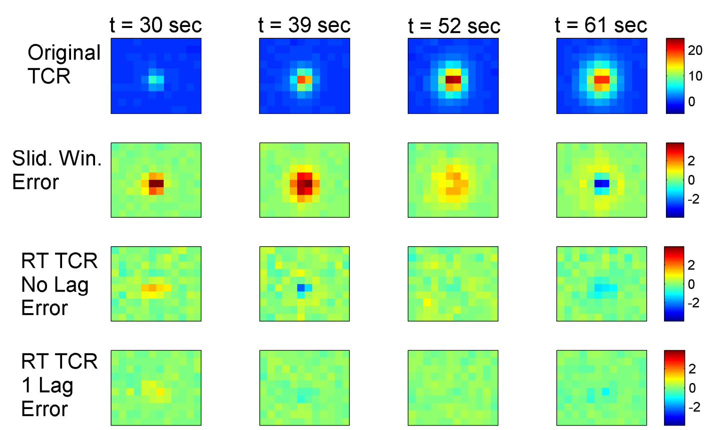

Figure 2. Top Row: Hot spot from original TCR temperatures. 2nd Row: Error for sliding window reconstruction. 3rd Row: Error for real-time TCR reconstruction with no lag. 4th Row: Error for real-time TCR reconstruction with one time frame lag.

REFERENCES: 1. Todd et al. MRM; 62(2),406-19, 2009. 2. Todd et al. *Reconstruction of fully 3-D high spatial and temporal resolution MR temperature maps for retrospective applications.* MRM, In press, 2011.

ACKNOWLEDGEMENTS: The Focused Ultrasound Surgery Foundation, The Margolis Foundation, Siemens Medical Solutions, NIH grants R01 CA87785, and R01 CA134599.