

MIC-GPU: High-Performance Computing for Medical Imaging on Programmable Graphics Hardware (GPUs)

Visualization and Final Remarks

Klaus Mueller

Computer Science
Center for Visual Computing
Stony Brook University



Interactive Visualizations

Standard axis-aligned slices

Rapid reconstruction affords quick injection of more involved visualizations/renderings:

- arbitrary slices (non-axis aligned slices) → 3D slicing
- 3D X-ray views from arbitrary view points
- full 3D volume renderings

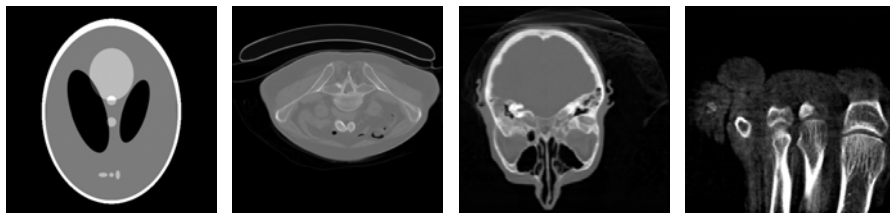
All have been shown to run at 20-30 frames/s

- recall, GPUs are meant for graphics
- data is already in texture memory
- simply load another kernel into the shaders
- frames (projections) produced for visualization have similar costs than frames (projections) consumed for reconstruction

Streaming CT

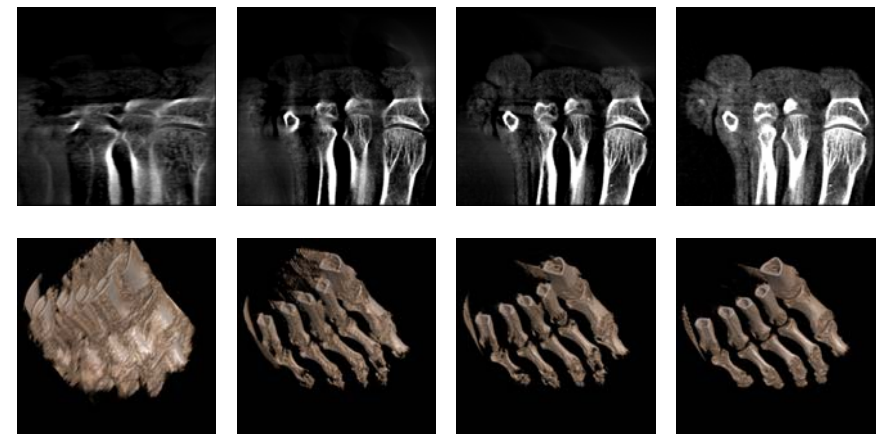
Reconstruct (consume) incoming (produced) projections without buffering

- from 360 1024^2 projections to a 512^3 volume at full floating point precision
- as shown, latest GPUs can achieve 30 projections/s reconstruction speed



Streaming CT With Direct Visualization

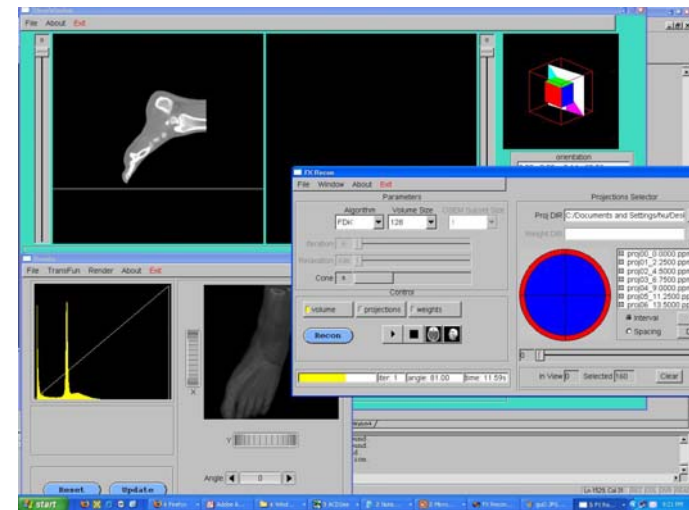
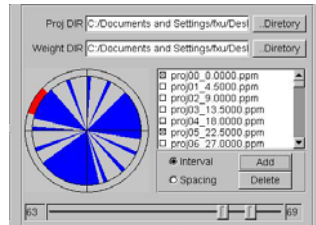
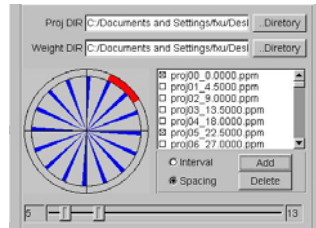
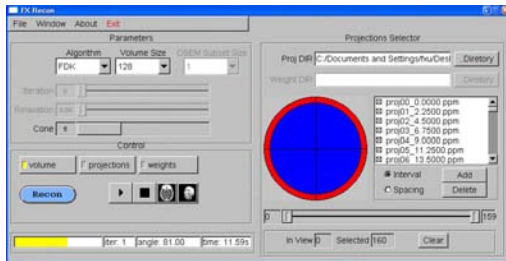
Watch the object evolve as it is acquired



Edit/tune on the fly:

- parameters
- projection sets
- algorithms

Couple with 2D/3D visualizations



Have shown that:

- GPUs are an excellent and very flexible platform for CT reconstruction
- GPUs are bound to become even more attractive for this purpose
- additional advantages provided by excellent visualization capabilities
- CUDA and CTM will provide even better interfaces for MIC-GPU computing
 - thread management
 - memory management
 - access to more generalized computational resources
 - but with the extra benefit of super-fast interpolation, rasterization, and texture interpolation

Introduction

GPU architecture, programming model, and programming facilities

GPU programming examples (image processing)

CT reconstruction pipeline components

GPU-acceleration of individual components

Various CT reconstruction pipelines, load balancing and load estimation

Reconstruction visualization and final remarks

Further Information

Check at <http://www.rapidCT.com> for latest:

- tutorial updates
- fragment code samples
- executable applications of all routines (soon)
- applications
- publications
- bulletin board
- contacts info
- community news and feedback

Any Questions?

