Technical Note: Design and implementation of a high-throughput pipeline for reconstruction and quantitative analysis of CT image data

John Hoffman[a)]
Department of Radiological Sciences, David Geffen School of Medicine at UCLA, Los Angeles, CA 90024, USA

Nastaran Emaminejad and Muhammad Wahi-Anwar
Physics and Biology in Medicine Graduate Program, David Geffen School of Medicine at UCLA, Los Angeles, CA 90024, USA

Grace H. Kim and Matthew Brown
Department of Radiological Sciences, David Geffen School of Medicine at UCLA, Los Angeles, CA 90024, USA
Physics and Biology in Medicine Graduate Program, David Geffen School of Medicine at UCLA, Los Angeles, CA 90024, USA

Stefano Young
Department of Radiological Sciences, David Geffen School of Medicine at UCLA, Los Angeles, CA 90024, USA

Michael McNitt-Gray
Department of Radiological Sciences, David Geffen School of Medicine at UCLA, Los Angeles, CA 90024, USA
Physics and Biology in Medicine Graduate Program, David Geffen School of Medicine at UCLA, Los Angeles, CA 90024, USA

Purpose: With recent substantial improvements in modern computing, interest in quantitative imaging with CT has seen a dramatic increase. As a result, the need to both create and analyze large, high-quality datasets of clinical studies has increased as well. At present, no efficient, widely available method exists to accomplish this. The purpose of this technical note is to describe an open-source high-throughput computational pipeline framework for the reconstruction and analysis of diagnostic CT imaging data to conduct large-scale quantitative imaging studies and to accelerate and improve quantitative imaging research.

Methods: The pipeline consists of two, primary "blocks": reconstruction and analysis. Reconstruction is carried out via a graphics processing unit (GPU) queuing framework developed specifically for the pipeline that allows a dataset to be reconstructed using a variety of different parameter configurations such as slice thickness, reconstruction kernel, and simulated acquisition dose. The analysis portion then automatically analyzes the output of the reconstruction using "modules" that can be combined in various ways to conduct different experiments. Acceleration of analysis is achieved using cluster processing. Efficiency and performance of the pipeline are demonstrated using an example 142 subject lung screening cohort reconstructed 36 different ways and analyzed using quantitative emphysema scoring techniques.

Results: The pipeline reconstructed and analyzed the 5112 reconstructed datasets in approximately 10 days, a roughly 72× speedup over previous efforts using the scanner for reconstructions. Tightly coupled pipeline quality assurance software ensured proper performance of analysis modules with regard to segmentation and emphysema scoring.

Conclusions: The pipeline greatly reduced the time from experiment conception to quantitative results. The modular design of the pipeline allows the high-throughput framework to be utilized for other future experiments into different quantitative imaging techniques. Future applications of the pipeline being explored are robustness testing of quantitative imaging metrics, data generation for deep learning, and use as a test platform for image-processing techniques to improve clinical quantitative imaging. © *2019 American Association of Physicists in Medicine* [https://doi.org/10.1002/mp.13401]

1. INTRODUCTION

Quantitative imaging studies often require large amounts of clinical imaging data. While modern machine learning methods (such as training neural networks) are typically the most notable in terms of data requirements, other applications, and studies also would benefit from broader access to custom datasets. Applications include investigations into robustness of quantitative imaging approaches, image-processing algorithm development and testing, and development of new quantitative imaging techniques such as computer aided diagnosis/detection (CAD).

At present, access to these high-quality, large datasets is somewhat limited. Publicly available datasets such as LIDC,[1] NLST,[2] Maastro NSCLC,[3] etc., represent one means to access large scale datasets, however are limited by the number

of reconstructions available per patient (typically one), the reconstructions are selected (optimized for human readers, not necessarily computer vision or algorithms), and are typically heterogeneous in terms of scanner and protocols used to acquire the data. While this is a good representation of clinical variability, it is challenging if not impossible to achieve a highly controlled dataset when reconstruction parameters are to be varied and investigated. Finally, large scale public datasets quickly go out-of-date due to the turnover and update cycle of CT imaging systems. Another approach is retrospective assembling of datasets from PACS. This allows for more control over the reconstructions acquired, however is still limited to reconstructions optimized for/selected by human readers, and requires substantial time and effort to assemble datasets large enough for some studies.

A promising approach pursued in our work has been the collection and storage of raw projection data from CT scanners, and then subsequently returning to the scanner at a later date to perform reconstructions. This has been employed in Young et al.[4,5] to great effect, and furthermore this allows for preprocessing of the raw projection data, such as simulated noise addition/dose reduction and projection domain denoising, opening up new research pathways not previously possible.

However, the workflow of collecting, processing, and returning raw data to the scanner (illustrated in Fig. 1) is not without substantial logistical limitations as well. Raw projection data must be re-imported to the scanner, the scanner reconstructions cannot be operated in "batch-mode," and reconstructions different than the clinical protocols often cannot be preprogrammed, and must be manually configured via a graphical user interface. Finally, all image data must be exported and returned to the lab site, uploaded to a secure network share and organized for storage and future use. When cohort sizes are small and only a few reconstructions per subject required, this is a viable approach, however it quickly becomes burdensome. For example, to assemble the dataset used in Young et al.[5] required 6 months (three reconstructions per patient for 481 patients). Increasing the number of parameters investigated, or investigating an additional source of variation (e.g., acquisition dose, reconstruction kernel, or slice thickness) dramatically increases the time and labor required to generate datasets used in QI analysis. If timely, large scale quantitative imaging research is to be conducted, an improved approach is required.

Since there is significant interest in performing quantitative imaging work, it would be highly valuable develop a high-throughput system of reconstructing raw projection data
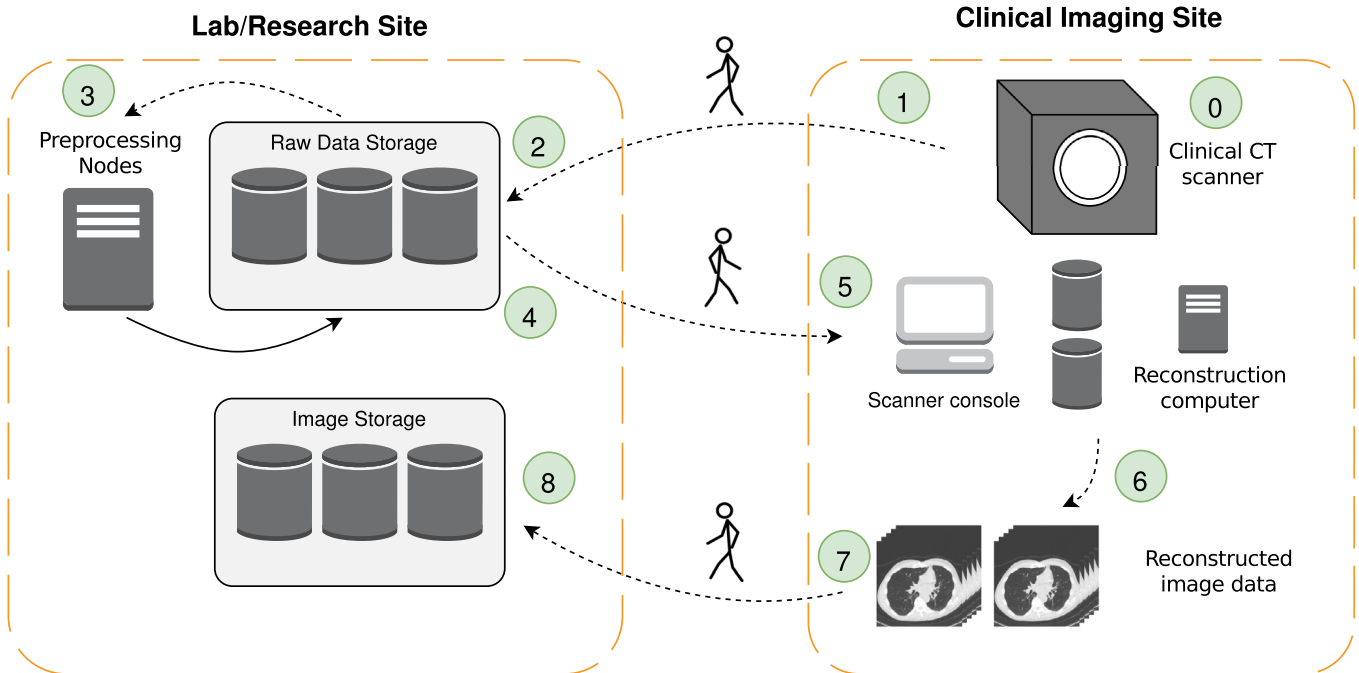
Scanner-based Reconstruction Workflow

