Title

Technical Note: FreeCT_ICD: An open-source implementation of a model-based iterative reconstruction method using coordinate descent optimization for CT imaging investigations

Journal

Medical Physics, 45(8)

ISSN

0094-2405

Authors

Hoffman, John M
Noo, Frédéric
Young, Stefano
et al.

Publication Date

2018-08-01

DOI

10.1002/mp.13026

Peer reviewed

TURN UNCERTAINTY INTO PEACE OF MIND

Get ready for Beam Commissioning 4.0

Be the first to know about the all-new solution that revolutionizes your Beam Commissioning.

Quality. Performance. Peace of Mind.

Soon to be revealed.

PROTECT + ENHANCE + SAVE LIVES

Technical Note: FreeCT_ICD: An open-source implementation of a model-based iterative reconstruction method using coordinate descent optimization for CT imaging investigations

John M. Hoffman[a)*]
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

Frédéric Noo*
Department of Radiology and Imaging Science, University of Utah, Salt Lake City, UT 84112, USA

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

Scott S. Hsieh and 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

(Received 24 January 2018; revised 21 May 2018; accepted for publication 22 May 2018;
published xx xxxx xxxx)

Purpose: To facilitate investigations into the impacts of acquisition and reconstruction parameters on quantitative imaging, radiomics and CAD using CT imaging, we previously released an open-source implementation of a conventional weighted filtered backprojection reconstruction called FreeCT_wFBP. Our purpose was to extend that work by providing an open-source implementation of a model-based iterative reconstruction method using coordinate descent optimization, called FreeCT_ICD.

Methods: Model-based iterative reconstruction offers the potential for substantial radiation dose reduction, but can impose substantial computational processing and storage requirements. FreeCT_ICD is an open-source implementation of a model-based iterative reconstruction method that provides a reasonable tradeoff between these requirements. This was accomplished by adapting a previously proposed method that allows the system matrix to be stored with a reasonable memory requirement. The method amounts to describing the attenuation coefficient using rotating slices that follow the helical geometry. In the initially proposed version, the rotating slices are themselves described using blobs. We have replaced this description by a unique model that relies on trilinear interpolation together with the principles of Joseph's method. This model offers an improvement in memory requirement while still allowing highly accurate reconstruction for conventional CT geometries. The system matrix is stored column-wise and combined with an iterative coordinate descent (ICD) optimization. The result is FreeCT_ICD, which is a reconstruction program developed on the Linux platform using C++ libraries and released under the open-source GNU GPL v2.0 license. The software is capable of reconstructing raw projection data of helical CT scans. In this work, the software has been described and evaluated by reconstructing datasets exported from a clinical scanner which consisted of an ACR accreditation phantom dataset and a clinical pediatric thoracic scan.

Results: For the ACR phantom, image quality was comparable to clinical reconstructions as well as reconstructions using open-source FreeCT_wFBP software. The pediatric thoracic scan also yielded acceptable results. In addition, we did not observe any deleterious impact in image quality associated with the utilization of rotating slices. These evaluations also demonstrated reasonable tradeoffs in storage requirements and computational demands.

Conclusion: FreeCT_ICD is an open-source implementation of a model-based iterative reconstruction method that extends the capabilities of previously released open-source reconstruction software and provides the ability to perform vendor-independent reconstructions of clinically acquired raw projection data. This implementation represents a reasonable tradeoff between storage and computational requirements and has demonstrated acceptable image quality in both simulated and clinical image datasets. [https://doi.org/10.1002/mp.13026]

Key words: CT reconstruction, free, FreeCT, model-based iterative, software

1. INTRODUCTION

In previous work,[1] we presented a technical note describing a free and open-source implementation of a commonly used reconstruction concept—specifically, weighted filtered back-projection (wFBP)—for third-generation, helical, multi-slice CT. The purpose of that work was to provide a set of tools for offline reconstruction of raw projection data (i.e., sinogram data) for CT imaging research. These tools have enabled the development of reconstruction pipelines that are not dependent on the availability of clinical CT scanners, can be configured to operate in high-throughput batch modes, can incorporate simulated dose reduction techniques[2,3] and therefore can produce a large collection of image datasets that represent a wide range of acquisition and reconstruction settings such as different slice thicknesses and reconstruction kernels used in wFBP. The results of these reconstruction pipelines have contributed to the growing list of investigations evaluating the robustness of quantitative imaging, radiomics, and CAD methods across a range of scanner platforms, acquisition conditions and reconstruction parameters.[2,4–9]

While this previous work was valuable and informative, it was limited to conventional wFBP reconstructions. Modern scanners offer advanced image reconstruction techniques that use some form of statistical or iterative reconstruction. These advanced image reconstruction methods are an important clinical technique for reducing radiation doses in CT, but require further investigation for their effects on quantitative imaging, radiomics, and CAD performance.

This work extends the previous technical note by describing a model-based iterative reconstruction method that offers potential for dose reduction for vendor-independent reconstruction of clinically acquired raw projection data. This method provides a complementary set of tools to the wFBP tool already available and could be incorporated into the reconstruction pipelines described above. While open-source reconstruction packages exist for FBP, to the authors' knowledge there are no packages available that can directly reconstruct clinical datasets using model-based iterative reconstruction methods. The aim of this work was to fill this gap with software that can provide this capability. A complementary initiative is underway to provide the community with freely available raw data from clinical scanners.[10] The primary goal of this work is to introduce the software, provide sample results highlighting the package's ability to reconstruct data acquired from a clinical CT scanner, and provide a high-level evaluation indicating its fitness for use in a research setting. Rigorous evaluation of the core concepts underpinning the reconstruction approach, as well as their impacts on image quality, have been previously performed in other work.[11–15]

