A fully-automated, high-throughput reconstruction and analysis pipeline for quantitative imaging in CT

J Hoffman, M Wahi-Anwar, N Emaminejad, G Kim, M Brown, M McNitt-Gray

David Geffen School of Medicine, UCLA, Los Angeles, CA

Introduction

- Our goal is to study the impacts of CT protocol selection (e.g. dose, reconstruction, etc.) on quantitative imaging (QI) measures.
- Previous efforts [1] **required substantial time (>6 months) and labor (5 individuals)** to build datasets (381 patients, 1443 total reconstructions)
- **Logistical overhead** of case collection, on-scanner reconstructions, and data management of terabyte-scale collections of reconstructions **slows or prohibits large-scale investigations**
- We have developed a pipeline that achieves full automation of reconstruction, data-management and QI analysis
- **This work outlines the design of the pipeline, execution and preliminary observations of performance improvements in pilot studies**
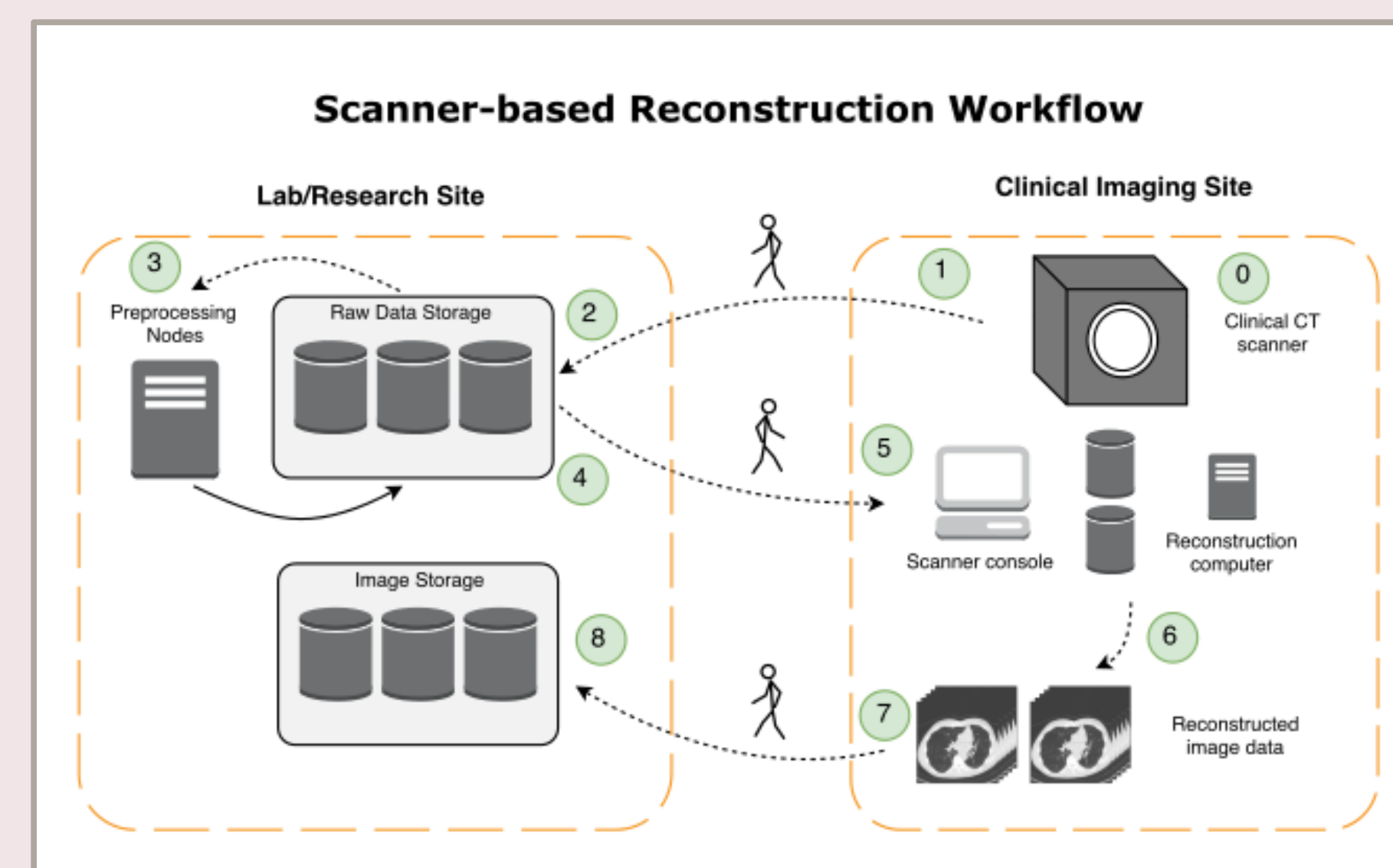
Background