FIG. 1. Illustration of workflow using clinical scanner reconstruction. (0) A clinical study is performed and the raw projection data are temporarily stored in the scanner database. (1) Raw projection data are exported to an encrypted hard drive, returned to the lab and then (2) loaded into a secure raw data storage. If preprocessing (3), such as reduced dose acquisition simulation is desired raw data are copied to a network node, preprocessing is performed, and the modified raw file is pushed back to network storage. When data are ready to be reconstructed for an experiment it is (4) loaded back onto an encrypted external hard drive, carried back to the scanner, and (5) uploaded back into the scanner database. (6) Reconstructions are manually configured and performed across all cases and desired reconstruction parameters. (7) All reconstruction image data are exported from the scanner back onto the encrypted hard drive, raw data are deleted from the scanner, and finally, image data are returned to the lab-based network storage location for reconstructed image data. Significant human intervention is required at each step of the seven-part workflow, indicated with dashed arrows. [Color figure can be viewed at wileyonlinelibrary.com]

to create the large datasets required to represent the desired range of acquisition and reconstruction parameters necessary for such studies. Such a system would decrease the time and labor required to perform such studies, as well as achieve a system of organization allowing multiple QI metrics to be tested on the same datasets. By developing such a system, more parameter configurations can be investigated, and larger cohorts can be tested providing more thorough, statistically powerful studies in less time and requiring less researcher labor.

In this work, we develop and detail the construction of such an automated system (referred to as "the pipeline"). It will be shown that the developed pipeline achieves the following: (a) allows for a wide range of acquisition and reconstruction parameters to be configured and applied to raw CT projection data, (b) performs the high-throughput reconstruction of large data sets, (c) automatically organizes the resulting reconstructed volumes for archiving and QI analysis, (d) allows for highly configurable post-processing and analysis to be applied to the reconstructions, (e) produces QI results in a manner to facilitate easy, rapid statistical analysis, and finally (f) functions as an automated tool that requires minimal human intervention after initial configuration. To illustrate the utility and performance of the pipeline in the setting of quantitative imaging, a cohort (N = 142) of low dose lung cancer screening exams with a wide range of acquisition and reconstruction conditions (36 combinations of slice thickness, reconstruction kernel and simulated acquisition dose) was created for analysis in another manuscript; the focus of this manuscript is on the technical developments and

computational performance of the pipeline while the focus on the future manuscript will be the analysis results of the image datasets across the acquisition/reconstruction parameter space.

2. BACKGROUND

It was identified in previous studies[4–6] that the lack of automation in the clinical reconstruction process was a substantial limitation building datasets for quantitative imaging. Although this is often a highly repetitive task when employed in research (i.e., configuring the same set of reconstruction parameters across large cohorts of patients) very suitable for automation, clinical systems are engineered for manual configuration to fit in better with the clinical workflow. While, some manufacturers offer proprietary "recon boxes," that provide some measure of offline reconstruction and automation, these are not widely available, difficult to acquire, and not fully customizable for a researcher's needs. Additionally, they are likely to be tied to proprietary data formats which prevent the possibility of simulation studies, which are often of interest to the reconstruction and quantitative imaging community. In a previous technical note, free, open-source weighted filtered backprojection reconstruction software for diagnostic CT was released,[7] which allows the offline reconstruction of clinical datasets, effectively enabling researchers to build their own "recon boxes" using a published, clinically similar reconstruction method.

The workflow illustrated in Fig. 2, shows the simplification possible with pervasive automation. The user is only
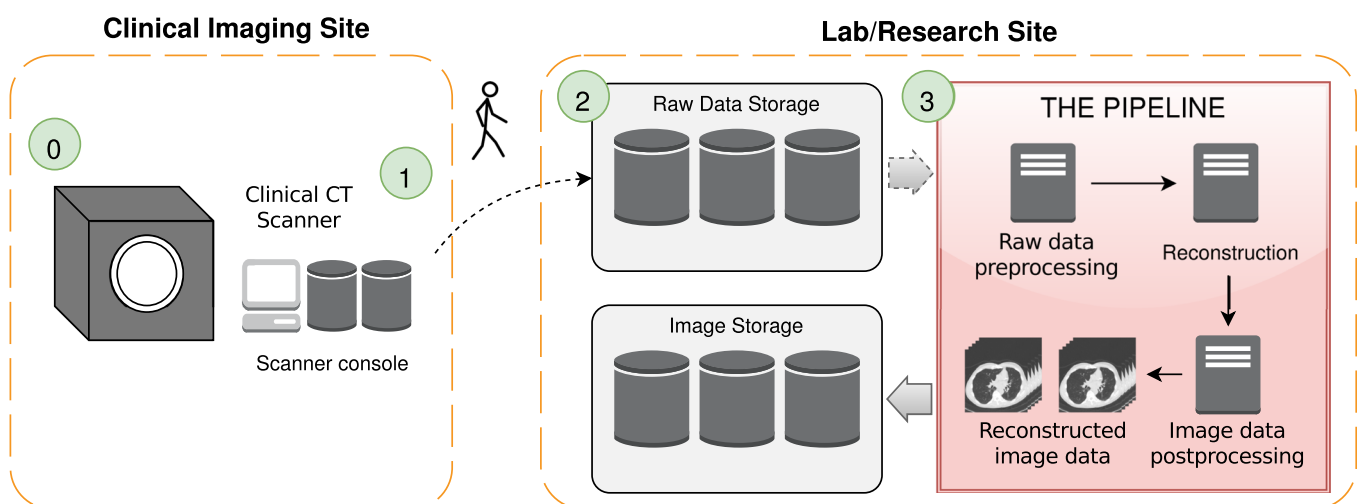
Pipeline Reconstruction Workflow