Model-based iterative reconstruction offers the potential for substantial radiation dose reduction,[16] but comes with a challenging computational burden. Part of this burden lies in the size of the system matrix, which for a typical CT scan can be 1000 times larger than system memory for a typical desktop computer. Standard approaches avoid storing the system matrix by focusing on the evaluation of matrix-vector products on the fly, but this limits the choice of system matrix (or forward projection model) to that which can be quickly computed. Model-based iterative reconstruction depends on the accuracy of the CT system model; more detailed models of the x ray source or detector responses may lead to improved resolution and image quality. Storing the system matrix offers the potential for modeling these higher order effects, obviating the need to re-calculate them on the fly at each iteration, at high computational expense. Previously published work proposed reconstructing on rotating slices to exploit helical symmetry, enabling the re-use of the system matrix from view to view, which results in practical storage requirements.[12,17,18] In this work, we used this stored-system-matrix approach together with iterative coordinate descent (ICD) optimization. A penalized-least-squares objective function, with a quadratic or edge preserving penalty term as a regularizer, is minimized voxel-by-voxel, sequentially iterating along the axial direction first, followed by the transaxial direction. Eight in-plane (transaxial) neighbors are used for the calculation of the regularizer. Iterations are accelerated with multi-CPU OpenMP libraries.[19,20]

There exist many ways to define the system matrix while exploiting helical symmetry through the use of rotating slices. Xu et al.[12] used a blob representation of the object, which led to matrix sizes of about 27 GB for a clinical reconstruction. Guo and Gao[21] recently described a more memory efficient method that calculates exact intersection lengths between voxels and rays as in Siddon's method[22] rather than using blobs for computing ray paths, which led to matrix sizes on the order of 6.6 GB. It is known, however, that Siddon's method is more prone to generating images with discretization errors compared to models using linear interpolation such as Joseph's method.[13,23,24] In this work, we have used a unique approach that involves trilinear interpolation across rotating slices in combination with the principles of Joseph's method. This approach offers a compromise between the accuracy offered by blobs and the lower memory requirement offered by Siddon's approach. This compromise provides a reasonable trade-off for an implementation of ICD that can be incorporated into a CT image reconstruction pipeline.

The source code is freely available and can be used and extended by the research community. In Section 2, we describe the features of our specific implementation of ICD including some background of the method and specific features we have built into the software such as user configurations. In Section 3, results from clinical reconstructions involving both the ACR CT accreditation phantom and a pediatric chest patient are presented. This includes comparisons between conventional wFBP reconstructions and our iterative reconstruction approach. Section 4 provides further discussion of the method's potential, limitations and planned future work. An Appendix is included in which we provide some of the mathematical details of the unique aspects of our approach. More information, including documentation, detailed licensing, and source code, can be found via FreeCT's website[25] (http://cvib.ucla.edu/freect), or the FreeCT Github page (https://github.com/freect).

2. ALGORITHM DESCRIPTION AND SOFTWARE FEATURES

The FreeCT_ICD software performs statistical iterative reconstruction posed as an optimization problem of the form

$$\hat{x} = \arg\min_{x} \left\{ \frac{1}{2}(y - Ax)^T W(y - Ax) + R(x) \right\}$$

where y is the sinogram raw data, x is the reconstructed image, W is the statistical weights, A is the system matrix, and $R(x)$ is the regularization term.[11] Although typically understood as volumes, x and y are represented in this equation as one-dimensional vectors. The system matrix A relates x and y together and can physically be understood to be forward projection. The diagonal matrix W contains statistical weights that can be chosen as the inverse variance of the measurements or as uniform weights (employed in this work) when the noise level is similar across the measurements. In ICD, the reconstructed image x is optimized by sequentially optimizing one element at a time.

2.A. System matrix definition

As mentioned earlier, our ICD implementation uses a stored system matrix. To make the size of this matrix practical, we have adopted the concept of Xu et al.,[12] which exploits the helical symmetry through the use of rotating slices. Within this concept, there exists many ways to define the system matrix elements. We have used a unique method that can be briefly described as follows. We start as in Ref. [12] by picturing a number of slices along the patient bed direction, with fixed distance from slice to slice that is chosen as a multiple of the distance covered by the source over two successive readings. Next, within each slice, we picture a 2D system of Cartesian coordinates that is centered on the rotation axis with an orientation that changes from slice to slice to follow the helical geometry. Within each slice, we use this system to define a grid of uniformly distributed samples. Then, we decide that the object value at any location can be estimated in three steps: (a) two neighboring slices are identified; (b) bilinear interpolation is performed across the samples within each neighboring slice to obtain an approximate object value above and below the location of interest (in the patient bed direction); (c) the two obtained values are linearly interpolated in the slice direction. Using this approximation concept, any line integral can now be evaluated by summing the trilinearly interpolated object values over the ray path. Such an evaluation may be seen as a straightforward extension of the bilinear method described by Hahn et al.[26] for 2D fan-beam tomography. However, as presented in Ref. [22], the size of the system matrix corresponding to this approach turns out to be much larger than that offered by Joseph's method while yielding very similar image quality. To avoid this disadvantage in our implementation, we decided to further refine our model by employing the steps that enable reducing the bilinear method to Joseph's method in 2D. A complete mathematical description of this unique forward projection model is given in Appendix. Note however that this description does not cover the overarching concept that yields a practical system matrix size through exploitation of the helix geometry with rotating slices; for explanations on this aspect, we refer the reader to Xu et al.[12]

Our system matrix approach provides results that are very similar to those one might expect using Joseph's method in 3D on a conventional Cartesian grid. We have verified this aspect using computer simulation with the FORBILD head phantom (for brevity, this is not reported here). We expect the blob approach of Xu et al.[12] can provide results with fewer discretization errors, particularly in the absence of a regularizer, but this advantage comes with a much higher memory requirement. Compared with the intersection length based approach of Guo and Gao,[21] the opposite effect is expected: fewer discretization errors at the cost of an increase in memory requirement. We did not consider employing a Siddon-based approach as our experience in 2D fan-beam tomography is that Siddon's approach is suboptimal for practical CT geometries. This experience of ours is in agreement with the analysis in Refs. [20] and [21].