Figure 1: Illustration of previous QI study reconstruction workflow. Dashed lines indicate required human intervention. (0) A patient is scanned using a clinical protocol. (1) Raw data is collected via export to encrypted external hard drive and (2) is subsequently imported to long-term storage. (3) Preprocessing (e.g. simulated dose reduction via noise addition) is applied to the collected raw data and processed files are stored back in long-term storage. Desired reconstructions are decided upon, (4) preprocessed raw data is exported from long-term storage, (5) re-imported to the scanner, and (6) reconstructed according to desired protocols. (7-8) Reconstructed image data is exported and returned to long-term image storage for QI analysis.

- In previous work, limitations were primarily due to scanner interface, data management (import/export from scanner), and **required user intervention (configuration, GUI interfaces, etc.) that led to computational latency** (Fig. 1)
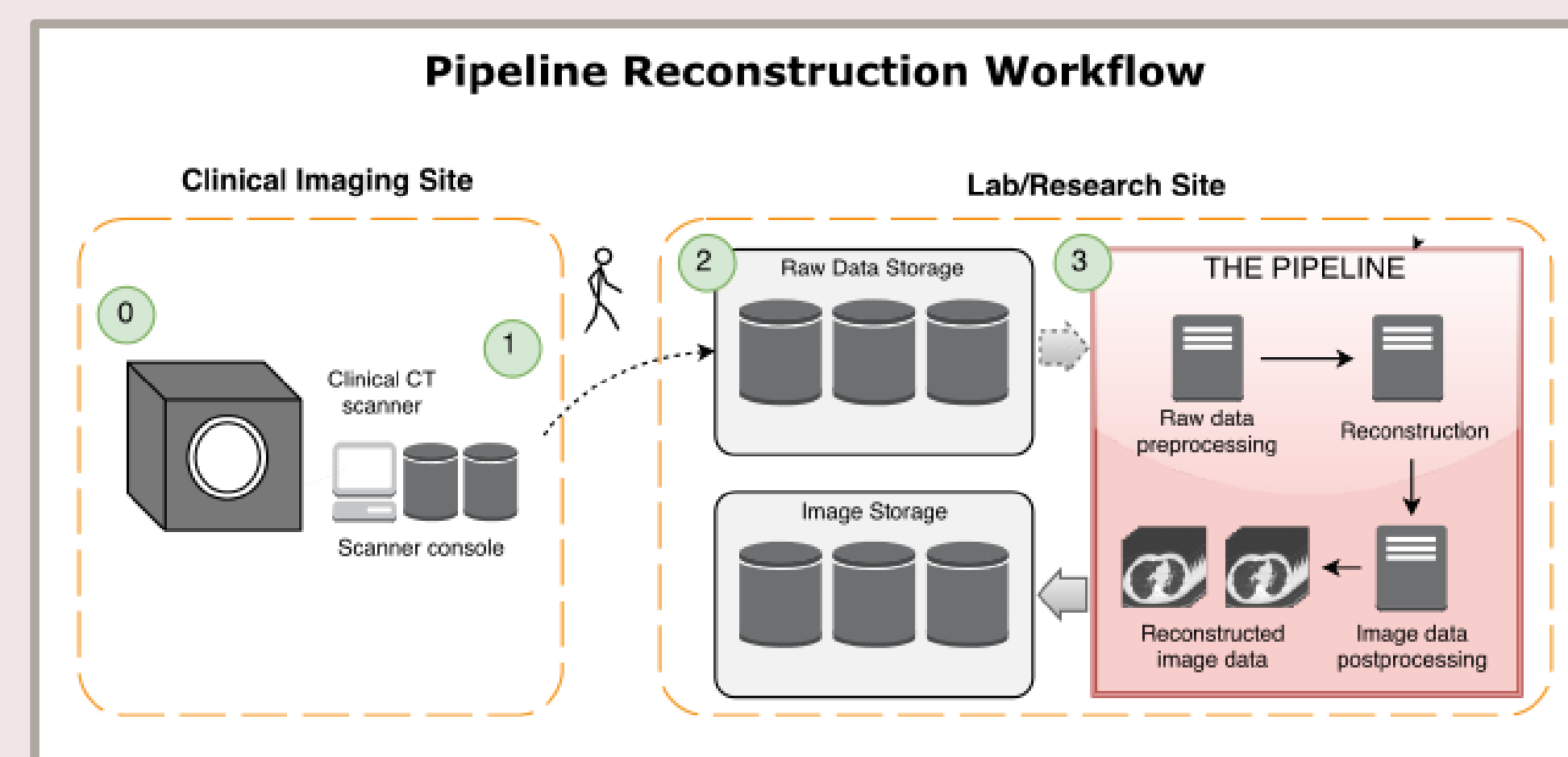

