Next Generation Image Processing for Computed Tomography

CASIS 2013

05/22/13

Karina Bond, Dan Schneberk, Trevor Willey, Steve Azevedo, Harry Martz

Lawrence Livermore National Laboratory

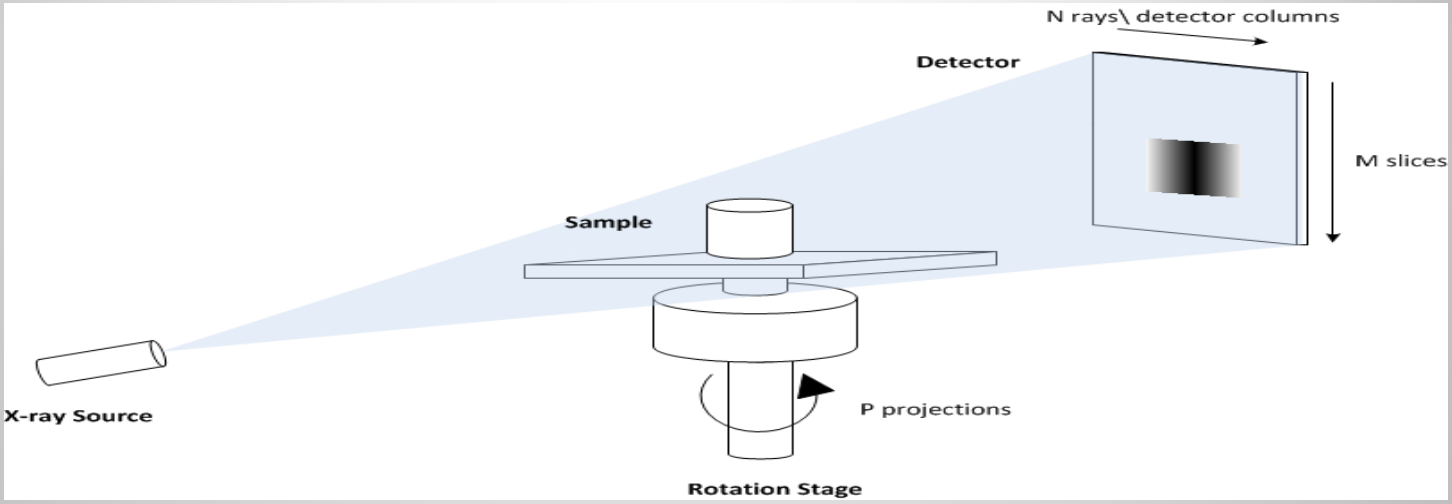
LLNL-PRES-636374

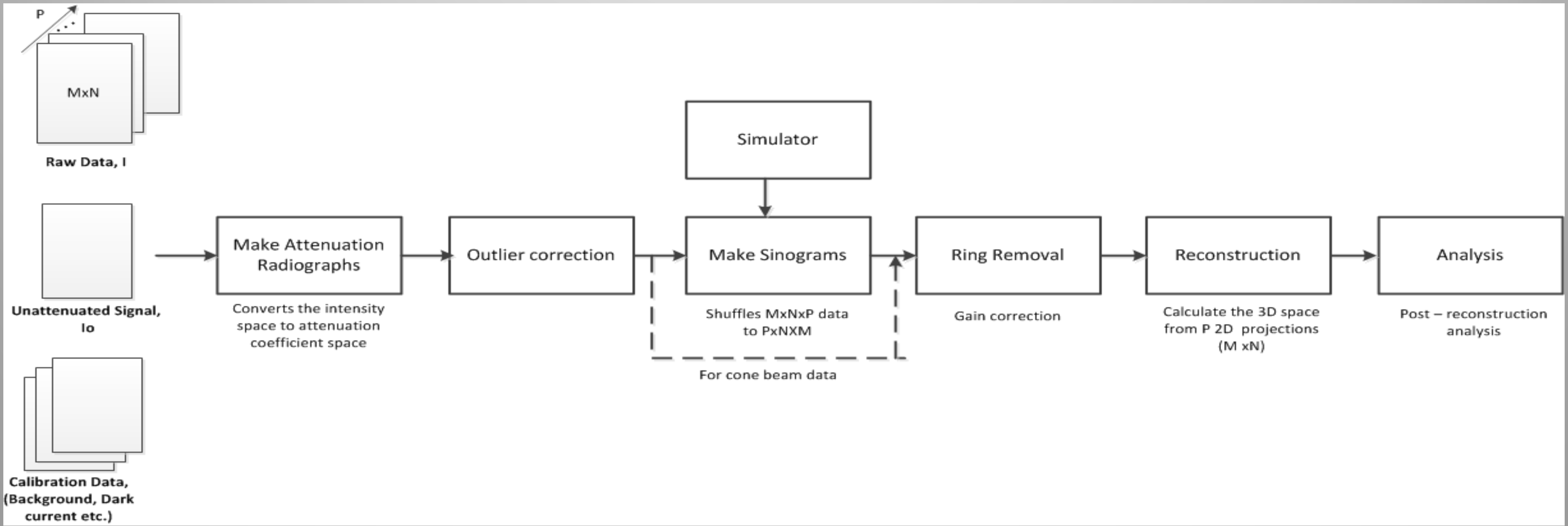
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Summary