2.B. Additional features/key aspects of the program

The program includes a penalty term in the objective function as a regularizer, with two choices for the potential function: quadratic or Fair potential.[15] In the quadratic case, the single coordinate optimization problem is solved analytically; in the Fair potential case, it is solved via the bisection method. The program sequentially iterates along the axial direction first, followed by the transaxial direction, so that the elements of the stored system matrix need only be accessed once per iteration. Eight transaxial neighbors are used to calculate the penalty term. Iterative coordinate descent does not lend itself easily to GPU parallelization, so the system matrix calculations and iterations are performed on a normal desktop CPU architecture. However, individual iterations are accelerated with multicore CPU OpenMP libraries,[19] which produces up to a factor of 5 speed-up.

One key offering of FreeCT_ICD is that it can be initialized from a filtered backprojection reconstruction using FreeCT_wFBP, which dramatically reduces the number of iterations required to achieve a converged solution. As long as the potential is strictly convex and differentiable (like the quadratic or Fair potential currently offered in our implementation), the objective function is strictly convex and differentiable, and therefore admits a unique minimizer. In such a setting, the ICD method is known to converge to this minimizer for any initial input image.[27] To take advantage of wFBP initialization, users will have to install FreeCT_wFBP and have a suitable GPU. For the requirements of FreeCT_wFBP, readers are referred to the FreeCT_wFBP technical note[1] and the FreeCT_wFBP documentation.[25]

2.C. Software features common to FreeCT_wFBP

Several similarities to FreeCT_wFBP are described here. FreeCT_ICD also reconstructs helical data from third-

generation multidetector CT scanners. This software also does not reconstruct axial scans or helical scans with gantry tilt. In the current implementation, there is some flying focal spot capability (in-plane flying focal spot has been implemented and others are under development and should be available in the near future).

Like FreeCT_wFBP, FreeCT_ICD uses raw projection data. The data provided to FreeCT_ICD should be the post-log attenuation data (i.e., not raw count data) with any desired physical corrections applied, since FreeCT_ICD does not perform any projection domain corrections in an effort to remain vendor-independent. While these data are not always readily available from clinical scanners, there are efforts to develop a recently proposed vendor neutral raw data format based on the DICOM standard.[10] The reader library packaged with FreeCT_wFBP is capable of directly reading this open-format DICOM raw data as well as binary files of floating-point data. The reader library for FreeCT_wFBP and FreeCT_ICD ("FreeCT_Reader") does not read directly from vendor-specific proprietary data formats. Exact specifications for how data are read into the software can be found in the documentation,[25] further providing users the opportunity to supply their own raw data through simulation or other means.

```
# ====================================
# Sample parameter file for a FreeCT_ICD reconstruction
# Paths
sinogram_path:./n_ffs_short_acr.bin
output_dir:   ./
output_file:   n_ffs_short_acr.img
...
# Scanner Geometry
acquisition_fov:                50.0
n_channels:                     736
num_views_per_turn_without_ffs: 1152
focal_spot_radius:              59.5
...
# Recon Geometry
recon_fov:        25.0 # Diameter of
nx:               512
...
# Iterative recon parameters
wfbp_initialize:        1
penalty:                edge-preserving
...
num_iterations:         100
# ====================================
```

Listing 1: Excerpts from a configuration file for FreeCT_ICD indicating the types of parameters users must specify for a reconstruction, as well as what can be modified to adapt FreeCT_ICD to other system geometries. Configuration files are written in YAML. The "#" denotes commented lines that are not parsed for configuration; "..." indicates regions that have been removed for brevity.

The software was designed so that it could be implemented into a high-throughput computational pipeline that would be useful for reconstruction of a large number of cases with different reconstruction settings. Therefore, reconstructions with the software can be configured using parameter files such as those described in Ref. [1]. Listing 1 provides an excerpt of one of the configuration files used to initialize FreeCT_ICD for reconstruction. Many of the parameters required are geometry specifications common to third-generation helical CT scanners (e.g., source to isocenter distance, source to detector distance, fan angle, number of detector channels, etc.). Other parameters specify how the scan was acquired such as collimation, pitch, and usage of flying focal spots. Finally, the user must specify "standard" reconstruction specific parameters such as range of locations to reconstruct and reconstruction grid dimensions, as well as iterative-specific parameters such as which penalty to utilize, number of iterations, and whether or not to initialize the reconstruction with a FreeCT_wFBP reconstruction. A full sample input file for the software can be found at https://github.com/FreeCT/FreeCT_ICD/blob/master/resources/sample_prm.yaml.

2.D. Requirements and dependencies

In this subsection, we provide a brief overview of the resources needed to run FreeCT_ICD. For a full description of the hardware and software requirements please refer to the documentation.[25]

2.D.1. Software

FreeCT_ICD was developed on Linux (Ubuntu 14.04LTS, Canonical, Ltd, London, UK) and should compile and run on all modern Linux distributions with little to no modification. Only two major external dependencies for building and running the software are required: (a) the Boost uBLAS C++ library (http://www.boost.org) and (b) the "yaml-cpp" library (https://github.com/jbeder/yaml-cpp). The Boost libraries come preinstalled on most Linux systems and/or are easily available through the distribution's package manager along with the YAML-cpp library. More information on installing these two dependencies can be found in the online documentation.[25]

2.D.2. Hardware

Recommended system specifications for running FreeCT_ICD are: a Linux computer with at least 8 GB of RAM, a 4-core CPU, and at least 10 GB of free hard drive space. Resources in excess of this, in particular additional CPU cores, will help FreeCT_ICD run faster or will allow the user to reconstruct larger datasets.

Because the stored system matrix approach utilized for FreeCT_ICD necessitates substantial data storage, effort has been made to limit FreeCT_ICD's RAM requirements

to run on any system that has at least 8 GB installed. This has been accomplished through the periodic flushing of system matrix data to the hard drive. The amount of hard drive space required for storage of the system matrix is specific to the raw projection data being reconstructed and desired reconstruction (namely the number of pixels in each slice). However, for reference, the matrix sizes for the reconstructions of clinical data shown in this work were ~7–15 GB. Sparse storage approaches are utilized to reduce requirements, which becomes substantially more important as the number of detector rows increases and/or flying focal spots are employed. During reconstruction (as opposed to the matrix generation step), the system matrix is only read one column at a time (~10–100 MB) and thus RAM requirements for the entire program are typically less than 2 GB.