Figure 2: Overview of the automation framework for reconstruction removes user intervention after collection of raw data from scanner (1). The automated approach (2-3) reduces resource latency and eliminated potential configuration errors compared to approach outlined in Fig. 1. Step (3) is outlined in detail in Fig. 3.

- A more efficient, simpler approach utilizing a fully-automated "pipeline" can leverage preexisting resources (i.e. FreeCT_wFBP [3]) and accelerate existing efforts in quantitative imaging dataset generation (Fig. 2)

Pipeline Design and Implementation

The launcher parses the configuration file and list of cases (i.e. raw projection data files) into a queue listing all of the desired reconstructions for a study. After the launcher generates the queue file, its final job is to spawn the pipeline daemon (dashed line).

Human-readable YAML configuration files (shown above) specify all input necessarily to execute a full pipeline study. A GUI interface (shown below) to configuration is also provided for commonly requested settings and for case list management.

The pipeline daemon runs in the background and manages the queue. As GPU resources become available, the daemon dispatches individual reconstruction jobs as "pipeline queue items" to target devices.

Each **queue item** is responsible for overseeing a reconstruction from start to finish. This includes fetching and management of raw data (i.e. retrieval from network location, ensuring that we do not unnecessarily duplicate large files), preprocessing (e.g. simulated dose reduction, denoising of sinogram data, filtration, etc.), reconstruction into image data, and finally organization of image data and metadata into the study directory for analysis. Many individual queue items are launched during the execution of the pipeline.

As jobs are dispatched and completed, their system exit status is evaluated by the daemon. Successful queue items are pushed onto the "done" stack, while jobs that encountered errors are pushed onto the "error" stack. Tracking and recording of this information allows for rapid debugging and identification of pipeline problems.

The pipeline GPU queuing framework (i.e. launcher, daemon, queue item) is implemented using open-source tools: Python (www.python.org), PyCUDA [2]. The reconstruction code is the free, open-source package FreeCT_wFBP [3]. A free, fully-3D, model-based iterative reconstruction package, FreeCT_ICD, is under development for future use with the pipeline