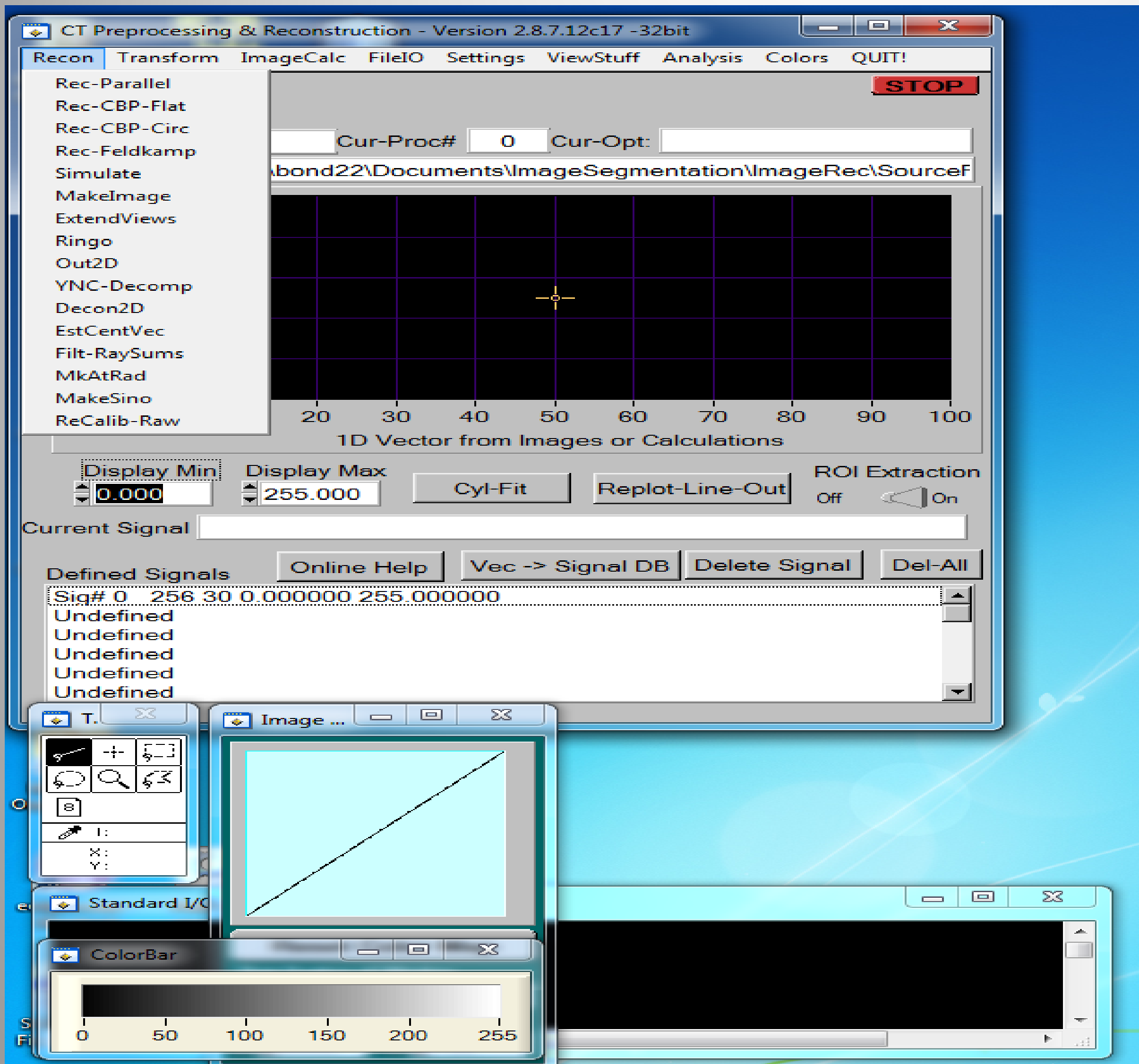
- A brief overview of IMGREC – the workhorse Computed Tomography (CT) code at LLNL
  - Large CT processing feature set.
  - Wide customer base
- Plans to maintain IMGREC, and assess requirements for new and improved CT image processing toolbox.
  - Software-development Methodology
  - Key Requirements driving a re-design of the current IMGREC
  - Software architecture options

CT Measurement and Processing





IMGREC – CT Processing



Making Attenuation Radiographs

- aSi (amorphous silicon)
- Camera-scintillator
- CMOS

Outlier Removal

- Image is median filtered (3x3) following by thresholding.