FreeCT_ICD benefits greatly from running on a multicore system but is capable of running on a single core only. The fastest configuration to run FreeCT_ICD is using the maximum number of cores available on the computer, although it should be noted that threading and hyperthreading of cores can cause apparent drops in performance when using values below the maximum. By default, FreeCT_ICD is configured to use all available cores of the system up to a total of 12.

2.E. Licensing

Similar to FreeCT_wFBP, FreeCT_ICD is freely distributed under GNU GPL v2.0 in an effort to encourage further research and education using diagnostic CT. Licensing under the GPL v2.0 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. FreeCT_ICD will be maintaining individual, versioned releases through the website and GitHub, which will provide a consistent "history" needed for reproducibility. For those interested in developing or modifying the software, FreeCT_ICD, along with FreeCT_wFBP, is an ongoing effort open to feedback, suggestions, and contributions from the larger community.

3. SOFTWARE EVALUATION

In this section, we report on reconstructions of datasets acquired on a clinical scanner (Definition AS 64, Siemens Healthineers, Forchheim, Germany) at UCLA that were carried out to evaluate image quality and to demonstrate the software's ability to correctly reconstruct data from a clinical diagnostic CT scanner. Reconstructions are provided for both the phantom from the American College of Radiology (ACR) CT Accreditation Program and a pediatric thoracic scan. In both cases, the raw projection data was collected from the scanner and then image data was reconstructed using FreeCT_ICD. These are described below.

TABLE I. Acquisition and reconstruction parameters for both the ACR phantom scan and the pediatric thoracic scan (including penalty term and edge-preserving parameters).

Scan	ACR phantom	Pediatric chest
Acquisition parameters		
Tube voltage [kV]	120	100
CareDose4D	Off	On
Quality Reference mAs	—	180
Effective mAs	100	73
Collimation	16 × 1.2 mm	16 × 1.2 mm
Pitch	1.0	1.0
Flying focal spot	Off	Off
Rotation time [s]	0.33	0.33
Reconstruction (ICD) parameters	wFBP initialization	wFBP initialization
Voxel grid dimensions	512 × 512 × 132	512 × 512 × 163
Voxel size [mm]	0.58 × 0.58 × 1.5	0.98 × 0.98 × 1.5
FOV radius [mm]	300	500
Edge-preserving parameter	0.005	0.005
Penalty term parameter	0.1	0.1
System matrix size [GB]	8.5	14.6
Iterations	50	50

3.A. ACR CT accreditation phantom

The ACR CT accreditation phantom was scanned on the clinical scanner using a helical scan protocol with acquisition parameters described in Table I. The raw data was captured from the scanner and reconstructed using both wFBP and the ICD algorithm using wFBP initialization. The reconstruction parameters used are also described in Table I. Figures 1 and 2 shows images through the reconstructed ACR phantom from the ICD reconstruction.

Below are the results from the CT number module, the uniformity module, the low-contrast module, and the resolution module.

3.A.1. CT number module results

Using the image shown in Fig. 1, the CT number of all materials was evaluated according to ACR CT Accreditation Program instructions.[28] The results are shown in Table II. All reconstructed CT number values were within the acceptable ranges as defined in the accreditation instructions.

3.A.2. Low-contrast module results

The low-contrast module reconstruction gave a CNR of 3.83, primarily due to a very low standard deviation value of 1.73. For adult abdomen protocols, the accreditation program guidelines specify that the CNR should be >1.0, so this value is acceptable (it should be noted that there is no CNR specification for a routine chest protocol). It should be noted that the CNR is affected by the reconstruction kernel in FBP, and by a number of parameters in the iterative reconstruction

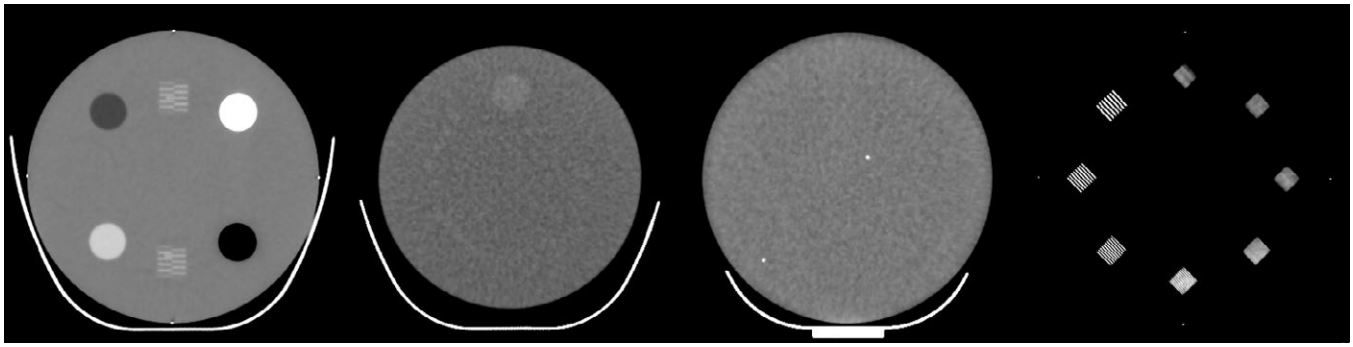


FIG. 1. Images from the FreeCT_ICD reconstructions of the ACR accreditation phantom using the acquisition and reconstruction parameters described in Table I with the edge preserving penalty function. This figure shows from left to right: the CT number module; the low-contrast module; the uniformity module and the spatial resolution module. Each image has been windowed and leveled to the values recommended by the ACR; labels correspond to the module numbers identified by the ACR.[28] Left to right (W/L specified in HU): 400/0, 100/100, 100/0, 100/1050.

