Northeastern University

Accelerating 3D CT Reconstruction Using GPUs

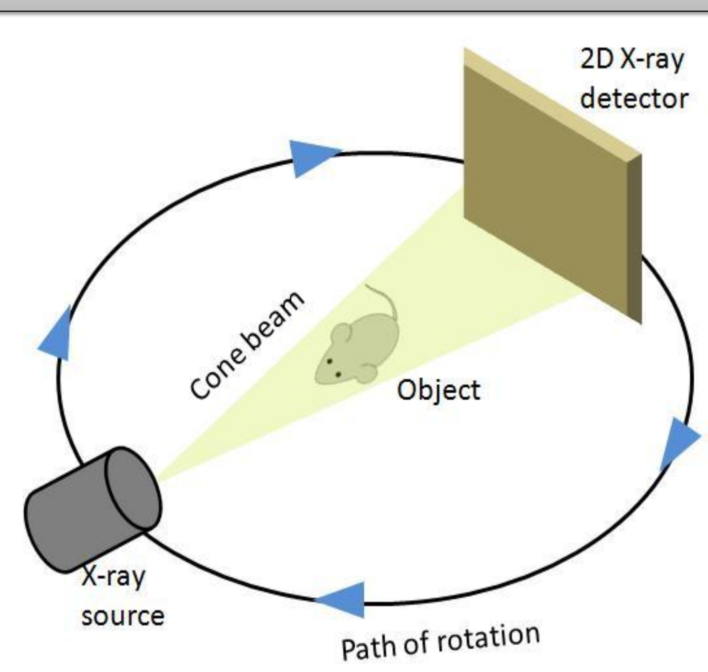
Saoni Mukherjee, Nicholas Moore, James Brock, Miriam Leeser

Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA

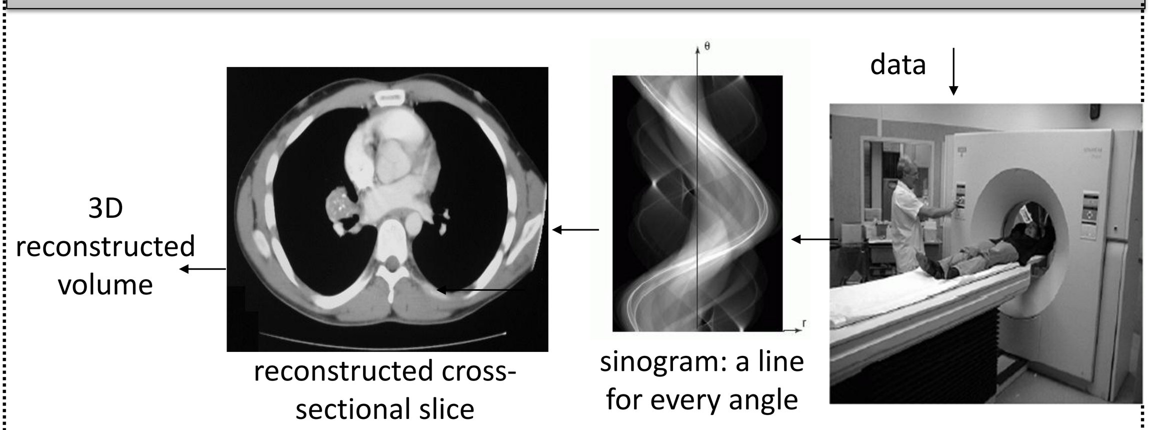
Abstract

Biomedical image reconstruction applications with large datasets can benefit from acceleration. Graphic Processing Units(GPUs) are particularly useful in this context as they can produce high fidelity images rapidly. An image algorithm to reconstruct conebeam computed tomography (CT) using two dimensional projections is implemented using GPUs. The implementation takes slices of the target, weighs the projection data and then filters the weighted data to backproject the data and create the final three dimensional reconstruction. This is implemented on two types of hardware: CPU and a heterogeneous system combining CPU and GPU. The CPU codes written in C, OpenMP and MATLAB are compared with several heterogeneous versions written in CUDA-C and OpenCL. The relative performance is tested and evaluated on a mathematical phantom as well as on mouse data. Speedups of over 40 times using the GPU are seen for phantom data and close to 90 times for the larger mouse datasets over multithreaded C implementation.