Making Sinograms

- Re-shuffles the data for slice by slice reconstruction

Ring Removal

- Processing to remove ring artifact associated with non-uniform gain in the detector pixels

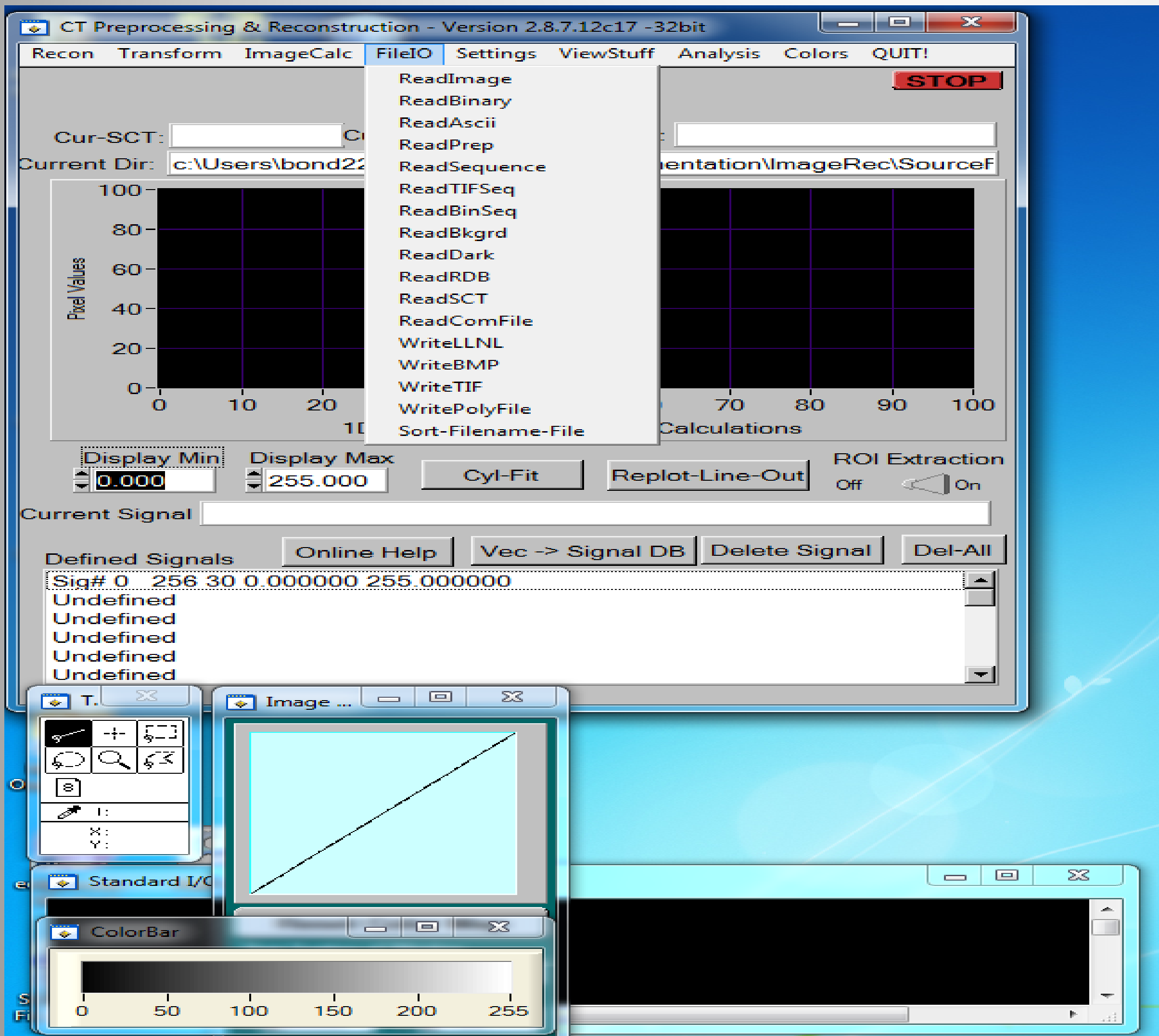
Beam hardening correction

- Based on beam hardening coefficients provided by the user

Reconstruction (analytical)

- Parallel Beam
- Fan Beam , flat detector
- Fan Beam, curved detector
- Cone beam (Feldkamp)

IMGREC – File Input/Output



Images
- sdt/spr format
- pgm,pfm,tif,bmp
- User-defined binary

Image Sequences
- sdt/spr
- User-defined binary
- tif

CT paramater file
- sct file

Misc. files
- Dark current (sdt/spr)
- Background (sdt/spr)