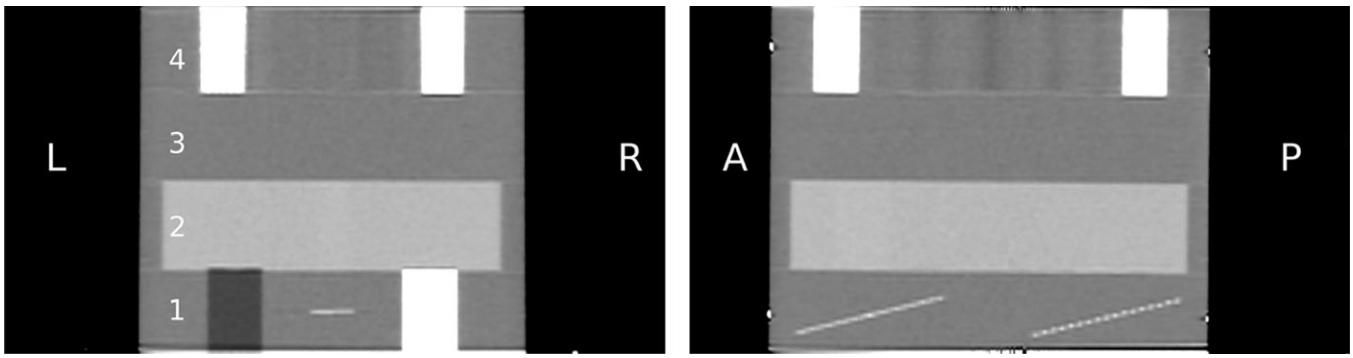


FIG. 2. Coronal (left) and sagittal (right) reformats of the FreeCT_ICD reconstructions of the ACR phantom using the edge preserving penalty function. Module numbers are labeled according to ACR specifications. Shown with window/level of 400/0 HU.

TABLE II. Results from CT number evaluations of the ACR CT accreditation phantom shown in Fig. 1. Equivalent ROI results are provided for the FreeCT_wFBP initializer.

Material	Acceptable range [HU]	Reconstructed value: FreeCT_ICD [HU]	Reconstructed value: FreeCT_WFBP [HU]
Polyethylene	−107 to −84	−89	−89
Bone	850 to 970	864	863
Water	−7 to 7	−2	−3
Acrylic	110 to 135	123	122
Air	−1005 to −970	−988	−987

algorithm. Depending on the selection of these parameters, the apparent CNR can be increased or decreased. The CNR for the FreeCT_wFBP initializer reconstruction was 2.21, demonstrating that FreeCT_ICD improves the CNR relative to the wFBP reconstruction alone. This is in addition to the improved spatial resolution demonstrated in Section 3.A.4.

3.A.3. Uniformity module results

For the uniformity module, the maximum difference from center was 1.1 HU, indicating acceptable uniformity in the reconstructed image. The ACR specifies a range of ±5 HU as acceptable. For the FreeCT_wFBP initializer reconstruction, maximum observed difference from center was 0.9 HU.

3.A.4. Resolution module

For the resolution module, the image in Fig. 1 indicates a resolution of 8 lp/cm was achieved with our reconstruction. The ACR no longer requires this evaluation, but does require resolution evaluation as part of annual QC testing. While there is no limiting resolution value stated for adult head protocols, the adult abdomen protocol limiting resolution is 6 lp/cm. Therefore, this resolution can be judged to be acceptable.

To emphasize the differences between the conventional wFBP reconstruction (which served as the initial condition to the ICD) and the ICD reconstruction, Fig. 3 shows some additional images of module 4 which evaluates spatial resolution. This figure shows the resolution section reconstructed from

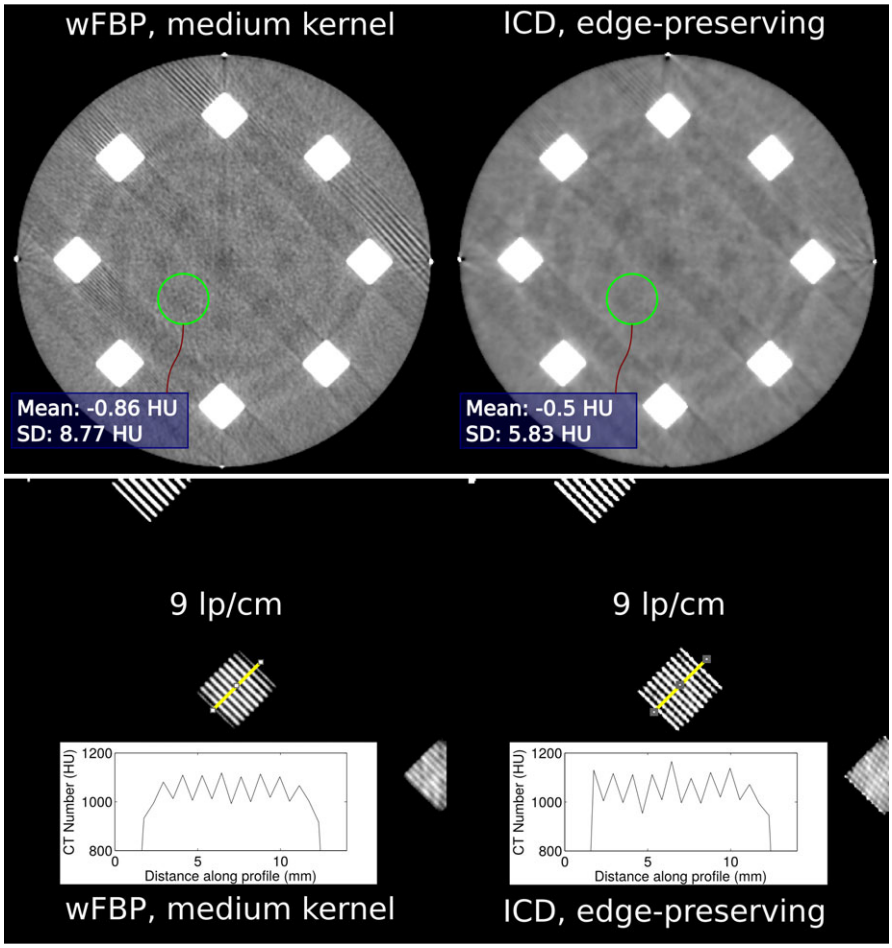


FIG. 3. Module 4 (resolution module) of the ACR phantom reconstructed with both wFBP (medium kernel) and ICD (edge preserving penalty term). The top row displays the images at the Window and Level settings similar to soft tissue (window: 200 HU, level: 0 HU) windows to demonstrate the reduced streak artifacts and reduced image noise resulting from ICD reconstructions. The bottom row displays a zoomed region of the above images with a window of 100 HU and level 1040 HU, and shows the comparable resolution provided by ICD (even at reduced noise level).

