Parallel Computing for Simultaneous Iterative MIMO Tomographic Image Reconstruction by GPUs

Ricardo López (Undergraduate)[1], Colleen Rogers (Undergraduate)[2] ,Yuanwei Jin (Faculty Member)[3] , Enyue Lu(Faculty Member)[2]

[1]University of Puerto Rico, Río Piedras [2] Salisbury University [3] University of Maryland Eastern Shore

Highlights of Contribution

1. Leverage GPU resources to accelerate nonlinear tomographic image reconstruction with Simultaneous Iterative Reconstruction Technique (SIRT), accounting for varying degrees of noise
2. Address higher memory usage of SIRT versus the traditional Algebraic Reconstruction Technique (ART) by developing a scalable implementation of the SIRT algorithm
3. Apply a novel non-uniform weighting to accelerate the convergence of SIRT

Motivation

Ultrasonic Tomography is an imaging process that has many applications in medical imaging, oceanic engineering, seismic engineering, etc. Due to it's highly computational nature, we can accelerate this process by leveraging the massively parallel computational resources of a Graphics Processing Unit (GPU). We consider the Simultaneous Iterative Reconstruction Technique (SIRT) as a way to not only accelerate the image reconstruction process, but also to better handle noise in the initial measurement data, which is a concern in real-world scenarios.

Introduction

Ultrasonic tomographic imaging involves propagating a sound wave through an imaging field and then algorithmically reconstructing an image of the target based on sensor data and the acoustic wave propagation model. The process has been previously accelerated by using a Multiple Input, Multiple Output (MIMO) approach, whereby sensors are divided into groups and each sensor in a group sends a wave through the field simultaneously. This facilitates parallel execution of the image reconstruction, at the cost of image accuracy.

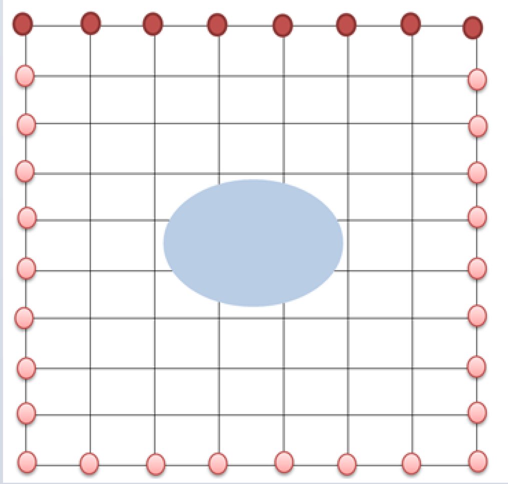


Figure 1: Tomographic Imaging Field

The **goal** of the image reconstruction problem is to reconstruct the image vector f from the measured data, r, and the underlying wave model $R()$ taking into account a certain degree of noise or disturbance, η

$$r = R(f) + \eta$$

The Propagation-Backpropagation (PBP) Method

1. Calculate predicted value based on some initial value
2. Calculate difference between sensor measurement and the previous estimate
3. Based on the difference signal perform the necessary updates

Image Reconstruction Techniques

Algebraic Reconstruction Technique

- A new image is reconstructed at each step of PBP method
- Image update from current sensor group depends on image produced from previous sensor group; work must be done sequentially

$$f^{k+1} = f^k + \omega\delta f^k(S_m) \qquad m = 1,2 \dots M$$

Simultaneous Iterative Reconstruction

- Image is updated when all sensor groups have executed PBP method by taking an average
- Work from each sensor group excitation can be executed in parallel

$$f^{k+1} = f^k + \omega \sum_{m=1}^{M} \delta f^k S_m$$

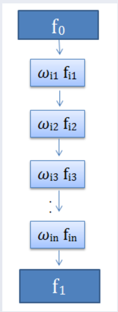


Figure 2: ART

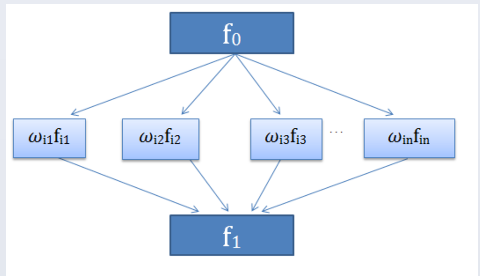


Figure 3: SIRT

Advantages of SIRT over ART

- Higher degree of data parallelism → faster iterations
- Averaging of images →better handling of noise in measurement data