Image data and reconstruction metadata are output from the pipeline into a highly-organized directory structure for further quantitative imaging analysis. Sample pipeline reconstructions for one patient are shown in Fig. 4.
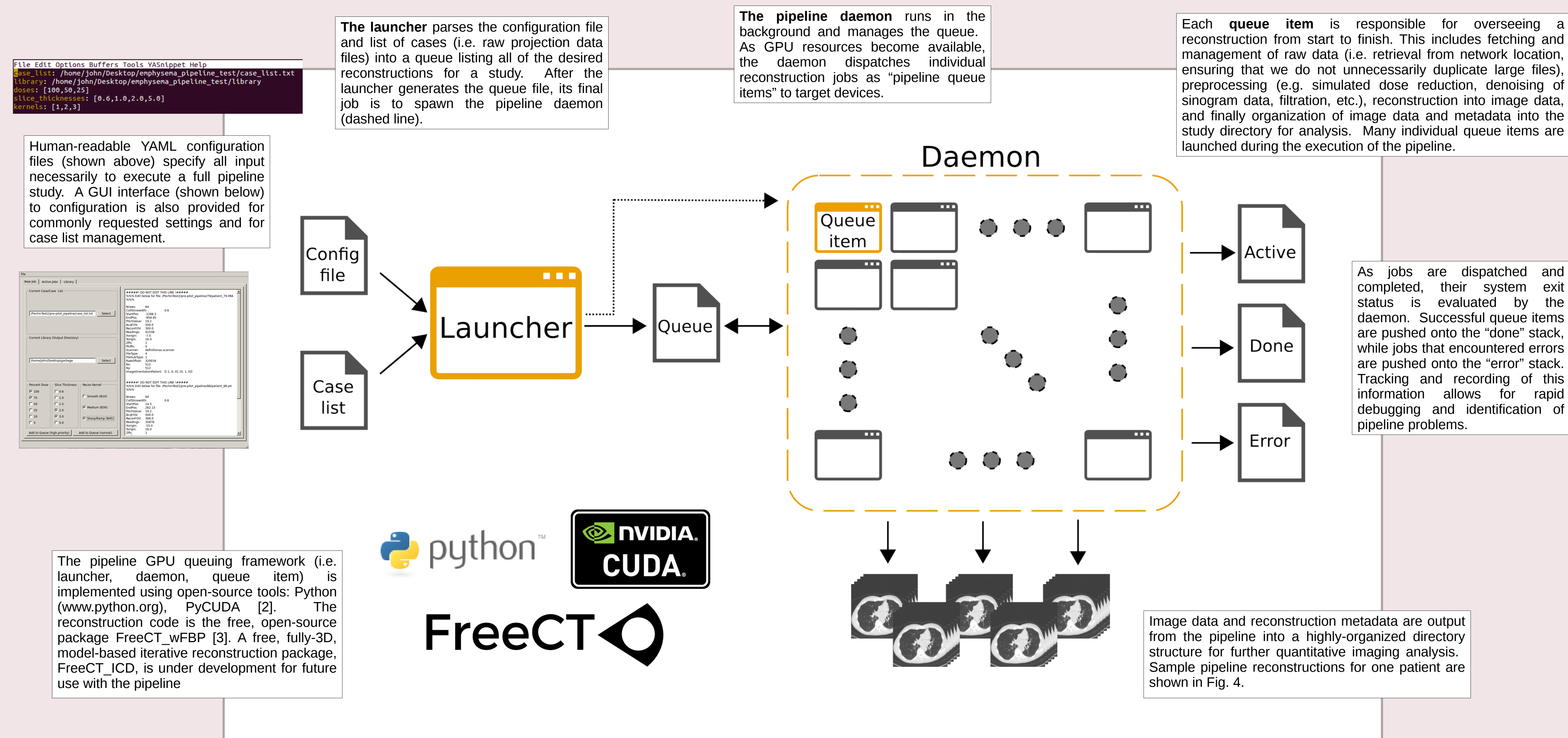

Figure 3: Primary pipeline components (highlighted in orange) are (1) the launcher program, (2) the pipeline daemon, and (3) the pipeline queue items.