both wFBP with a medium filter as well as with the ICD algorithm with the edge-preserving penalty term. These images are shown at both the window and level the ACR recommends for evaluation of the bar patterns (approx. L = 1100/W = 100) as well as a window/level setting that is closer to a soft tissue window, which allows us to evaluate both the noise level (evaluated through standard deviations in each image) as well as the reduced streaking artifacts observed in the ICD image. Thus, these images demonstrate that ICD is indeed providing comparable resolution at reduced noise compared to wFBP, as well as a reduction in the streak artifacts caused by the bar patterns. These images are meant to be illustrative and not definitive of the advantages of ICD over wFBP; it is recognized that the parameters selected (including those for wFBP) will have significant bearing on any comparisons of resolution, noise and image quality in general.

3.B. Clinical dataset—pediatric thoracic scan

A clinically indicated thoracic scan was performed on a pediatric (7-year-old) patient on the same multidetector CT (Definition AS 64, Siemens Healthineers, Forchheim, Germany). The raw projection data was obtained and anonymized under IRB approval at our institution. Our pediatric chest scans are performed with very low doses (CTDIvol for the 32 cm phantom for this scan was 2.5 mGy). Figure 4 represents a coronal image reconstructed from this pediatric thoracic scan using conventional wFBP (with a smooth reconstruction filter) as well as ICD using first a quadratic penalty term and then ICD using an edge preserving penalty term. In both ICD cases, the wFBP was used as the initialization. However, each ICD image can be shown to provide more detail than the wFBP (with smooth kernel) as evidenced by the clearer representation of fine details such as fissures and vascular markings. In this figure, the quadratic penalty term results in noisier images (higher standard deviation) than the edge preserving penalty term, although in general the resolution and noise characteristics depend on the specific parameters used in the regularizer.[15]

3.C. Reconstruction times and initialization with FreeCT_wFBP

Using wFBP as an initial condition is expected to provide robust benefits, especially in terms of reconstruction time