FIG. 2. Illustration of the reconstruction workflow utilizing the pipeline. (0) Clinical scans are acquired and raw projection data are temporarily stored in the scanner database. (1) Raw projection data are identified and exported to an encrypted hard drive, returned to the lab, and (2) uploaded to network-based network storage for raw data. When data are ready to be reconstructed, (3) the pipeline is set up with one configuration file or via the GUI interface. The pipeline then manages all data-fetching, preprocessing, reconstruction, and uploading to the network-based image storage. No human interaction is required after the pipeline configuration. Image post-processing (prior to storage and subsequent analysis) may include, for example, denoising or other image-domain enhancement technique. [Color figure can be viewed at wileyonlinelibrary.com]
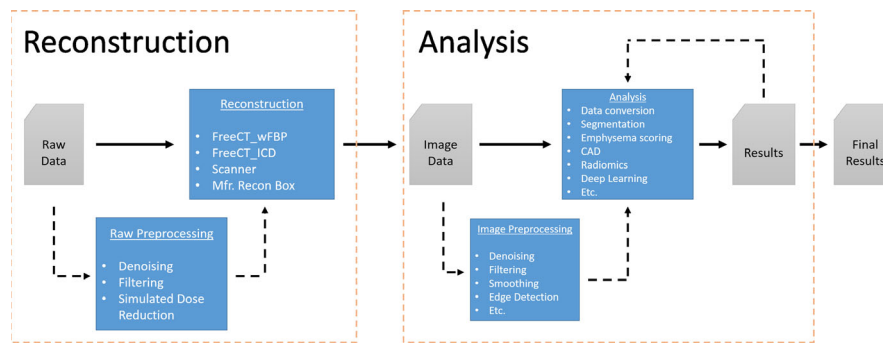
FIG. 3. Block diagram of pipeline workflow from raw projection data to final quantitative imaging data. Each block represents a self-contained task that can be encapsulated in one or more "modules." Dashed lines represent optional processing pathways. Modules can be programmed by the user and incorporated directly into the pipeline automation framework or can exist outside of the high-throughput framework as needed, such as the reconstructions coming from the scanner or some other alternative source. Examples of the types of modules are given in each block. [Color figure can be viewed at wileyonlinelibrary.com]

required to interact with the workflow during the export/import of raw data and the configuration of the pipeline, detailed in Section 3.A. While this automation is largely made possible via the customizable, offline reconstruction approach offered by FreeCT_wFBP, it is the pipeline (i.e., the automation framework) that leverages the reconstruction software and enables high-throughput studies.

3. PIPELINE DESIGN AND IMPLEMENTATION

The pipeline is a collection of compiled programs and Python scripts designed to carry out reconstruction and quantitative imaging analysis. While the pipeline should be thought of as a generalizable framework for high-throughput imaging work, it has been developed thus far with the specific application of robustness testing of quantitative imaging metrics for diagnostic CT, which involves the evaluation of a quantitative imaging test across a range of different acquisition and reconstruction parameters such as slice thickness, reconstruction kernel, and acquisition dose. This application however gives an excellent example of the different, more general components of the pipeline. For this manuscript, the pipeline has been specifically configured to test the robustness of quantitative emphysema scoring approaches using CT image data.

The pipeline workflow, illustrated in Fig. 3, is roughly the following (a) reading of the raw projection data (b) raw data preprocessing (c) reconstruction (d) image data processing (e) analysis, and (f) final results. Initially raw projection data must be parsed into a format readable by the reconstruction software. At present, the pipeline accepts a binary format as well as an open-format, vendor independent DICOM format.[8] Thus, the pipeline is easily extensible to work on non-standard, proprietary data via the programming of a small raw data reading module that converts into either of these two open formats. After reading the raw data, the desired preprocessing is applied. In the case of robustness testing, a calibrated noise addition module[4,9] is used at this stage to simulate reduced-dose scans, however this could also be other processing steps such as projection-domain filtering, or denoising algorithms.[10,11]

Reconstruction of the raw data is performed next to create the desired image datasets. In the current pipeline, FreeCT_wFBP[7] is utilized to perform the desired reconstructions, which can include variations of reconstructed slice thickness and reconstruction kernel.

It should be noted that the pipeline can accept image data from most sources, including other open-source reconstruction software, and image datasets reconstructed at the clinical scanner. This is made possible via a data conversion module that accepts image data in multiple formats (including DICOM, NIfTI, mhd, binary data, and others) and converts to the format utilized by the analysis modules. This also allows for image-domain processing (e.g., denoising, smoothing, etc.) to be applied after reconstruction if desired. In the case of robustness testing, image denoising algorithms are being tested to assess whether their ability to "stabilize" quantitative measures (i.e., reduce the variation caused by changing reconstruction and acquisition parameters).

Finally, analysis is carried out through a series of modules that perform tasks required to produce the final result. In this work, the analysis performed is emphysema scoring and requires the modules that perform image conversion, segmentation, calculation of a lung histogram, and finally the emphysema scoring and aggregation of final results. This is discussed more in Section 3.D. analysis modules. The pipeline is designed so that each module can be replaced based on the requirements for a given experiment, enabling future experiments to leverage the underlying high-throughput design and framework to automate and accelerate imaging data generation and analysis.

To achieve high-throughput for reconstruction, a custom graphics processing unit (GPU) framework was developed. While Fig. 3 illustrates *what* is happening in each stage of the pipeline, following subsections cover *how* the primary components of the computing framework achieve this. The primary components of the framework are: the "launcher" which starts the pipeline; the "daemon", which dispatches individual jobs and ensures system resources are optimally utilized; and the "queue item" script, which processes an individual reconstruction from start to finish. A schematic
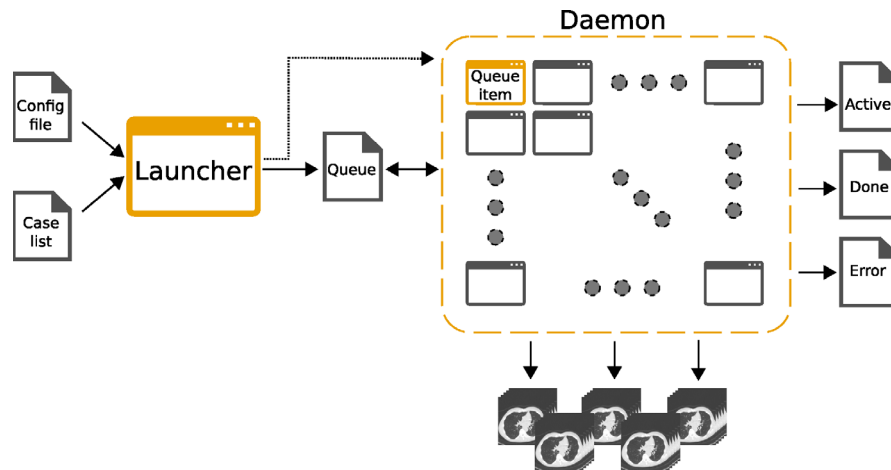
F_IG. 4. Flow diagram of pipeline reconstruction computation. Primary programmatic components are highlighted in orange (i.e., the Launcher, Daemon, and Queue item). Supporting files (e.g., Config file, Case list, Queue, etc.) are not programmatic components themselves, however are critical to the proper functioning of the pipeline. [Color figure can be viewed at wileyonlinelibrary.com]

flowchart view of the components discussed below can be found in Fig. 4.

3.A. Launcher and configuration files

The launcher script is the first of the three primary programmatic components and serves to parse a simple configuration file into a job-list, update the pipeline queue, and subsequently start full pipeline execution managed by the "daemon." After the pipeline launcher has started the daemon, the launcher exits and is not utilized further for a given set of cases.

Configuration files are written in YAML (http://yaml.org), a simple, human-readable "data serialization" format that is well supported across many platforms and programming languages, in particular Python. A sample configuration file is given in Listing 1. Users can request any number of doses, N_{dose}, and any number of slice thicknesses, $N_{s.t.}$, to explore; the pipeline is capable of assessing any number of reconstruction kernels (N_{kern}), however is limited to the offerings of the reconstruction software. The three used in this work are currently the only three offered with FreeCT_wFBP. In the pipeline's present implementation, the total number of reconstructions per patient will thus be

$$N_{dose} * N_{s.t.} * N_{kern}.$$

The launcher script's handling of job list creation, and spawning of all further processes simplifies the role of the researcher in experimental setup which increases throughput and reduces the risk of possible configuration errors.

library:	/data/DefAS_Full/library
case_list:	/data/DefAS_Full/case_list.txt
doses:	[100,50,25,10]
kernels:	[1,2,3]
slice_thicknesses:	[0.6, 1.0, 2.0]

Listing 1: Configuration file use in this experiment for the reconstruction of the cases listed in "case_list.txt." Cases will

be reconstructed with 4 simulated dose levels (100%, 50%, 25%, and 10%), three reconstruction kernels (smooth, medium, and sharp), and three slice thicknesses (0.6 mm, 1.0 mm, and 2.0 mm). The "library" parameter specifies the location where all reconstructed data will be stored, and where subsequent QI analysis will take place.