CT Scan Procedure

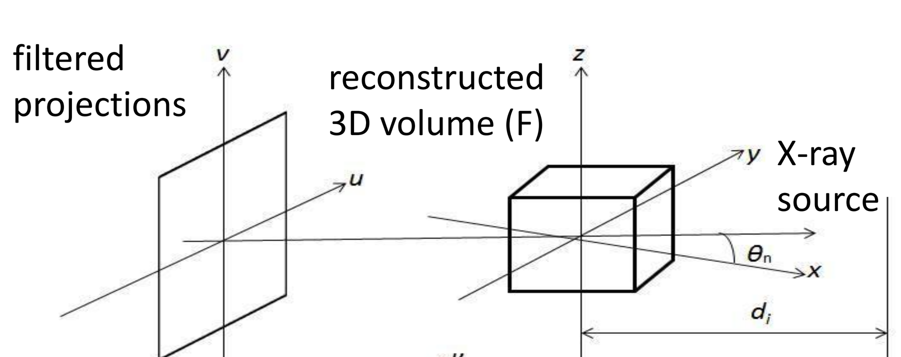


3D CT Reconstruction



Klaus Mueller, Introduction to Medical Imaging, Lecture 6: X-Ray Computed Tomography, Computer Science Department, Stony Brook University

Feldkamp Algorithm



Weighted Projection: Weighted and ramp filtered raw data produce filtered projections $Q_1, Q_2, \ldots, Q_K$, collected at an angle ϑ_n where $1 \le n \le K$. d_i = distance between the volume origin and the source. $F(x, y, z)$ = value of voxel (x, y, z) in volume F. Volume F in xyz space and Projections are in *uv* space.

Backprojection: The volume F is reconstructed using the following equations:

$$F(x,y,z) = \frac{1}{2\pi t}\sum_{i=1}^{t} W_2(x,y,i)\,Q_i\big(u(x,y,i), v(x,y,z,i)\big),$$

Co-ordinates

$$u(x,y,i) = \frac{d'(-x\sin\theta_i + y\cos\theta_i)}{d_i - x\cos\theta_i - y\sin\theta_i}$$

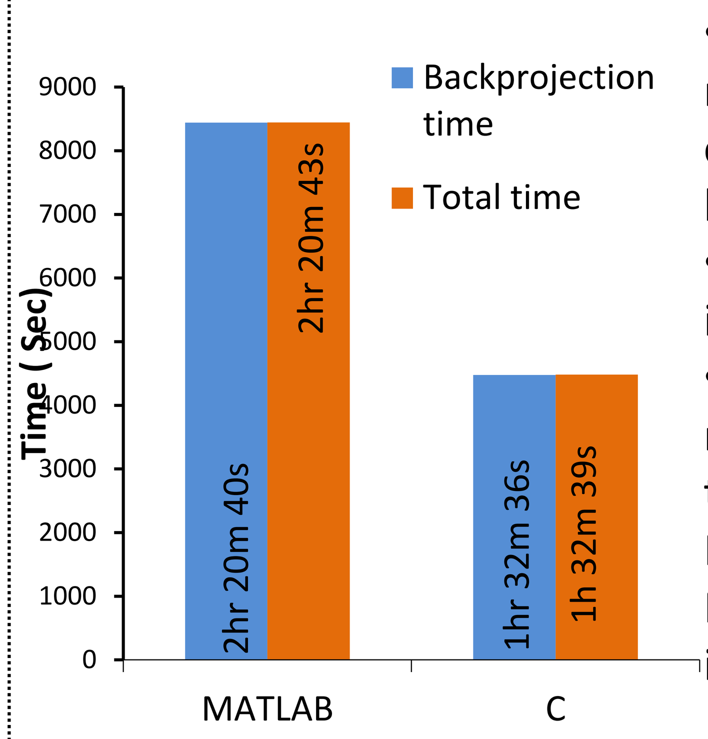
$$v(x,y,z,i) = \frac{d'z}{d_i - x\cos\theta_i - y\sin\theta_i}$$