Script files
- Text files with supported commands and arguments

IMGREC Tools

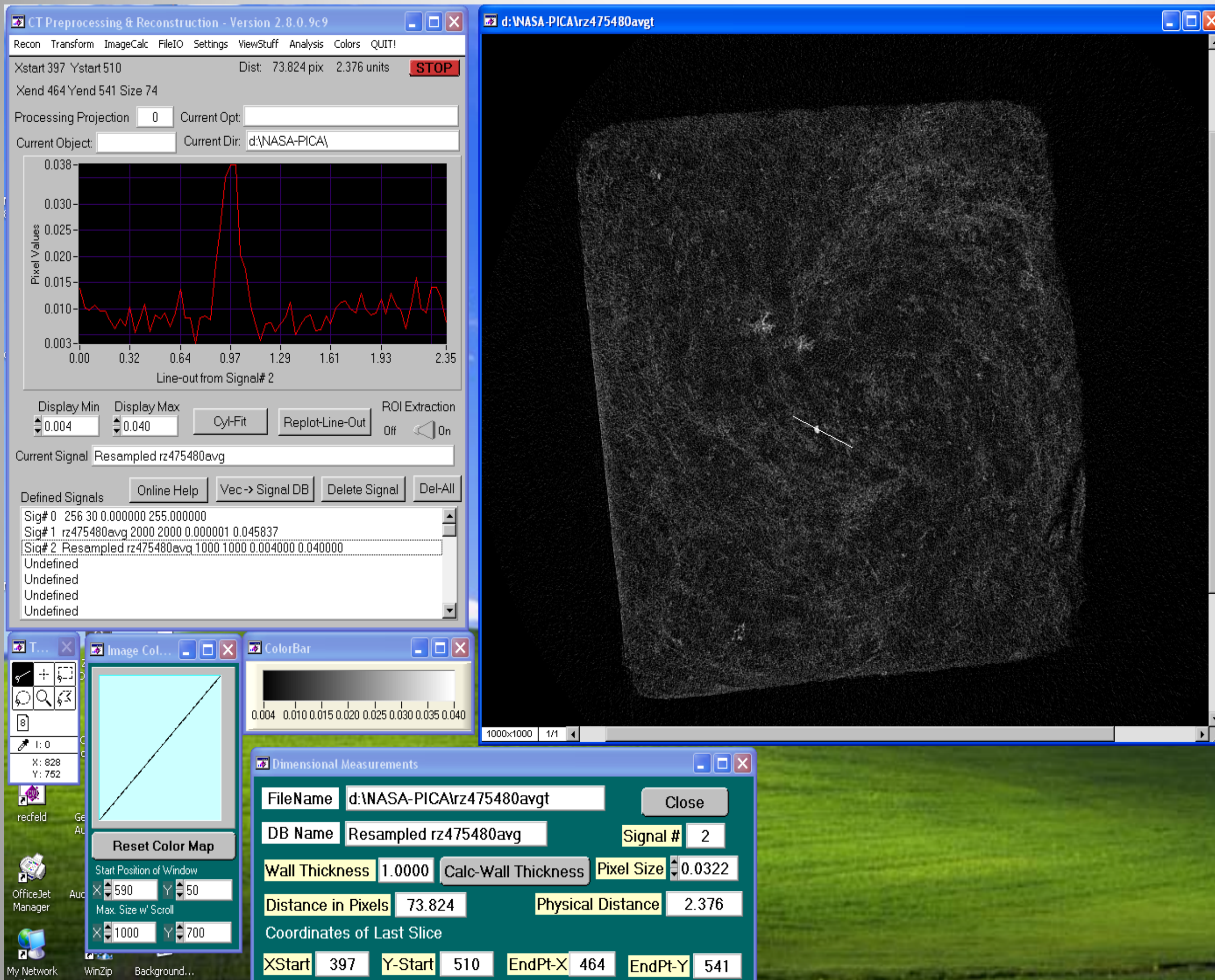


Image Operations

- The Line-out tool
- The Pixel=Value Query tool
- The Image Extraction tool
  The Image Magnifier tool
- The Draw Circle Tool
- The Draw Polygon Tool

Image Color Map

- To adjust the contrast and brightness
- To adjust the max\min intensities to display