3.B. Daemon

The daemon is a management script that runs in the background and ensures that system resources are utilized continuously and efficiently. The daemon performs three primary functions: (a) poll the GPU resources of the current system and detect when they are available, (b) spawn "queue_items" which handle the processing of individual reconstructions (see below) when GPU resources are available, and (c) evaluate the exit status of the queue items for logging/debugging purposes.

The daemon runs continuously once the pipeline is launched until there are no more items in the current queue. Checking for available GPUs is done via the polling of a directory of lock files every 5-s. If an available GPU is detected (i.e., no lock file is present) the daemon removes the next item from the job queue, assigns it to the GPU, generates a corresponding lock file, and spawns a new queue item. This ensures that multiple jobs do not compete for the same resources and that all of a system's available GPU resources are used continuously to maximize throughput.

3.C. Queue items

The queue item script handles the processing of an individual reconstruction from start to finish. After receiving its instructions from the daemon in regard to which reconstruction to compute, and which GPU to utilize, the primary steps of this process are data-fetch (i.e., retrieval of raw data from network storage), simulation of reduced dose data if required, and reconstruction according to requested parameters. In
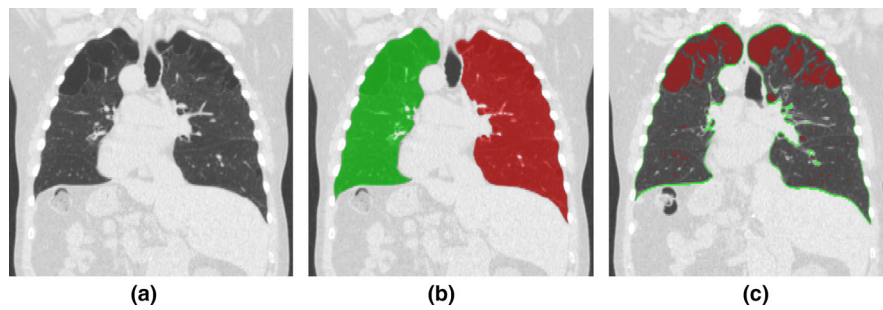
FIG. 5. Sample QA images utilized to verify reconstruction quality and that quantitative tests are being correctly applied. (a) Reconstruction, (b) reconstructions and lung segmentations, (c) quantitative emphysema scoring. [Color figure can be viewed at wileyonlinelibrary.com]

addition to these tasks, the queue item also manages data organization for the given reconstruction, generating a specific directory structure inside of the project library (a directory specified in the configuration file) that prepares the case for quantitative imaging analysis.

3.D. Other pipeline components

3.D.1. Analysis modules

A pipeline analysis module consists of an execution script, designed to process an individual reconstructed image dataset in a pipeline library, and a launch script that dispatches jobs to a network cluster for batch-mode execution of the analysis module for all reconstructions in the library. Because of this design and the fixed library directory structures (see below) of the pipeline, the analysis modules can be readily applied to new pipeline datasets without modification. Thus, a key advantage of the pipeline is the simplicity with which a user can apply quantitative image analysis tools to the large-scale pipeline datasets. An example of this is the automated lung segmentation module since many quantitative tests, such as

global emphysema scoring (performed in this work) or computer automated detection,[10] require the patients' lungs to be segmented. Because this is implemented as a module, the lung segmentation code can be readily utilized for any study. Running the different analysis modules in sequence produces the complete quantitative studies desired. (An example of the modular breakdown of the analysis for this work can be found in Fig. 8).

3.D.2. Quality assurance

A critical challenge of the pipeline is to ensure that reconstructions and analysis tasks were performed correctly. To accomplish this on the large scale required for the pipeline, slice visualizations are automatically generated and presented to researchers in structured HTML documents allowing for the rapid review of the thousands of image volumes generated. Samples visualizations are shown in Fig. 5. A sample HTML quality assurance document is shown in Fig. 6 highlighting the ease with which errors can be detected using this approach. Since the generation of QA documents is a key component of each analysis module, this approach can scale
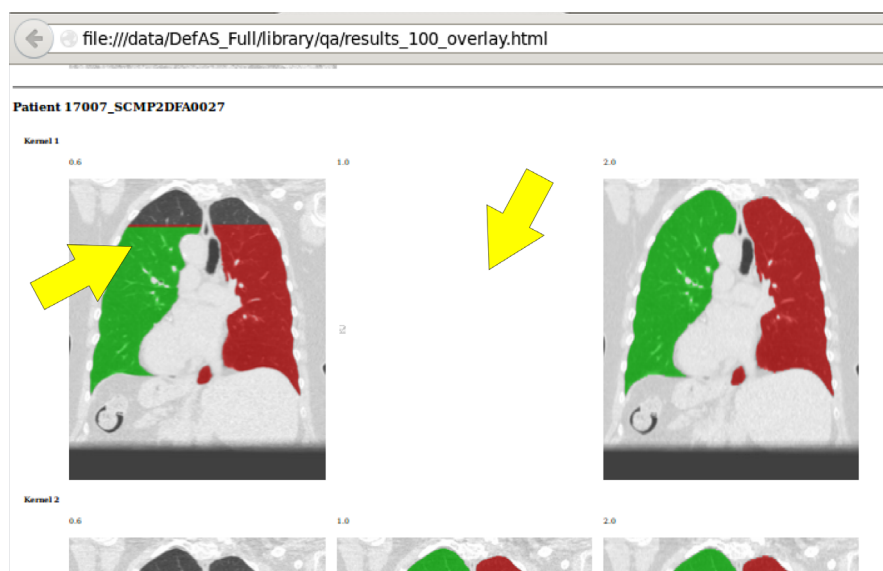

FIG. 6. Sample HTML document, viewable in a standard web browser. Two errors are caught (highlighted with arrows). A faulty segmentation is shown (top of the lungs is truncated in the 0.6 mm slice) and a missing reconstruction or segmentation file is caught via the image missing from the grid. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE I. Parent directory structure for a pipeline library. The top-level directories hold information for the entire library, such as aggregated analysis results and high-level logging information.

Directory element	Purpose
case_list.txt	Stores original file paths to each raw data file used in the current library, and a "hash" value of the raw data file. The hash serves as a unique identifier, and helps to ensure it is not duplicated unnecessarily
Eval/	Directory containing final, aggregated quantitative imaging data ready for statistical analysis and interpretation
Log/	Directory containing copies of all pipeline logging data including the daemon log and logs from individual queue items
Qa/	Directory containing auto-generated structured documents to assist with quality assurance
Recon/	Directory containing all of the reconstructed image data and results from individual queue items (note that there is further directory organization discussed in Table II)
Recons.csv	Contains a record of all reconstructions present in the library including data such as the source raw data file (and its unique hash value), parameter configuration information, and filepath to the image data

TABLE II. Individual reconstruction directory structure (all stored in the "recon" directory from Table I). Directories in this case store data only for one reconstruction. This includes raw quantitative analysis data such as segmentations, and individual test results.