Weight value,

$$W_2(x,y,i) = \frac{d_i}{d_i - x\cos\theta_i - y\sin\theta_i}$$

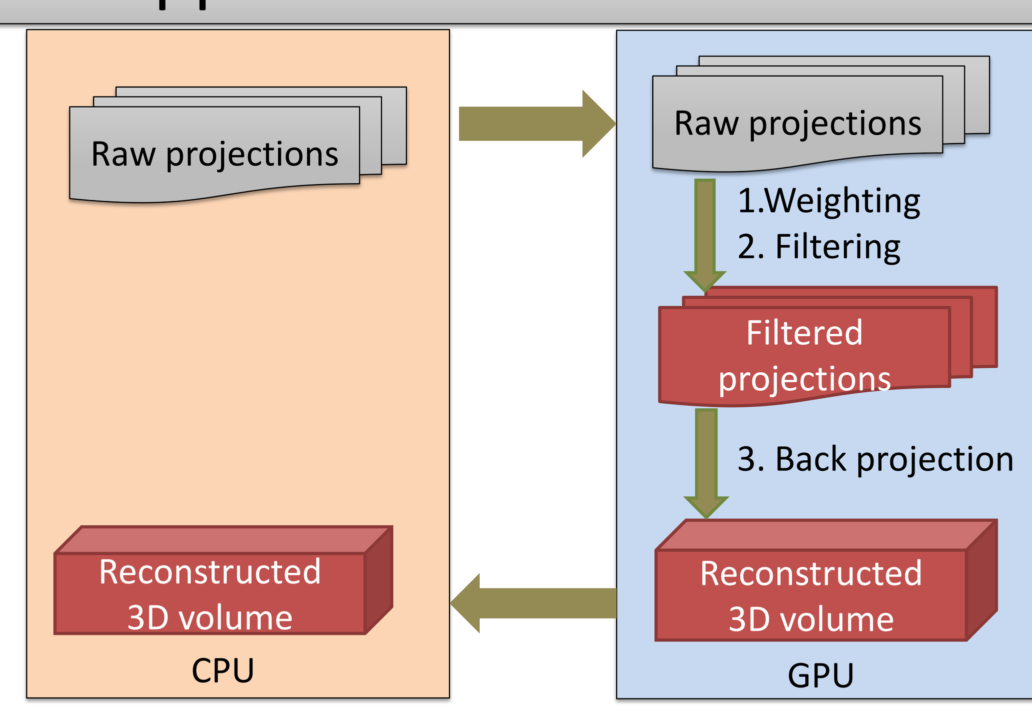
Advantage: Faster reconstruction of the final volume will help in treatment/diagnosis of patients. Capturing data takes only ~9 seconds and reconstruction takes ~3 hours.

Motivation

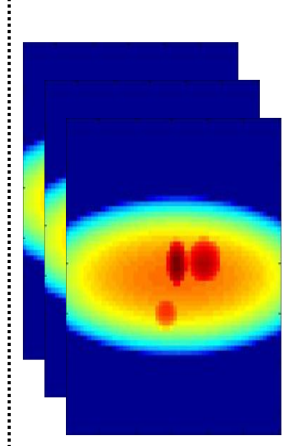


- Back projection has most intensive computations, but highly parallelizable.
- Different voxels are independent.
- Fessler's image reconstruction toolbox[6] implements Feldkamp CBCT in MATLAB. Widely used in academia.

Our approach

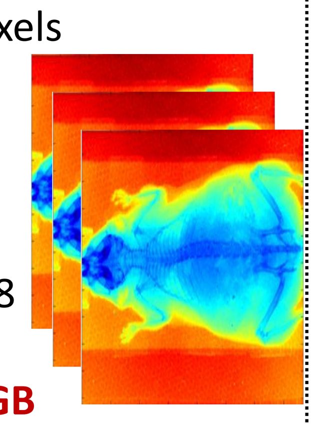


Sample Projections



Mathematical phantom
Input: 64 × 60 pixels with 72 projections
final volume: 64 × 60 × 50 voxels
Size = 1 MB + 1 MB