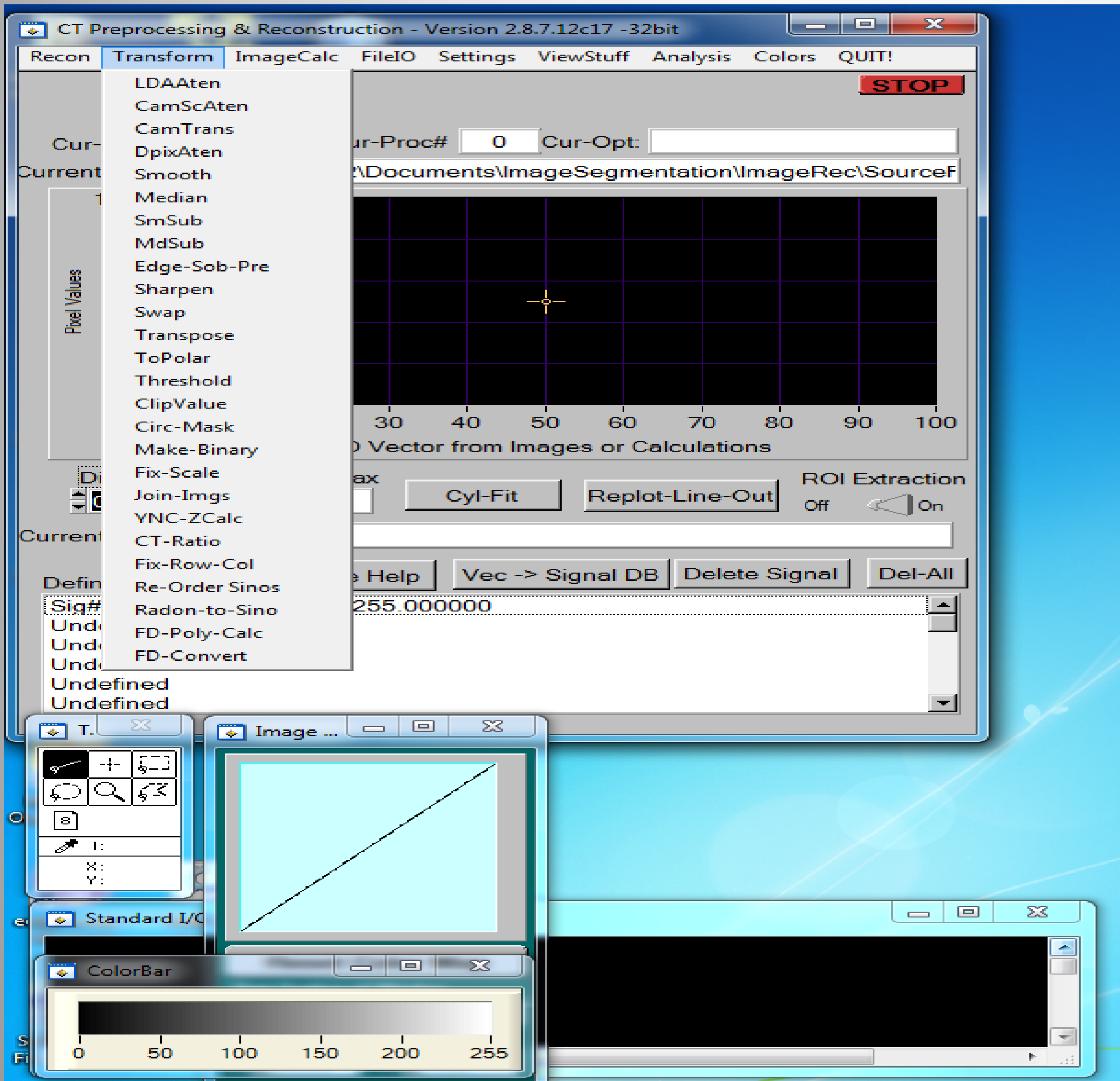
Quantitative color bar

- Shows relationship between the colors and intensity values

Dimensional Measurement Tool

- Calculate wall thickness
- Calculate pixel size
- Length of line drawn on the image

IMGREC – Image Transform Menu



Making Attenuation Radiographs

- make attenuation radiographs for LDA\Camera-scintillator\Dpix panels

Filtering

- Averaging\Median
- Subtract the input image from the averaged\median filtered image
- Edge detection (Sobel)
- Sharpening (Laplacian transform)
- Image is median filtered (3x3) following by thresholding.