Disadvantages of ART over SIRT

- Higher memory usage = O (# groups)
- More iterations to create equivalent image → slower convergence

Parallelizing SIRT on GPU

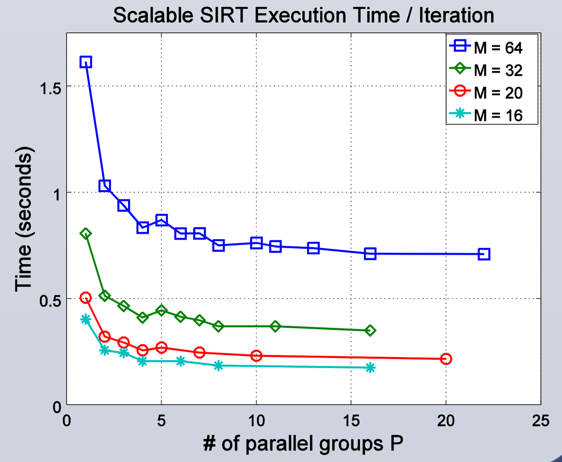
We utilized the NVIDIA Computed Unified Device Architecture (CUDA) programming model, which allowed us to strategically allocate GPU resources and dramatically accelerate the execution of SIRT, compared to CPU implementation .

	Group Size 10	Group Size 32
CPU	49.86984 s	15.741976 s
GPU	0.76253s	0.23898s
Speedup	65.4x	65.9x

Table 1: Average SIRT execution time /iteration

Scalability and Memory Constraints

In the SIRT implementation, each of the M image updates must allocate some space in GPU memory in order to execute simultaneously. For large enough M, this can overwhelm the available GPU memory. We scale the memory usage of the SIRT by executing P image updates in parallel, P ≤ M, and iterating until all M updates have been calculated.

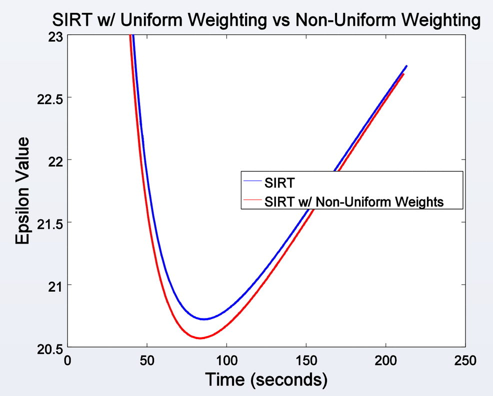


Non- Uniform Weighted SIRT

We can apply non-uniform weights to each of the intermediate images produced by SIRT when we update our current image, depending on the quality of each. We determine the quality by calculating an average over the intermediate images and measuring how close each intermediate image is to the average.

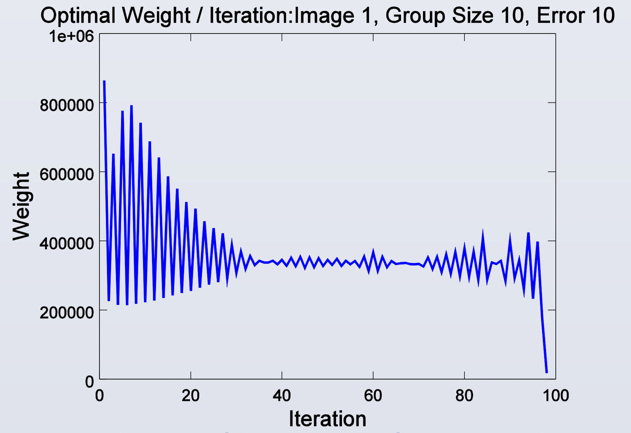
$$d_m = norm(\delta f^k(S_m) - avg(\delta f^k)$$