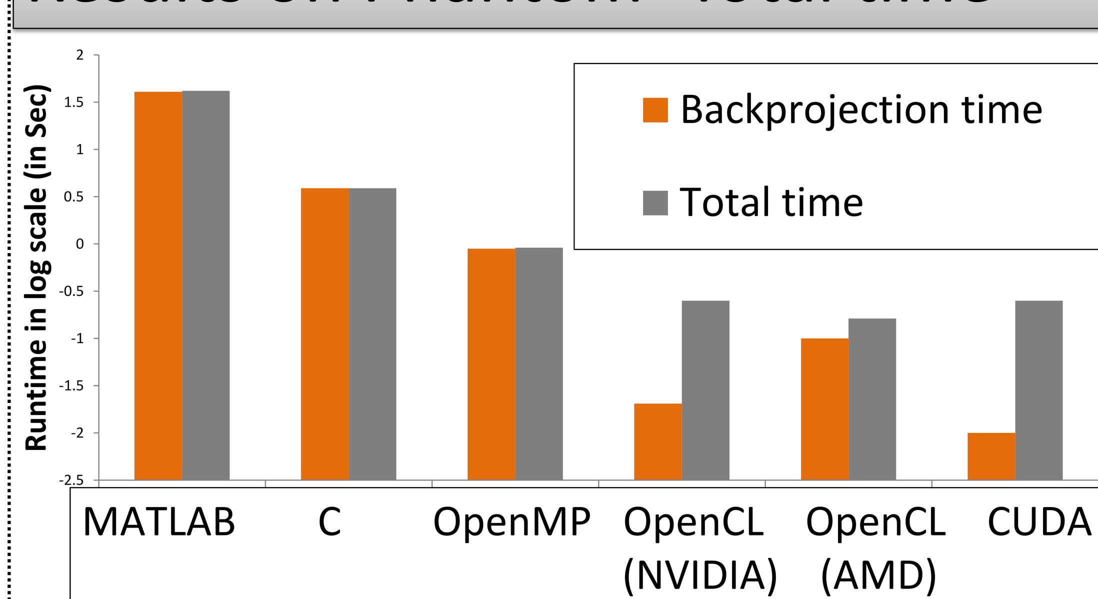
Mouse scan
Input: 512 × 768 pixels with 361 projections
final volume: 512 × 512 × 768 voxels
Size = 542 MB + 768 MB > **1GB**



Architectures and Languages used

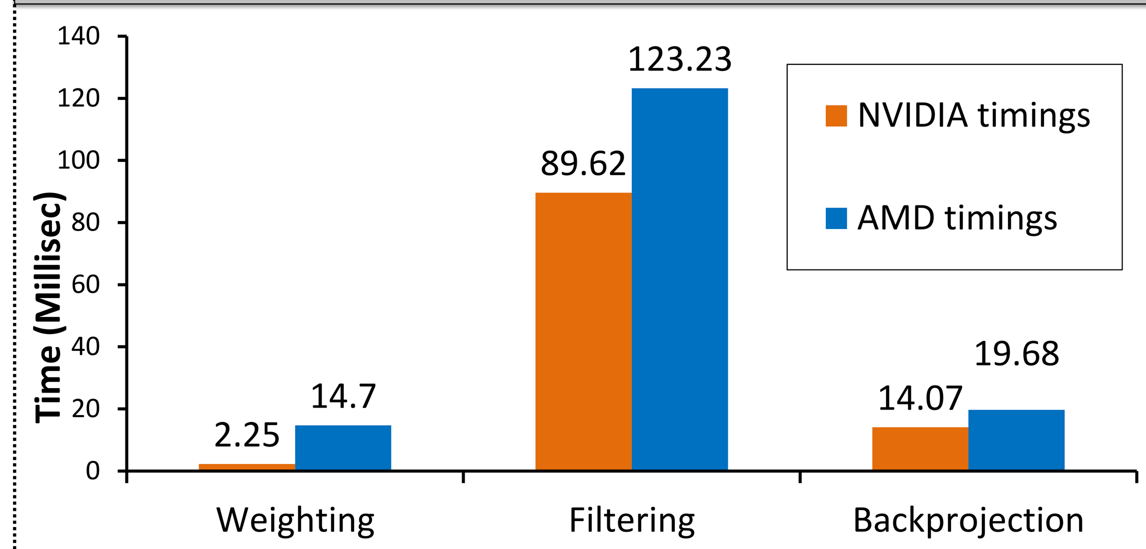
Host	Device	Language
Intel Xeon CPU E5-2620 0 @ 2.00GHz with 6 cores, Cache size: 15MB, RAM size: 32GB.		MATLAB MATLAB PCT C C with OpenMP
	NVIDIA Tesla C2075 AMD Raedon HD5870	CUDA OpenCL