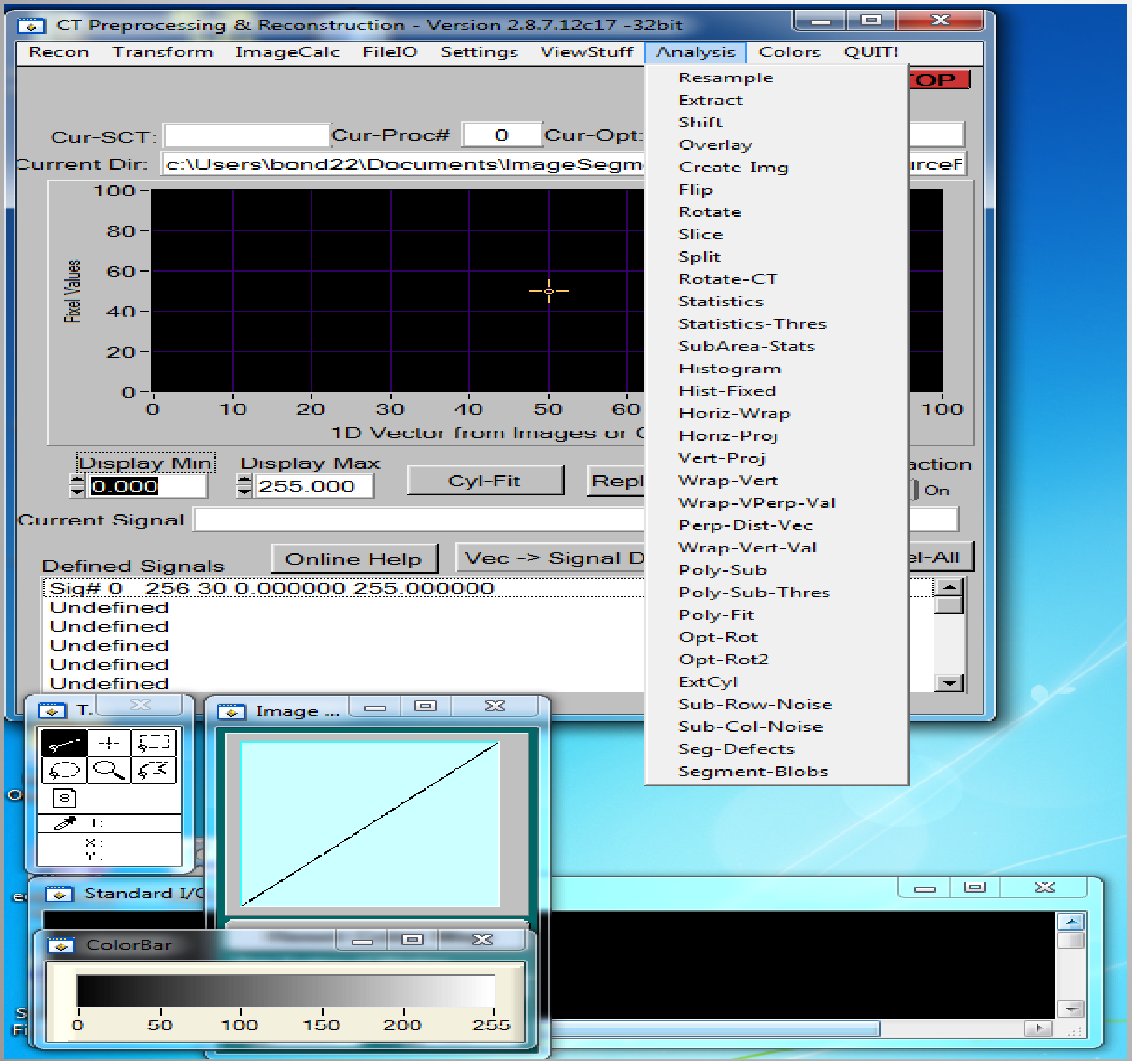
Thresholding

- Replace values above a certain value.
- Replace values below a certain value.
- Make binary image (value below a certain value a re ‘0’ and the rest ‘1’)
- Apply a circular mask to the data

Misc. processing

- Transpose
- Swap (bit level)
- Fix scale
- Join images

IMGREC – Analysis Menu



Resampling

- Bi-linear

Image processing

- Extract
- Shift
- Rotate
- Flip
- spilt
- slice

Histogram\Statistics

- Whole image
- User defined intensity range or ROI

Misc. processing

- Polynomial fit to the data
- Calculate optimal rotation
- Extract cylinder

About IMGREC….

- Emphasis on maintaining maximum **precision** through the processing pipeline (all floating point calculations).

- User has access to **intermediate data**.

- **Scripting** is supported to facilitate automation and batch processing.

- Written in **ANSI C** and theoretically portable – an R&D tool.

- Access to a variety of **pre-processing and reconstruction algorithms**.

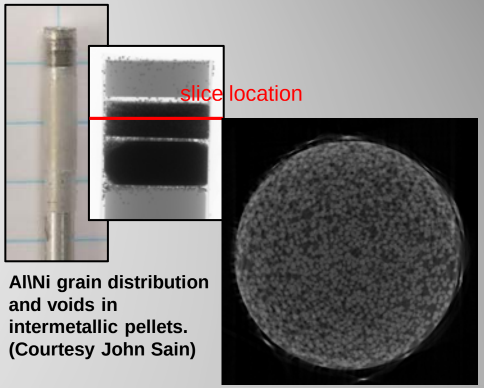
- Variety of image **analysis tools** are supported.

- Some of the processing is implemented using **threads for increased speeds** on machines that have multiple cores.

- Currently only has support for running on Windows (32-bit and 64-bit).