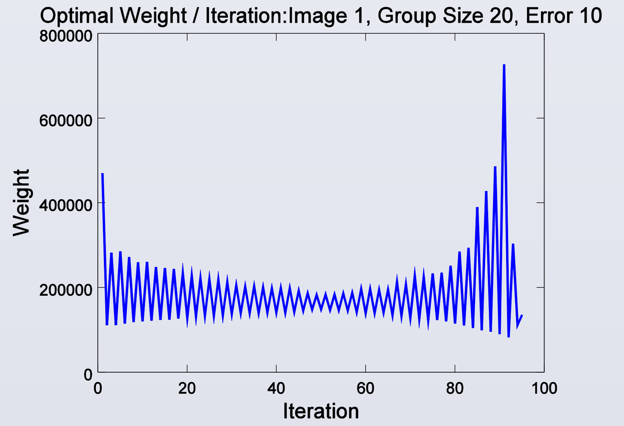
$$f^{k+1} = f^k + \omega \sum_{m=1}^{M} \frac{1}{d_m} \delta f^k(S_m)$$



Iterative Relaxation Factor

At the end of each iteration of SIRT, we can weigh how much the normalized image should contribute to updating the current image. We developed a program that greedily chose the best scaling factor at each iteration. Running this program with different images at different sensor group sizes, we noticed that the optimal scaling factor had an alternating pattern that would converge to some value after a certain number of iterations.





We approximate this pattern by proposing two alternating functions ω_i, with two distinct sets of parameters α_i and β_i:

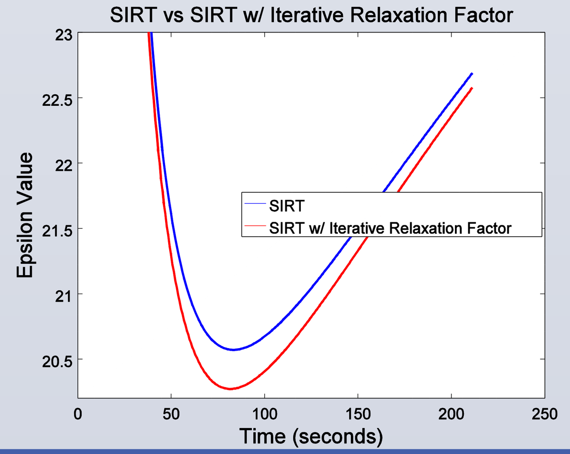
$$\omega_i(k) = \alpha_i + \frac{\beta_i}{k}, \qquad i = k \bmod 2$$

The new update function for SIRT becomes:

$$f^{k+1} = f^k + \omega_i(k) \sum_{m=1}^{M} \frac{1}{d_m} \delta f^k(S_m), \qquad i = k \bmod 2$$

By setting the parameters α_i and β_i for a given image, noise level, and group size