Results on Phantom- Total time

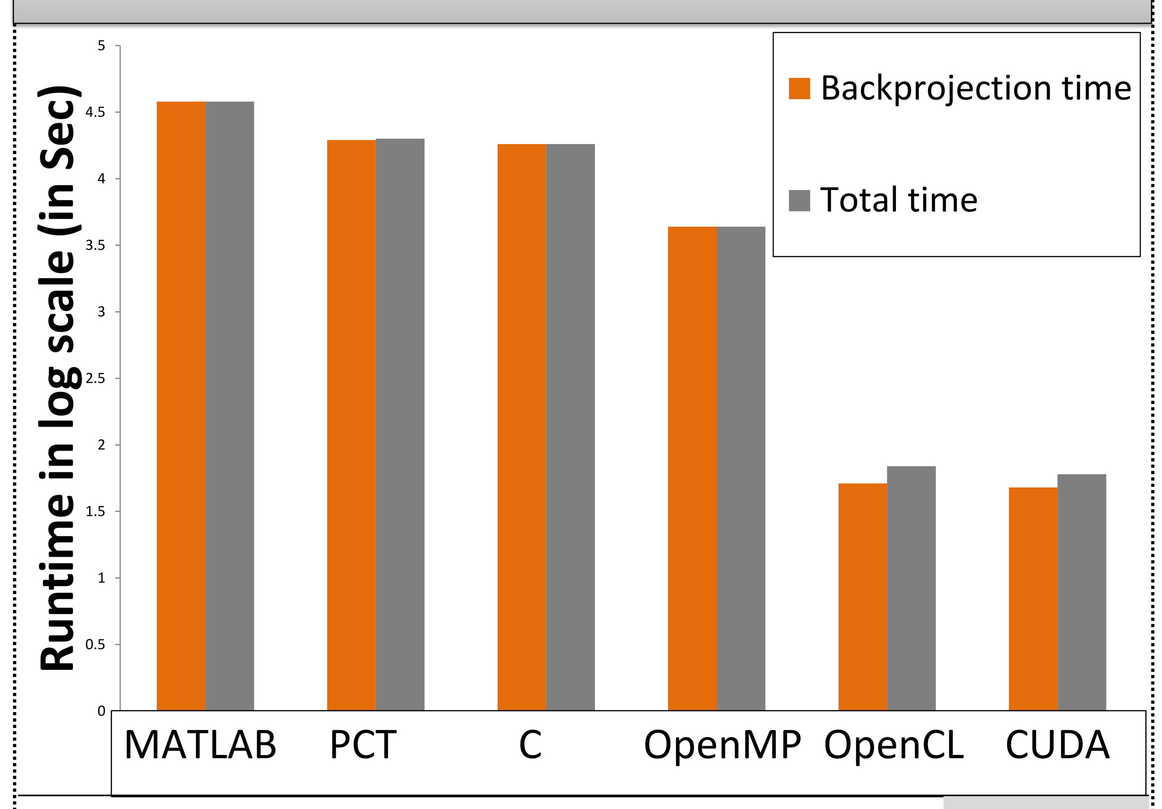


Programming Paradigm	Speedup over single threaded MATLAB	Speedup over single threaded C	Speedup over multi-threaded C
C with OpenMP	45x	4x	--
OpenCL (NVIDIA)	2026x	200x	45x
OpenCL (AMD)	400x	40x	8x
CUDA	4500x	430x	100x

Result on phantom- kernel runtime



Results on Mouse- Total time



Programming Paradigm	Speedup over single threaded MATLAB	Speedup over multi-threaded MATLAB	Speedup over single threaded C	Speedup over multi-threaded C
MATLAB PCT	2x	--	--	--
C with OpenMP	10x	5x	4x	--
OpenCL (NVIDIA)	700x	385x	350x	85x
CUDA	800x	415x	380x	90x

Future work

- Optimize other GPU kernels
- More configurations to be tested with auto-tuning
- Streaming for bigger datasets
- Overlapping computation and communication
- Improve performance on AMD device

References

[1] S. Mukherjee, N. Moore, J. Brock, M. Leeser, CUDA and OpenCL Implementations of 3D CT Reconstruction for Biomedical Imaging, Proc. of IEEE High Performance Extreme Computing, (2012).
[2] L. A. Feldkamp, L. C. Davis, J. W. Kress, Practical cone-beam algorithm, J. Opt. Soc. Am., Volume 1(A), (1984).
[3] F. Xu, K. Mueller, Real-time 3D computed tomographic reconstruction using commodity graphics hardware, Physics in Medicine and Biology, 52(12) (2007).
[4] F. Ino, S. Yoshida, K. Hagihara, RGBA Packing for Fast Cone Beam Reconstruction on the GPU, Proc. of SPIE, Vol. 7258, (2009).
[5] NVIDIA corporation, NVIDIA CUDA C Programming Guide, CUDA Toolkit 4.1.
[6] Fessler's image reconstruction toolbox, http://www.eecs.umich.edu/~fessler/irt/fessler.tgz.

Acknowledgements

NSF
MathWorks
MASSACHUSETTS GENERAL HOSPITAL

More information and software available: http://www.coe.neu.edu/Research/rcl//projects/CBCT.php