- Implemented in National Instruments LabWindows (ANSI C code).

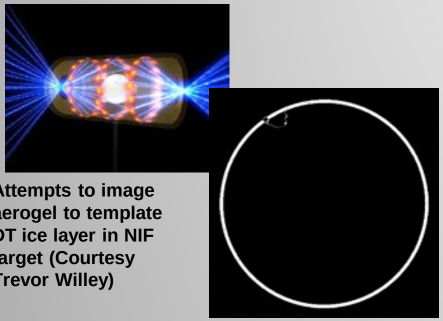
IMGREC Customers



Al\Ni grain distribution and voids in intermetallic pellets. (Courtesy John Sain)

Global Security

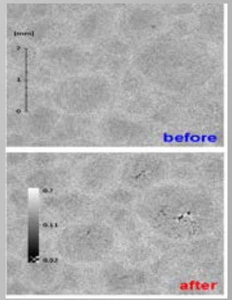
- Explosive characterization in Livermore explosive Detection program (LEDP)
- DOD Ordnance and Components: Analysis of conventional ordnance, ordnance components, and rocket motors.
- Visualization of microstructure within pressed intermetallic materials (S-Program, Predicting Shock Response)



Attempts to image aerogel to template DT ice layer in NIF target (Courtesy Trevor Willey)

NIF

- Inertial Confinement Nuclear Fusion Targets: Uniformity of Aerogels to Template Deuterium-Tritium Ice Layers
- Doping of Be for precise x-ray opacity & energy deposition into D-T
- Defect detection (and/or quality control) of materials and components used in NIF/Omega target assemblies



Microstructure of High Explosives before & after temperature cycling **(Courtesy Trevor Willey)**

WCI work

- Analysis of weapons and weapon components
- Microstructure of Insensitive High Explosives. How external insults (temperature / creep) affect microstructure? Visualization of potential impurities.
- Explosive and binder mixing: Empirical 3D input to modeling

ENG/CMNT , LLNL

- Visualization of additive manufacturing structures

PLS

- Detection and Concentration of Nanoparticles in Insects. Determine how cockroaches metabolize relatively toxic nanoparticles

LBL

- Possible partnership with ALS\NERC to include image processing algorithms in IMGREC as part of a measurement, storage and processing pipeline that is currently in development.

At external Labs

- Tyndall, TSL, Pantex etc.

“IMGREC maintenance and advancement” project goals

The goal of this project is propose how to modernize IMGREC so that it will support existing and future laboratory customers for many more years.

This goal involves:

Maintenance

- Understand and document the capabilities of IMGREC.
- Institute modern software management practices and version control.
- A process for configuration management (testing, release and managing sub-system dependencies) of IMGREC in conjunction with other software in the current NDE processing pipeline.

Advancement

- Determine the use cases of an IMGREC-like CT image processing toolbox
- Identify requirements for new development based on use-case findings
- Propose a plan of action to satisfy the identified requirements
- Initiate implementation of the plan of action. (stretch goal)

Future Development Methodology

Use-Case Capture (66% complete)

- Meet with individual classes of users to understand how they interact with IMGREC.
- Generate Use-Case (simple scenarios describing the user interactions with the application) Document.
- Stake-holders must review and sign-off on this use-case document.

Requirements Identification (25%)

- Requirements will naturally fall out of the Use-Case document; i.e., each use-case will drive one or more requirements.
- Stake holders will review and sign-off on each use-case document.

Software Architecture\Design and development plan

- This document will describe the software architecture and high-level design that will fulfill the requirements in the requirements document.
- The document will set a priority for the implementation of each requirement.

Use Case Example

Title:	**Analytical Fan Beam, Flat detector Reconstruction**
ID	UC - 10
Description:	A user reconstructs a 3D volume from sinograms measured on a scanner with fan beam with flat detector geometry.
Primary Actor:	An experienced user
Pre-conditions:	The user has an - CT parameter file (sct) describing the scanner geometry and processing parameters. - CT data in sinogram format.
Post-conditions:	The reconstructed slices will be saved to working directory in sdt\spr format.
Main Scenario:	1. The user selects the “Read CT Parameter” option from the “File IO” menu and selects a SCT file for sinograms captured from a scanner with fan beam with flat detector geometry. 2. The user might do some processing on the sinograms like ring removal and or beam hardening correction. 3. The user might adjust the reconstruction parameters in the “CT Parameters” dialog of the “Information” menu. 4. The user selects the “Fan Beam – flat detector Recon” from the “Reconstruction” menu on the GUI. 5. The system calculates the fan beam with flat detector reconstruction of from the input sinograms. 6. The system stores the reconstructed data to disk in the SDT\SPR format along with the associated SCT file.
Extensions:	1a. The most recent data in the system is a set sinograms for fan bean with flat detector beam geometry scanner that were calculated from attenuation radiographs.
Variations	
Frequency of Use:	High
Status:	
Owner:	
Priority:	