Directory element	Purpose
Eval/	Directory containing "compiled" quantitative imaging data (e.g., complete multi-score results for the emphysema module)
Img/	Directory containing all image data and metadata for the current reconstruction
Log/	Directory containing all logging data for the current reconstruction as well as stdout and stderr output
Qa/	Directory for the storage of reconstruction-specific data used for quality assurance, for instance, a PNG visualization of overlay of the segmentation on top of the reconstruction
Qi_raw/	Directory containing "raw" quantitative imaging results, such as a histogram of voxel values inside of an ROI, computer automated detection reports/results, etc.
Ref/	Directory containing non-pipeline data specific to the reconstruction (i.e., "reference" data). For example, a clinical reconstruction being used for comparison against with the pipeline data; human-reader markings being used for CAD comparison
Seg/	Directory containing and segmentations for the current dataset

with dataset size and allows researchers to easily review and correct problems.

3.D.3. Pipeline automation

Finally, the pipeline is highly automated, however it is intentionally not completely fully automated. While complete automation of all pipeline components, reconstruction, and analysis, is achievable, key breakpoints have been implemented to allow for quality assurance (discussed above) to be performed on the results.

3.D.4. Logging and tracking

Output of each individual process (stdout, stderr, and Python logging) is captured and stored in logfiles. In addition to its utility for debugging, the extensive logging with timestamps allows for the mining of performance data from pipeline runs. This is useful for identifying bottlenecks in processing and further improving pipeline throughput.

3.D.5. Data organization

The pipeline creates a unique directory structure designed specifically for quantitative imaging analysis and stores

imaging data and study metadata directly into the structure for further analysis. The elements of the directory structure are described in Tables I and II. Because the directory structure is standardized across all pipeline runs, post-processing and analysis tasks are simple to apply across all image volumes in a pipeline library, and all analysis data are stored with their respective image data making aggregation for statistical analysis efficient and straightforward. Furthermore, multiple QI analyses can be performed on the same dataset using different analysis modules.

3.D.6. Code availability

The pipeline GPU queuing framework source code as well as the analysis cluster framework code is being made available under the free, open-source GNU General Public License version 3.0 to encourage usage in research and quantitative imaging. Full details can be found on the Github page,[11] however in brief this means that users are free to copy, distribute, and modify the software provided changes are identified and dated in the source code and any modifications are made freely available under the same license. The reconstruction software has also previously been made freely available.[7] At present, the code for the calibrated noise addition and the analysis modules (segmentation, internal data format conversion, etc.) cannot be released due to proprietary code and

research agreements, however free, open-source versions would work within the frameworks being released, and future efforts from our research group may result in the release of some of this code. The pipeline automation infrastructure is not platform specific and should run on any system with Python3 installed, however at this time the pipeline requires at least one CUDA-capable GPU to be installed on the system.

Unfortunately, not all of the modules can be made publicly available at this point (i.e., the noise model, proprietary raw data readers, etc.), however the purpose of this work is to highlight the utility and flexibility of the pipeline in its current form. Additionally, a key intent of making the pipeline open-source and freely available is to enable users to develop their own modules and leverage the pipeline's automation infrastructure to improve experimental throughput. Details of the reconstruction modules, including filtered backprojection (FreeCT_wFBP) and a recently released iterative reconstruction package (FreeCT_ICD) can be found in Refs. [7] and [12], respectively.

4. EXPERIMENTAL METHODS

To illustrate the utility and performance of the pipeline, datasets were created for a project in which lung cancer screening CT datasets representing a wide range of acquisition and reconstruction parameters were created. This dataset used the raw data from 142 subjects to create image datasets that represented four dose levels (original and three simulated reduced dose levels), three reconstruction kernels and three slice thicknesses. The experiment was conducted under IRB approval and raw projection data were separated from any PHI.

4.A. Dose reduction and reconstruction

To explore robustness of emphysema scoring to protocol variation, a range of parameters were selected to capture the possible variability one might see clinically, and additionally some configurations that would push the limits of study "acceptability" for diagnosis, in particular with respect to dose reduction. For this study, four doses were explored: 100%, 50%, 25%, and 10% of the original screening studies (approximately 2.0, 1.0, 0.5, and 0.1 mGy CTDIvol), three reconstructions kernels: smooth, medium, and sharp (corresponding roughly to Siemens B10f/B20f, B40f/B50f, and B60f, respectively), and three slice thicknesses: 0.6, 1.0, and 2.0 mm. Thus, each study was evaluated under 36 different parameter configurations and samples of each of these parameter configurations are shown in Fig. 7.

Simulated dose reduction was performed on the raw data with a noise model,[9] an implementation of which has been validated and utilized for similar, previous experiments.[4,5] The model adds noise to individual projections considering quantum and electronic noise. Electronic noise is an important consideration since the starting dose of CT lung cancer screening is already low; samples of electronic noise were acquired directly from the scanner on which all patients were scanned. Additionally, a realistic attenuation model of the bowtie was generated using measurements from the scanner. For the pipeline, a GPU implementation of the noise model has been developed that achieves an acceleration of roughly $12\times$.

All reconstructions were performed using the open-source, free software FreeCT_wFBP, designed to be similar to the clinical weighted filtered backprojection algorithms utilized on Siemens scanners. While not precisely the same algorithm, when applied to raw data from the scanner utilized, it has been shown to meet the criteria specified by the ACR CT accreditation protocol,[13] and produce clinically similar reconstructions.[7]

4.B. Analysis

Threshold- and histogram-based emphysema scoring was chosen as an example task on which to test the pipeline because they are established approaches and have been the
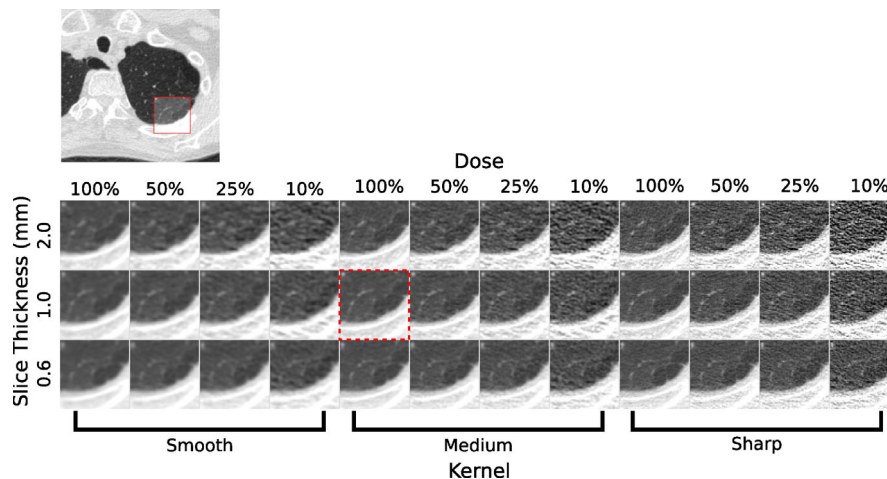