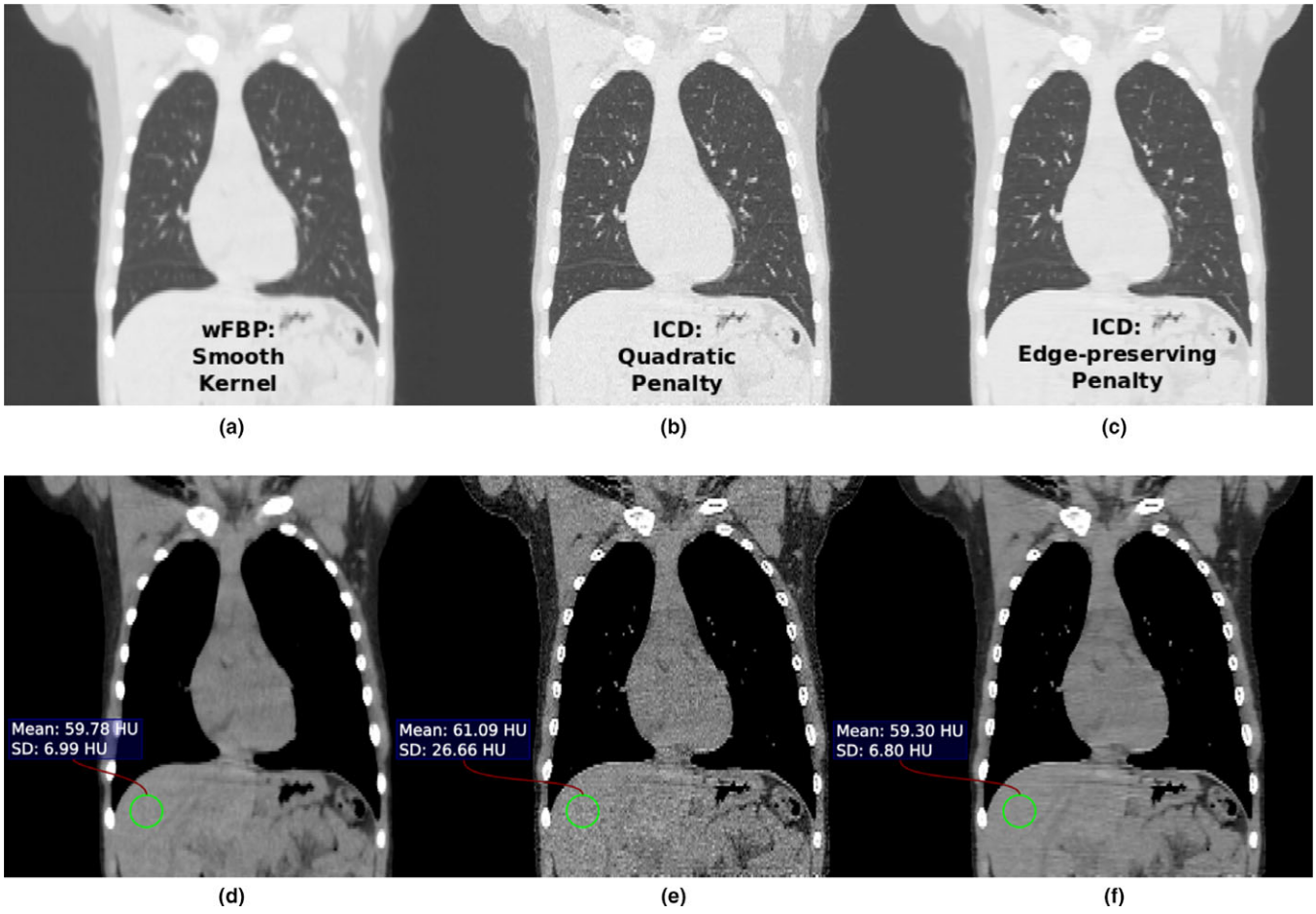


FIG. 4. Coronal reformat image of a pediatric thoracic CT exam from the same raw projection data to illustrate the differences in reconstructions. The top row shows images displayed at lung windows for: (a) wFBP using a smooth reconstruction kernel, (b) ICD using a quadratic penalty term and (c) ICD using an edge preserving penalty term. The bottom row shows the same images but displayed at soft tissue windows and with a region of interest (ROI) within a homogeneous area in the liver which demonstrates the similarity in mean values across reconstructions as well as differences in standard deviation values across reconstructions. The order of images is the same as the row above: (d) wFBP using a smooth reconstruction kernel, (e) ICD using a quadratic penalty term and (f) ICD using an edge preserving penalty term. (a)–(c) are shown with window/level of 1600/600 HU. (d)–(f) are shown with window/level of 400/40 HU.

required for ICD, primarily through the reduction in the number of iterations required to obtain acceptable results. This is illustrated in Fig. 5 which shows the results of reconstructions of the uniformity module of the ACR phantom (Section 3) both without initialization (initial conditions are that all voxels are 0 attenuation) and with initialization using wFBP image of the same section.

Reconstruction times are strongly dependent on both the geometry of the scanner from which the projection data originates (i.e., number of rows, number of channels, usage of flying focal spots, etc.), as well as the geometry of the reconstructed volume (i.e., field of view, number of voxels per slice, etc.). For both the pediatric reconstruction and ACR phantom FreeCT_ICD reconstruction examples shown in this work, reconstructions required approximately 12 hours of compute time on a 6-core CPU workstation. More discussion of run-time and possible strategies to leverage FreeCT_ICD for research are provided in the next section.

4. DISCUSSION

This technical note describes FreeCT_ICD, which is model-based iterative reconstruction software for helical CT images that uses an iterative coordinate descent approach. This method represents a reasonable tradeoff in computation time and memory requirements for practical implementation of offline reconstructions. This tool was designed to facilitate CT imaging research such as investigations into the effects of radiation dose reduction and reconstruction method and parameter selection on CT image quality, quantitative imaging and CAD performance. The offline (i.e., away from the clinical scanner) capabilities provided, coupled with standard representation formats for raw projection (sinogram) data,[10] may provide advantages in terms of the breadth and depth of investigations that can be performed. This tool was intended as a complement to the weighted filtered backprojection tool already developed and made available[1] via the FreeCT website. It is hoped FreeCT_ICD be a useful addition for the

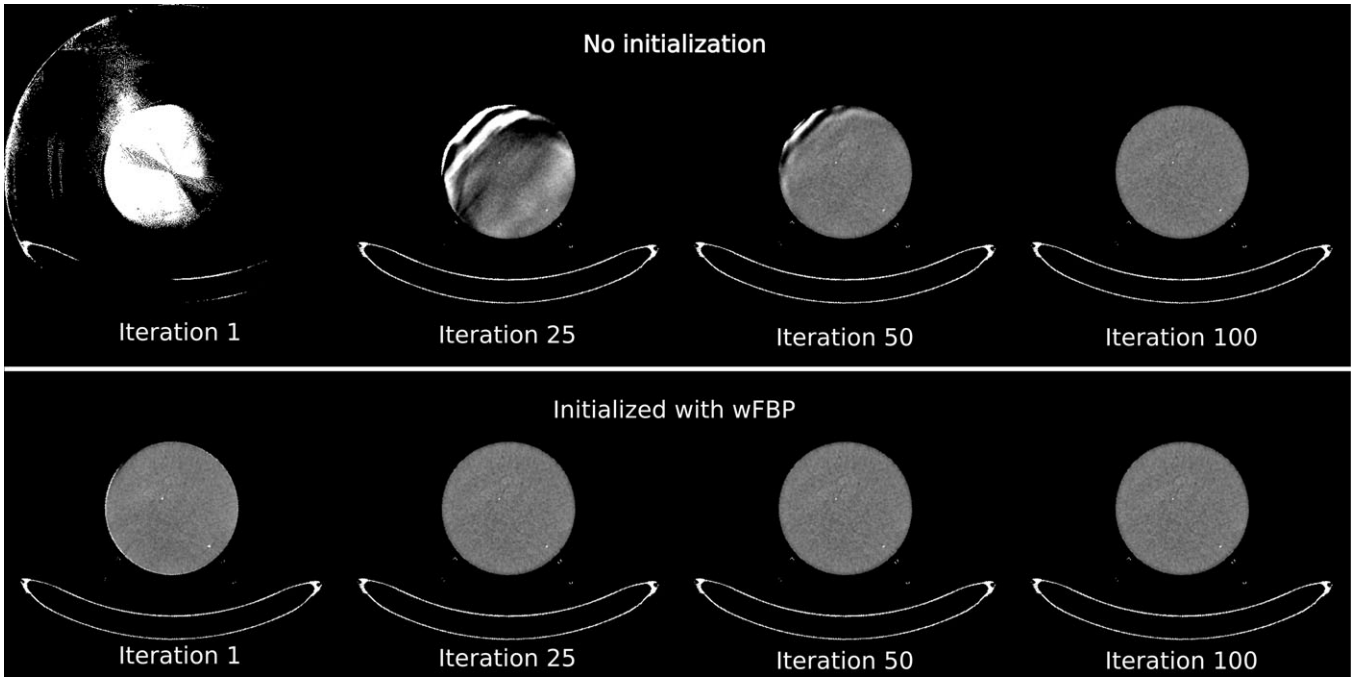


FIG. 5. Figure demonstrating the effects of initialization using wFBP image data using the ACR uniformity module (module 3). The top row shows reconstructions of this module when no initialization (all voxels start as 0 attenuation value) is used. The bottom row shows reconstructions of this module when the wFBP image data is used as the initial condition and how much faster the image converges to the expected answer. All images shown with window/level of 100/0 HU.

medical physics community and the broader research community.