Key Requirements driving a new architecture for IMGREC

- **Processing Speed :**
  Use portable parallel processing API (OpenMP) and/or GPUs to improve speed.
  Current speed improvements (IMGREC algorithms stripped out of the GUI with slight modifications and addition of OpenMP):
  >10x speedup of cone beam reconstruction algorithm.
  >10x speedup per-slice parallel beam

- **Portability:**
  Move to a portable architecture, leverage LC, efforts at ALS/LBNL/NERSC

- **An intuitive user-interface that follows the workflow.**

- **Extensibility to iterative reconstruction algorithms.**
  - Increased memory requirements
  - Increased processing cycles (leverage parallel processing or GPUs)

Recent Improvements to Parallel-Beam Reconstruction in IMGREC

- **Code for filtering improved**
  - rewritten for fftw3
  - OpenMP implemented to filter multiple slices simultaneously
  - Two sinograms simultaneously by each thread; one in real, one in imaginary
- **Code for backprojection improved**
  - OpenMP enables straightforward parallel implementation
    - y-loop is outermost loop and is parallelized
      - ensures thread safety; each is working on different line
    - serial code incremented pointers; indexed arrays required for parallel code
  - enabled ability to process multiple slices in memory
  - optimizations in gcc & pgi are much better than NI C compiler
- **Significant improvement (>>10X) realized on 8-processor desktop**

original code

```
for (iproj=0; iproj<Nproj; iproj++)
{
  …
  for (iy=0; iy<Ny; iy++)
   {
     …
     for (ix=0; ix<xsize; ix++)
     {
       …
     }
   }
}
```

outermost loop: sinogram line

OpenMP: distribute iy loop iterations to threads
outermost loop: reconstruction rows
added multiple slices processed in memory

easy method to ensure thread safety: all parameters must ultimately be computed from lookup tables indexed by iy

innermost loop: reconstruction columns

new code

```
#pragma omp parallel for private(iz, …
for(iy=0;iy<Ny;iy++)
for(iz=0;iz<Nz;iz++)
{
  …
  for(iproj=0;iproj<Nproj; iproj++)
  {
    …
    for(ix=0;ix<xsize;ix++)
    {
      …
    }
  }
}
```

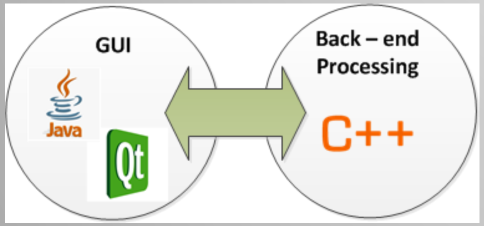
Options for a fast, portable and extensible CT processing software package?

Keep the (back-end processing) algorithms separate from the UI code.

Option 1: Java or Qt for the UI and the back-end processing.

Pro: All the source code is in one development project. Interfaces are somewhat flexible.
Cons: Less flexibility in the optimizing algorithms for speed.

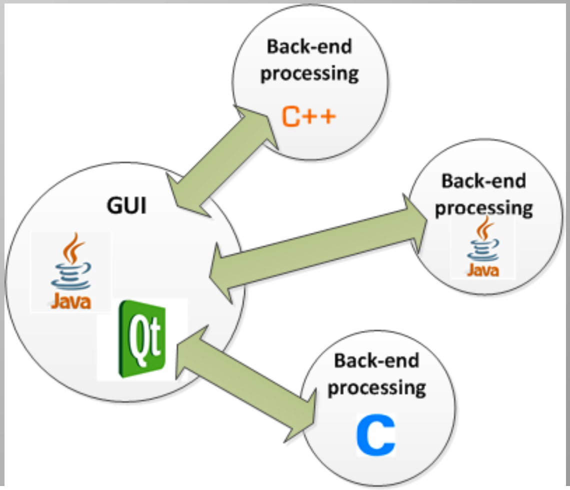
Option 2: Java or Qt to implement the UI. One application for back-end processing in C++ or C.



Pros: Algorithms can be optimized for speed without needing UI support.
Cons: Interfaces between the UI and back-end need to be somewhat well – defined and enforced.

Option 3: Java or Qt to implement the UI. Multiple back-end processing application in different languages.

Pros: Research algorithms can be called from the UI without needing to port the code.
Cons: More interfaces to manage. Complicated software architecture.



Summary

- A brief overview of IMGREC – the workhorse Computed Tomography code at LLNL
  - Large CT processing feature set.
  - Wide customer base

- Plans to maintain IMGREC, and assess requirements for new and improved CT image processing toolbox.

  - Software-development Methodology

  - Key Requirements driving a re-design of the current IMGREC

  - Software architecture options