FIG. 7. Sample reconstructions of an ROI in the lungs across the parameters utilized for this experiment. ROI includes a small pocket of emphysema (right side, against lung wall). The appearance and contrast of the emphysema pocket relative to the lung parenchyma changes substantially with parameter selection. The scan most similar to what is performed clinically at our institution is highlighted with a red, dashed rectangle. [Color figure can be viewed at wileyonlinelibrary.com]
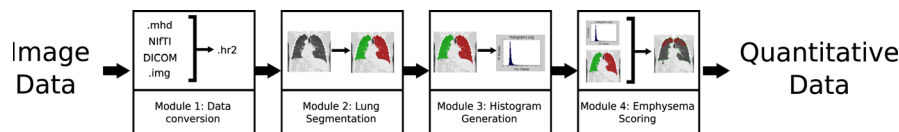
FIG. 8. Analysis modules used to generate quantitative results for emphysema scoring approaches assessed in this study. Module 1 converts from standard image output file types to the file type used by the other analysis modules. Module 2 accepts converted image data and performs automated segmentation of the lungs. Module 3 then leverages the output of the segmentation module as well as the image data to create a histogram of the lungs. Finally, module 4 utilizes the results of all previous three modules to evaluate the different emphysema scores for this experiment. Each analysis module is designed in such a way that it can be used for future experiments. [Color figure can be viewed at wileyonlinelibrary.com]

subject of much research to date.[14–19] Four analysis modules were utilized to carry out the analysis: (a) data format conversion (b) lung segmentation (c) calculation of the lung histogram, and (d) emphysema scoring (shown in Fig. 8). The data format conversion reads the reconstructed image data and image metadata and converts it to a custom format suitable for use with the automated segmentation tool. The lung segmentation module reads the converted image data and runs previously published, fully automated lung segmentation software.[20] The histogram calculation module then utilizes both the converted image data and the generated segmentation to create a histogram of the lung voxels. Finally, the emphysema scoring module pulls on all of the previously generated data to achieve final scoring values for the various executed tests. The metrics calculated for this experiment were density mask metrics,[21] calculated by evaluating the number of voxels in the lung below the given threshold (e.g., -950 HU), and percentile metrics, calculated by evaluating the location of the Nth percentile of the lung parenchyma histogram (most commonly the 15th percentile). Additionally, mean, median, and volume of the lung were computed. Density mask metrics were computed using thresholds from -900 to -970 HU in increments of 10 HU, and 10th, 15th, and 20th percentiles were computed.

4.C. Computing hardware

The reconstruction portion of the pipeline was run on an Alienware Aurora R4 computer with an Intel i7-4960X CPU (3.6 GHz, 15 MB L3 cache), 32 GB of RAM, and two Nvidia GeForce GTX 780 GPU (2304 cores, base clock speed of 863 MHz) with 3.2 GB of global memory each. Analysis of the reconstructed volumes was performed on an in-house computing cluster built with HTCondor cluster software with 15–25 computers in use at a time.

4.D. Performance analysis

Log files generated by the pipeline were mined for timing data using a Python script that is part of the pipeline software package.[11] Start and stop times for each major processing step were recorded for each individual reconstruction (i.e., data-fetch time, simulated dose reduction, and image reconstruction), and elapsed times were computed for both the individual steps as well as the overall execution time for the job queue items (times for all individual steps plus any pipeline overhead). Average times across all 5112 reconstructed image datasets were computed and compared with previous similar experiments conducted by our research group.

5. PERFORMANCE RESULTS

5.A. Reconstruction

Table III summarizes the timing results for the experiment conducted and the average times required for each processing step of the pipeline run.

For the data-generation portion of the pipeline (reduced dose simulation, image reconstruction, and post-reconstruction data handling), the most time-consuming step is reconstruction requiring on average approximately 4.4 min, while simulated dose reduction and data handling requires less than 10 s on average. In general, the GPU implementation of simulated dose reduction requires approximately 1.5 min to process a full case and the data-fetch requires approximately 30 s when these steps are required, however if the required raw data file is already in the library, or the required dose reduction has already been simulated, the pipeline does not re-compute them. Thus, most scans end up not requiring a separate data-fetch or dose reduction step, reducing the average time required for these steps dramatically.

The total size for the reconstructed dataset consisting of 5112 reconstructions and 568 raw data files (142 $\times$ 4 dose levels) required approximately 2 TB of storage. Raw data files were on average approximately 2 GB and reconstructed image data ranged from 200 to 600 MB per reconstruction.

TABLE III. Timing results for pipeline run. While speedup is notable by itself, it is also important to consider that no researcher involvement beyond initial configuration is required during the 8.75 days of run time, while substantial time and attention was required for Young et al. (2017).[5] "Queue item time" considers any computational or data organization overhead, in addition to the time required to perform data-fetch, dose reduction, and reconstruction. Total time is dependent on the system used to run the pipeline. Modern GPUs coupled with a greater number of GPUs in a system will substantially reduce the total run time since the individual reconstructions will run faster, and a greater number of reconstructions will be processed concurrently.

Mean data-fetch time	1.74 s
Mean dose reduction time	8.81 s
Mean reconstruction time	4.40 min
Mean queue item time	5.57 min
Total time, full dataset (2 GPUs)	8.75 days
Approximate speedup over Young et al. (2017)[5]	72$\times$

5.B. Analysis

Analysis required approximately 1 day to complete all steps for the 142 cases reconstructed 36 different ways (a total of 5112 reconstructed image datasets). The most time-consuming part of analysis proved to be the cluster configuration overhead of each node when a job is dispatched. This step is required to ensure that the node has access to the network and remote resources required to run the given analysis step (i.e., mounting network drives, configuring the path to ensure access to pipeline scripts, etc.). This configuration step proved to be a substantial computational burden since it is called for each reconstruction volume and each module (5112 recons × 4 modules = 20 448 times for this experiment). In total, it accounted for nearly 80% of the computing time, or roughly 20 h of the total 24 h required to analyze this dataset. Potential avenues to mitigate this for future experiments are discussed below, however despite the computational demands of the auto-mount script, the capability to analyze 5112 unique reconstructions in approximately 1 day confirms the overall efficiency and strengths of the high-throughput model utilized by the pipeline.

5.C. Quality assurance

Thirty-one reconstructions did not succeed on the first try, and had to be re-queued and re-processed. The failures were likely due to GPU memory conflicts since one of the graphics cards was being used to drive a computer display (which would not occur on a dedicated system); all succeeded on the second try. The pipeline software package provides a script that automatically identifies failed reconstructions and adds them back to the job queue making this "clean up" step simple and fast to perform for the researcher.

Additionally, automated segmentation is known to be imperfect, and the structured QA documents allowed for the fast identification of segmentations with errors. Thirty subjects were identified as having one or more segmentations that needed revision. Most errors occurred on the 0.6 mm slice thickness. Criteria for error identification included substantial airway inclusion in the segmentation, and/or "truncation" of the upper lung which were easily visible in the segmentation QA document. Once the failed cases were identified, segmentations for them were manually edited to correct errors. Quality assurance of all results required less than 1 day.

6. DISCUSSION

In total, the pipeline required slightly over 1 week for data generation and analysis for 142 lung screening patients, assessing 36 unique reconstruction configurations of each scan. The total number of reconstructions analyzed was 5112 and 15 quantitative imaging metrics were computed for each reconstruction (all 15 were histogram-based). The pipeline allowed for this experiment to be conducted approximately

80 times faster, and with substantially less researcher involvement required than the most comparable experiment conducted by our research group, which required approximately 6 months for data generation (i.e., simulated dose reduction and reconstruction) and more for quality assurance and analysis.[5] While a larger cohort was assessed in Ref. [5] (N = 481), only one "dimension" of CT parameter space (i.e., acquisition dose) was assessed, and only three data points per patient were tested (i.e., 100%, 50%, 25% of clinical dose) for a total number of reconstructions of 1443.

The reconstruction portion of the pipeline is highly optimized and performed extremely well. Reconstructions were generated as expected with few failures. Failures were easily identified and reprocessed to complete the dataset which helps make the pipeline be an efficient and robust tool for large-scale quantitative imaging work.

The pipeline is programmed to automatically scale to the system on which it is running. Improved GPU hardware and a greater number of GPUs on the pipeline system will further accelerate the pipeline without any code modifications required. "Deep learning" workstations, such as one recently acquired by our research group, are becoming more common in research groups and typically contain four, state-of-the-art GPUs. Preliminary tests suggest that the new workstation's Nvidia GTX 1080Ti GPUs reconstruct cases 5× faster than the GTX 780 GPUs utilized for this work due to a faster clock speed and a greater number of computing cores. With four GPUs in the new machine, this suggests that all of the data for this work could have been generated in approximately 1 day.