We are able to improve the convergence of the SIRT algorithm



Testing Environment

GPU: Tesla k20c

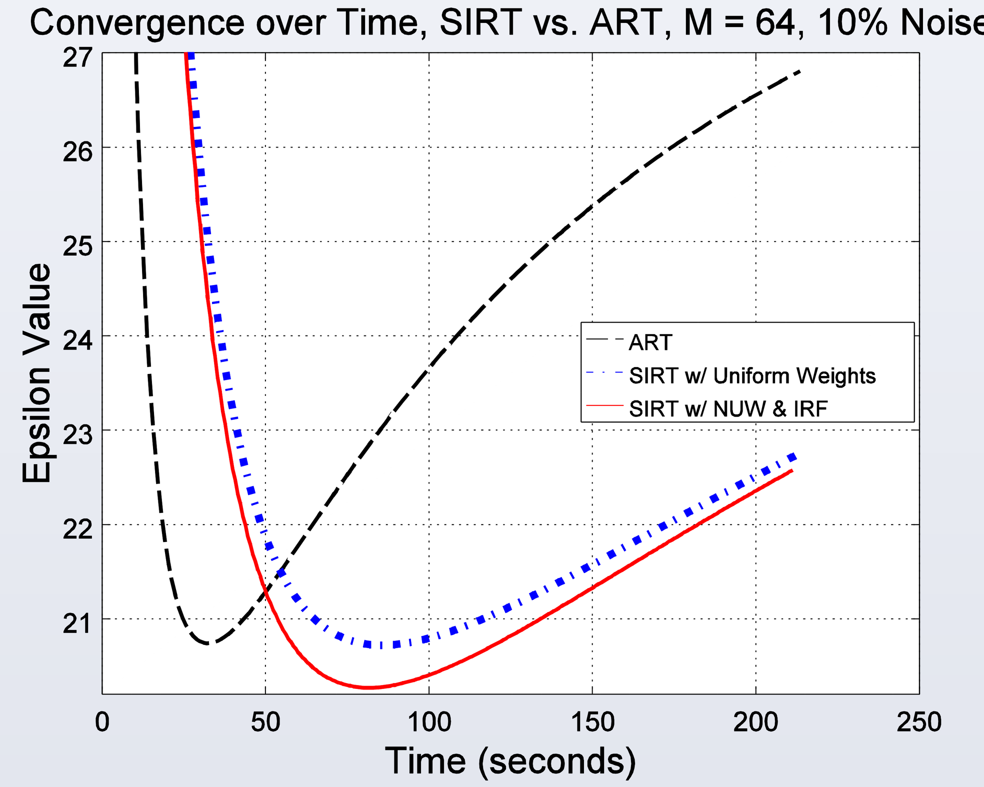
- 2496 CUDA Cores @ 0.71 GHz
- 4.69 GB RAM

CPU: Intel Xeon E5-2660 v2

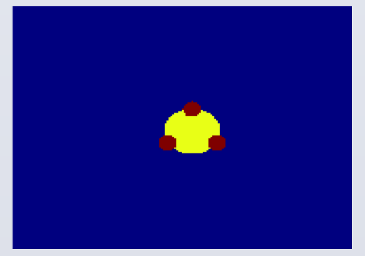
- 8 Cores @ 2.20 GHz
- 94.5 GB RAM

Preliminary Results: SIRT vs ART

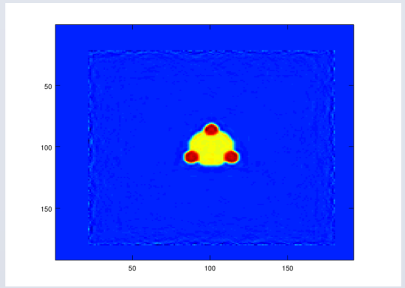
We compare the image reconstruction with SIRT using non-uniform weighting and iterative weighting function vs. ART using different constant weight factors.



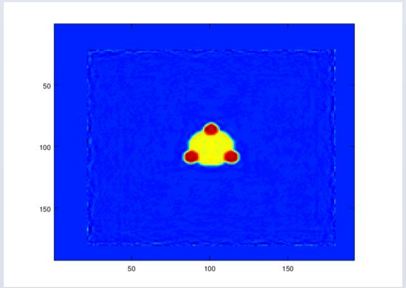
Ground Truth



SIRT



ART



Conclusion and Discussion

We've shown that we're able to improve the SIRT's convergence in the presence of noise by the use of non-uniform weighting schemes. Although minimal, this SIRT implementation shows improved convergence when compared to the traditional ART, albeit more slowly and with greater memory usage. Given a larger problem set, we conjecture that the Non-Uniform Weighted SIRT will show an even larger improvement over the ART.

Acknowledgements

This work was funded by the National Science Foundation under grant no. CCF-1460900 and the Army Research Office under grant no. W911NF-11-1-0160.