Performance Results

- Testing dataset:
  - 5,040 total reconstructions
    - 140 subjects (lung screening CT)
    - **36 reconstructions/subject** (4 dose levels, 3 slice thicknesses and 3 reconstruction kernels) **as shown in Fig. 4**.
- Performance testing hardware:
  - Alienware Aurora R4
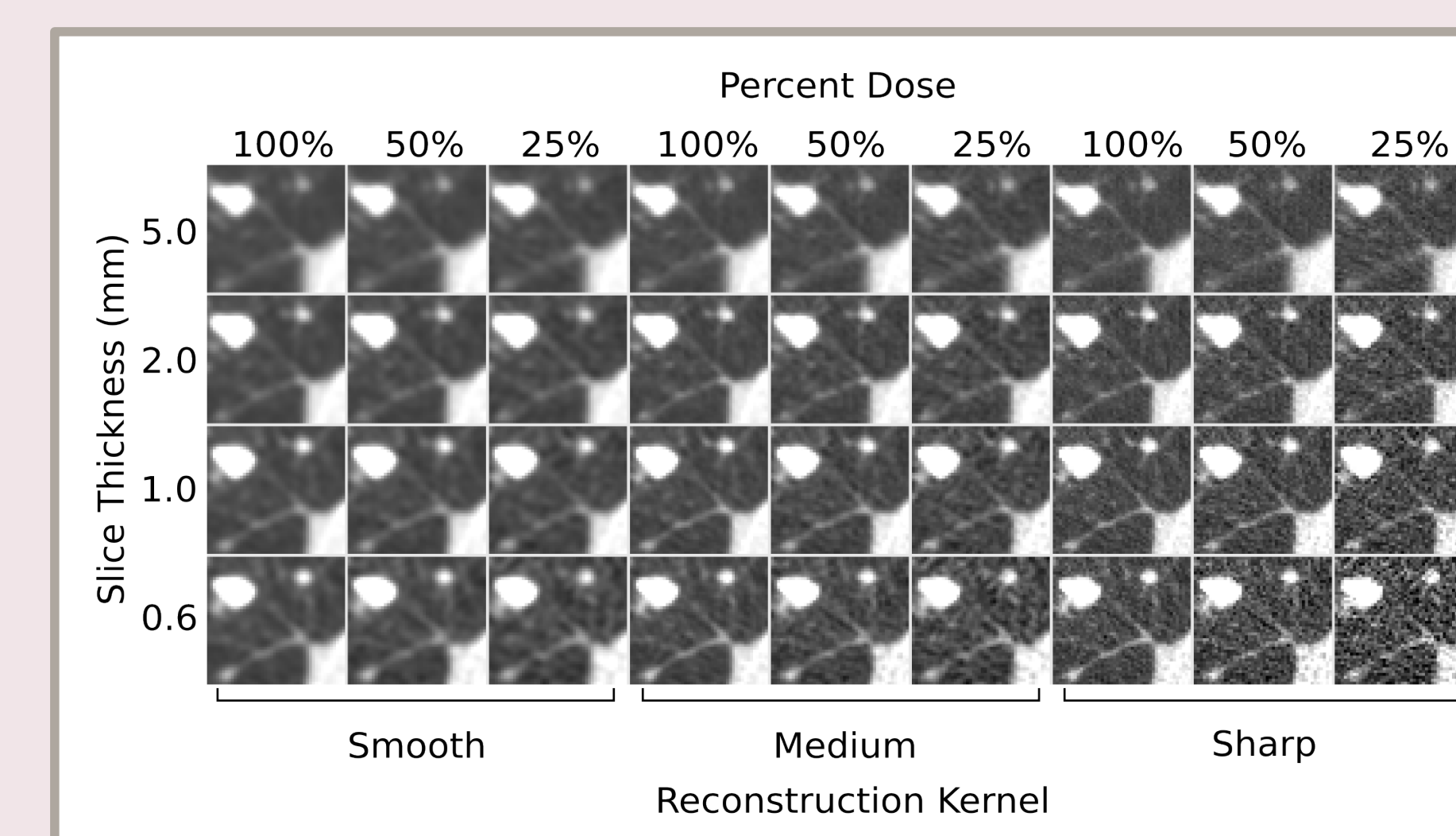   - 2 Nvidia GTX 780 GPUs

Figure 4 (right): Sample of reconstructions of lung parenchyma for one subject. Large numbers of reconstructions across a variety of acquisition and reconstruction parameters can be quickly and efficiently generated for quantitative imaging analysis.

Table 1: Performance results for individual steps of pipeline execution as well as total execution time for testing dataset. A dramatic improvement in over previous experiments (72x) was observed. Additionally, **no human intervention is required** using the pipeline, while [1] demanded significant attention throughout the generation of the dataset.

Mean data fetch time	1.74 s
Mean dose reduction simulation	8.81 s
Mean reconstruction time	4.40 min
Mean queue item time	5.57 min
Total time (2 GPUs)	8.75 days
Acceleration factor over [1]	**72x**

Conclusions

- The pipeline substantially reduces time and effort required for certain types of QI experiments
- **Pipeline is an efficient, fully-automated approach to building datasets for quantitative imaging**
- **Source code available** at: https://github.com/captnjohnny1618/CTBB_Pipeline_Package
- See our other abstracts that utilize the pipeline: **SU-K-201-9, WE-F-605-4, WE-G-201-4**

Download a hi-res version of this poster!

References:

[1] S. Young, P. Lo, G. Kim, M. Brown, J. Hoffman, W. Hsu, W. Wahi-Anwar, C. Flores, G. Lee, F. Noo, J. Goldin, and M. McNitt-Gray, "The effect of radiation dose reduction on computer-aided detection performance in a low-dose lung cancer screening population," Med. Phys., vol. 44, 2017.

[2] A. Klockner, "PyCUDA." [Online]. Available: https://mathema.tician.de/software/pycuda/.

[3] J. Hoffman, S. Young, F. Noo, and M. McNitt-Gray, "Technical Note : FreeCT _ wFBP : A robust , efficient , open-source implementation of weighted filtered backprojection for helical , fan-beam CT," Med. Phys., vol. 43, no. 3, p. 10 pp., 2016.