While faster than any previous alternatives, it was observed that the current implementation of the analysis modules could be improved. Namely, the computational overhead required for the cluster node configuration script was a substantial burden requiring more time than the actual quantitative image-processing steps. The simplest potential solution would be to revise the node configuration script and optimize for only the specific resources required for a given experiment (a "general" version giving access to all resources was used in this work). This will be done for the next experiment using the pipeline, however it is not clear that this will yield substantial improvements in performance. Another potential pathway to improve this would be to let a single cluster node process all modules for a given reconstruction. While this is a promising route forward, it is somewhat less easily adapted for general quantitative imaging than the current modular implementation due to the dependence of some processing steps on data from other reconstructions (e.g., in this case, segmentations from the 100% cases were utilized for the reduced dose cases, instead of attempting to segment the low-dose cases). Implementation of modules would be experiment specific, and a parent "analysis" script (analogous to the reconstruction "queue items") would be required. These scripts would also be required to build in protection for race conditions and shared resources to ensure accurate results and reduce node latency.

While the highest throughput is achieved when using the pipeline for both reconstruction and analysis, the pipeline workflow has been constructed to accept image input from multiple sources. In particular, there may be instances where it is beneficial to perform quantitative evaluation of image data from a clinical scanner or other devices (such as a manufacturer "recon box"), which is fully achievable under the data paradigm utilized. Even non-standard or proprietary image formats are usable, however a small data conversion tool or script will likely need to be added to the data conversion module to translate into a format directly usable by the pipeline.

Future improvements to the pipeline are planned, namely a more robust, integrated interface that combines reconstruction process monitoring, quality assurance, and analysis module configuration into one application. New reconstruction algorithms are under development, namely a free, open-source fully model-based iterative reconstruction for clinical diagnostic CT data, and will be added as a configuration option to the pipeline allowed users to select from weighted filtered backprojection or iterative reconstruction[22] and more fully capture the broad variety of clinically realistic reconstructions. Lastly, while the pipeline's current implementation is intended for CT imaging, the GPU job queuing framework developed here is generalizable beyond just CT reconstructions and would be an important tool allowing researchers to leverage multi-GPU workstations and servers in a manner not currently available without custom scripts and substantial programming investment.

There are many possible research applications of the pipeline, namely any application in which a quantitative test or image-processing technique needs to be evaluated across a wide variety of reconstruction, acquisition conditions, or a large cohort of subjects. For example, robustness testing of existing and newly developed metrics is an important application, where users would ensure that their quantitative technique reflects the underlying patient biology and disease rather than just the reconstruction parameters utilized (i.e., if the test is robust, little or no change should be observed in the quantitative output). This was explored in Ref. [23] to evaluate emphysema scoring, as well as the possible benefits of image denoising for emphysema scoring. Another key application of the pipeline is in the generation of data for research into CAD and/or deep learning. In these cases, different reconstructions or simulated acquisition changes can be used to realistically augment training and testing data, and then the analysis portion of the pipeline could be utilized to evaluate the results.

Additional modules will further augment the pipeline's utility. While we have described modules for reconstruction, noise addition, and a limited set of analysis tasks, additional modules could be designed to perform more complex analysis, as well as data augmentation. Some possible examples include model observer studies,[24–26] simulating lesions and other pathology,[27,28] evaluating denoising techniques,[23,29,30] evaluating image normalization approaches,[31,32] and many more. Because of the modular design of the pipeline, many

studies could theoretically benefit from the automation possible, however users will typically observe the most benefit with large-scale studies in which manually running all individual components is not viable.

In order to take full advantage of the pipeline, some prerequisites need to be met. First, users must have access to raw projection data, be it from the clinical scanner or from a simulation. This can be a challenge if physical access to the scanner console is not available, or if manufacturers do not allow the exporting of the projection data. Second, users must be able to decode the raw projection data if it is stored in a proprietary format. This has been recognized as a challenging problem in the diagnostic CT community and effort has been made to overcome this such as the introduction of a DICOM-based vendor-independent CT raw data format.[8] Finally, to achieve the calibrated noise reduction discussed in this work, a "noise model" for each scanner must be developed which involves characterizing the scanner's bowtie filter as well as the detector's electronic noise and gain. This requires physical access to the scanner as well as a means of acquiring a zero tube current scan (i.e., no x-ray production), typically via service mode.

While the pipeline is easy to use and eliminates much of the need for users to write their own automation scripts and software, it is recommended that users have some basic familiarity with Python for any module programming/reprogramming that is desired, and YAML for the writing of configuration files. These languages are widely used and well-supported with substantial online resources. Including new or modified modules into the pipeline is very straightforward and is accomplished by adding or changing one line of code to the pipeline "queue item" script. Custom user-created modules can be implemented using any language and added the same way. Use of the pipeline with additional image pre-processing scripts (i.e., bilateral filtering carried out in MATLAB) was performed in Ref. [23]. Custom modules will be necessary to extend the pipeline's automation framework to different research tasks.

7. CONCLUSION

The pipeline's demonstrated reduction of the time required to go from experiment conception to finalized quantitative results is an important advancement for the future investigation of quantitative imaging. The pipeline represents an impressive amount of computing power, however its most important development are the new studies it can enable that were previously intractable due to logistical overhead of data acquisition, and the new scientific insights possible with such a data paradigm.

This work has shown that the pipeline is a high-throughput, automated framework for the systematic exploration and analysis of quantitative imaging across many CT parameter configurations, and performs well in the context of robustness testing of a histogram-based quantitative imaging metric. It allows for the processing and analysis of large imaging datasets and is particularly powerful when starting from raw CT

data, although the analysis modules can be applied to any image data. The highly organized data output is suitable for rapid processing and customization for future analysis and finally, the pervasive automation present substantially reduces the labor burden to conduct such large-scale studies. Indeed, the pipeline has already found applications in robustness studies, deep-learning training data generation and output evaluation,[29] and simulation data generation for other experiments.[33–35] Future expansions of the reconstruction and analysis modules would make the pipeline suitable to other cutting-edge areas of CT research, such as photon-counting CT or dual-energy CT.

While previous investigations into the impacts of quantitative imaging have been performed[12–17] they have not been comprehensive enough to establish robust confidence for widespread clinical use. One potential application of the pipeline is the exhaustive search of clinical parameters to establish "acceptable" conditions under which a given quantitative test can be performed reliably. From these conditions, future CT protocols can be designed in which clinicians can confidently use quantitative image tests to assess their patients. The pipeline is the first tool of which the authors are aware that allows for the thorough investigation of such interplay between CT imaging and reconstruction parameters and represents an exciting new pathway toward new experiments.

ACKNOWLEDGMENTS

a)Author to whom correspondence should be addressed. Electronic mail: johnmarianhoffman@gmail.com

REFERENCES

1. Armato SG, McLennan G, Bidaut L, et al. The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. *Med Phys*. 2011;38:915–931.
2. Aberle DR, Adams AM, Berg CD, et al. Reduced lung cancer mortality with low-dose computed tomographic screening. *N Engl J Med*. 2011;365:395–409.
3. Aerts HJ, Velazquez ER, Leijenaar RT, et al. Data from NSCLC-radiomics-genomics. *Cancer Imaging Arch*. 2017. Available: https://wiki.cancerimagingarchive.net/display/Public/NSCLC+Radiogenomics.
4. Young S, Kim HJG, Ko MM, Ko WW, Flores C, McNitt-Gray MF. Variability in CT lung-nodule volumetry: effects of dose reduction and reconstruction methods. *Med Phys*. 2015;42:2679–2689.
5. Young S, Lo P, Kim G, et al. The effect of radiation dose reduction on computer-aided detection (CAD) performance in a low-dose lung cancer screening population. *Med Phys*. 2017;44:1337–1346.
6. Lo P, Young S, Kim HJG, Hoffman J, Brown M, McNitt-Gray MF. The effects of CT acquisition and reconstruction conditions on computed texture feature values of lung lesions. In: *AAPM* (Anaheim, CA); 2015.
7. Hoffman J, Young S, Noo F, McNitt-Gray M. Technical note: FreeCT_wFBP: a robust, efficient, open-source implementation of weighted filtered backprojection for helical, fan-beam CT. *Med Phys*. 2016;43:1411–1420.
8. Chen B, Duan X, Yu Z, Leng S, Yu L, McCollough C. Technical note: development and validation of an open data format for CT projection data. *Med Phys*. 2015;42:6964–6972.
9. Zabić S, Wang Q, Morton T, Brown KM. A low dose simulation tool for CT systems with energy integrating detectors. *Med Phys*. 2013;40: 031102.
10. Emaminejad N, Wahi-Anwar M, Hoffman J, et al. WE-G-201-4: evaluation of CAD nodule detection performance in low dose CT lung cancer screening across a range of dose levels, slice thicknesses and reconstruction kernels. In: *58th Annu. Meet. Exhib. AAPM (Denver, Colorado)*; 2017.
11. Hoffman J. *CTBB Pipeline Github*; 2017. Available: https://github.com/captnjohnny1618/CTBB_Pipeline_Package
12. Hoffman JM, Young S, Hsieh SS, Mcnitt-gray M. Technical note: Free-CT_ICD: an open-source implementation of a model-based iterative reconstruction method using coordinate descent optimization for CT imaging investigations. *Med Phys*. 2018;45:3591–3603.
13. ACR. CT Accreditation Phantom Instructions; 2013: 1–14. Available: https://www.acraccreditation.org/-/media/ACRAccreditation/Documents/CT/CT-Accreditation-Testing-Instructions.pdf?la=en
14. Gierada DS, Bierhals AJ, Choong CK, et al. Effects of CT section thickness and reconstruction kernel on emphysema quantification. Relationship to the magnitude of the CT emphysema index. *Acad Radiol*. 2010;17:146–156.
15. Gierada DS, Pilgram TK, Whiting BR, et al. Comparison of standard- and low-radiation-dose CT for quantification of emphysema. *Am J Roentgenol*. 2005;188:42–47.
16. Nishio M, Matsumoto S, Ohno Y, et al. Emphysema quantification by low-dose CT: potential impact of adaptive iterative dose reduction using 3D processing. *Am J Roentgenol*. 2012;199:595–601.
17. Choo JY, Goo JM, Lee CH, Park CM, Park SJ, Shim MS. Quantitative analysis of emphysema and airway measurements according to iterative reconstruction algorithms: comparison of filtered back projection, adaptive statistical iterative reconstruction and model-based iterative reconstruction. *Eur Radiol*. 2014;24:799–806.
18. Boedeker KL, Mcnitt-gray MF, Rogers SR, et al. Emphysema: effect of reconstruction algorithm on CT imaging measures. *Radiology*. 2004; 232:295–301.
19. Mets OM, Willemink MJ, De Kort FPL, et al. The effect of iterative reconstruction on computed tomography assessment of emphysema, air trapping and airway dimensions. *Eur Radiol*. 2012;22:2103–2109.
20. Brown MS, Lo P, Goldin JG, et al. Toward clinically usable CAD for lung cancer screening with computed tomography. *Eur Radiol*. 2014;24:2719–2728.
21. Muller NL, Staples CA, Miller RR, Abboud RT. Density mask. *An objective method to quantitate emphysema using computed tomography*. *Chest*. 1988;94:782–787.
22. Young S, Hoffman JM, Noo F, McNitt-Gray MF. Vendor-independent, model-based iterative reconstruction on a rotating grid with coordinate-descent optimization for ct imaging investigations. In: *Annu. Meet. AAPM (Washington, DC)*; 2016.
23. Hoffman J. Characterizing and minimizing the impacts of diagnostic computed tomography acquisition and reconstruction parameter selection on quantitative emphysema scoring, UCLA; 2018.
24. Barrett HH, Myers KJ, Hoeschen C, Kupinski MA, Little MP. Task-based measures of image quality and their relation to radiation dose and patient risk. *Phys Med Biol*. 2015;60:R1–R75.
25. Wunderlich A, Noo F. Evaluation of the impact of tube current modulation on lesion detectability using model observers. In: *Conf. Proc. IEEE Eng. Med. Biol. Soc*; 2008: 2705–2708.
26. Hoffman JM, Noo F, Young S, Mcnitt-Gray MF. TH-AB-207A-9: tailoring TCM schemes to a task: evaluating the impact of customized TCM profiles on detection of lung nodules in simulated CT lung cancer screening biomedical physics and radiology, David Geffen school of medicine at UCLA, Los Angeles, C. In: *AAPM Annu. Meet*; 2016.

27. Solomon J, Samei E. A generic framework to simulate realistic lung, liver and renal pathologies in CT imaging. *Phys Med Biol*. 2014;59:6637–6657.

28. Pezeshk A, Sahiner B, Zeng R, Wunderlich A, Chen W, Petrick N. Seamless insertion of pulmonary nodules in chest CT images. *IEEE Trans Biomed Eng*. 2015;62:2812–2827.

29. Zhao T, Hoffman J, McNitt-Gray M, Ruan D. Low-dose CT image denoising using an optimized wiener filter in the BM3D algorithm. In: *Annu. Meet. AAPM (Denver, CO)*; 2017.

30. Chen H, Zhang Y, Kalra MK, et al. Low-dose CT with a residual encoder-decoder convolutional neural network (RED-CNN). *IEEE Trans Med Imaging*. 2017;36:2524–2535.

31. Gallardo-Estrella L, Pompe E, De Jong PA, et al. Normalized emphysema scores on low dose CT: validation as an imaging biomarker for mortality. *PLoS ONE*. 2017;12:1–12.

32. Gallardo-Estrella L, Lynch DA, Prokop M, et al. Normalizing computed tomography data reconstructed with different filter kernels: effect on emphysema quantification. *Eur Radiol*. 2016;26:478–486.

33. Hoffman J, Noo F, Young S, McNitt-Gray M. TH-AB-207A-09: tailoring TCM schemes to a task: evaluating the impact of customized TCM profiles on detection of lung nodules in simulated CT lung cancer screening. *Med Phys*. 2016;43:3861.

34. Hoffman J, Noo F, Mcmillan K, Young S, McNitt-Gray M. Assessing nodule detection on lung cancer screening CT: the effects of tube current modulation and model observer selection on detectability maps. In: *Proc. SPIE Med. Imaging*; 2016.

35. Hoffman J, Noo F, McNitt-Gray MF. Influence of tube current modulation on noise statistics of reconstructed images in low-dose lung cancer CT screening. In: *Annu. Meet. AAPM (Denver, CO)*; 2017.

INNOVATING

patient-first

CANCER TREATMENT

PRECISION   MOTION   WORKFLOW

It is not just a tagline. It is who we are, what we do and what makes us different. We have continually introduced important cancer treatment breakthroughs that help our customers improve outcomes for patients.

PATIENT-FIRST PRECISION

Confidently delivering effective treatments with minimal side effects.

PATIENT-FIRST VERSATILITY

Making personalized treatments practical for every patient.

PATIENT-FIRST EFFICIENCY

Ensuring fewer treatments and a faster return to daily life.

FIND OUT MORE ON THE NEW ACCURAY.COM/CONTACT-INQUIRY

#ACCURAYPATIENTFIRST

ACCURAY®