The approach described here is a model-based iterative reconstruction method that incorporates the stored-system-matrix approach together with iterative coordinate descent (ICD) optimization and has been based on a previously proposed method that performs reconstruction on rotating slices to exploit helical symmetry and reduce the storage requirements for the CT system matrix. The approach described incorporates a penalized-least-squares objective function with both a quadratic and edge-preserving penalty term. The forward problem is modeled via a unique approach that combines trilinear interpolation with the principles of Joseph's method to enable accurate reconstruction with low storage requirements. Iterations are accelerated with multi-CPU OpenMP libraries. The result is a reasonable balance between storage requirements and computational performance that leverages this computer architecture.

FreeCT_ICD differs from other iterative reconstruction approaches in that it stores the system matrix directly. To fit within memory, it is necessary to employ a rotating grid. The combination of stored system matrix and rotating grid has also been analyzed and studied in Ref. [12]. Our work differs from that work in that it has been verified with experimental and clinical data, uses ICD for optimization rather than the alternating direction method of multipliers (ADMM), uses a different representation of the system matrix, and will be released as an open-source, free package. In our implementation, we have used a CPU approach to make the software available to a wider community.

The final stored system matrix size is strongly influenced by reconstruction and acquisition parameters (e.g., collimation, reconstructed field of view, pitch, etc.). System matrix sizes for this work fell roughly between 10 and 20 GB using the modified Joseph's method described above. Taking into account the reconstruction and acquisition parameters, our matrix sizes were larger than those achieved by Guo et al.[21] (roughly 1–10 GB, Siddon-based), however, smaller than those achieved by Xu et al.[12] (roughly 27 GB, Blob-based approach). Based on the size of our stored system matrix and the recent analysis provided by Matenine et al.,[29] an efficient GPU refinement of our code may be possible in the near future for high end GPU cards. This warrants further investigation and development.

While it is known that Joseph's method[13] outperforms Siddon's method,[22] it may be possible to use yet more sophisticated projectors with the stored system matrix approach. Distance-driven projectors[30] or separable footprints[31] have the potential to offer more detailed modeling of the system geometry. However, past methods have been limited to what could be practically computed on the fly. With a precomputed matrix, there would be no need to, for example, make the approximation that the projector footprint is separable. Likewise, non-voxel representations could be used[32] without the restriction that the choice of image representation be calculated quickly.

A challenge with a rotating grid is the inclusion of a volumetric regularizer. The images presented in this work used only an in-plane regularizer. Because the voxels in each axial plane are rotated with respect to each other, the implementation of the regularizer becomes more complex and less uniform. A variety of approaches could be considered, but it is not clear which approach is best. For example, the strength of the regularizer could be inversely proportional to the distance

between pixels up to a predetermined maximum distance; or, interpolation of voxels could be used. The full implementation of 3D regularizers is left to future work, however, it is expected to largely involve identifying the equivalent Cartesian points on the rotated slices, and implementing an efficient interpolation scheme. It has been argued[33] that iterative reconstruction derives a great level of benefit from volumetric smoothing, so this choice is important. A separate complication of the rotating grid is that the final dataset must be interpolated back to a fixed grid for display or archival. This requires a rotational transformation of most slices to be consistent with the traditional Cartesian representation. Because much of the ongoing research in iterative reconstruction focuses on regularization/penalty function selection and design, FreeCT_ICD has been coded with a specific file dedicated to penalty functions ("penalties.cpp") making the extension and modification of the current implementation or the addition of new regularizers as straightforward as possible; users simply need to add a new C++ function describing the penalty function desired. More information on penalty functions in FreeCT_ICD can be found in the documentation.[25]

The choice to pursue a stored system matrix implementation was based on our belief in the long-term viability of this approach for detailed, accurate, and efficient reconstruction. We are not claiming in this work that using the stored system matrix approach together with the Joseph-style forward projector is necessarily optimal. The benefits/advantages of using a stored system matrix are likely to be realized when a more detailed modeling of system geometry and physics (focal spot size, detector element size, etc.) will be considered. The issue of more accurate modeling is the subject of further improvements of the software.

Further improvements we are planning for the future are more accurate modeling of the finite focal spot and detector size, acceleration of the reconstruction process, and extension of the regularizer to 3D. In addition, we believe that other popular projectors, such as distance-driven or separable footprints, could be extended into the rotating slice geometry and incorporated into FreeCT_ICD in future updates. We hope that releasing FreeCT_ICD as free, open-source software will lower the barrier to the testing of different and potentially novel projection models with simulated or clinical diagnostic CT data, since the rest of the code infrastructure has already been established.

To provide some preliminary evaluations, images were reconstructed of a standard phantom (ACR CT accreditation phantom) and a clinical case of a pediatric thoracic scan. Image quality was evaluated for each set of images and demonstrated to provide faithful reconstructions of mathematical phantoms as well as images that were comparable to those available from clinical CT scanner reconstructions for the ACR phantom and the pediatric thoracic scan. Images were provided to indicate both the benefits of initializing ICD with wFBP reconstructions as well as to indicate the differences between these two reconstructions.

As mentioned previously, FreeCT_ICD benefits immensely from running on multicore workstations, and can be easily modified to leverage more or less CPU power than the preconfigured usage of up to 12 CPU cores. While the current reconstructions times limit FreeCT_ICD primarily to research purposes, running the software on a computing cluster, via tools such as HTCondor (https://research.cs.wisc.edu/htcondor/) or Amazon Web Services (https://aws.amazon.com/) can achieve many reconstructions in a short period of time, making it viable for investigations requiring an offline, high-throughput alternative to clinical, scanner-based reconstructions. In addition, methods to accelerate the current implementation are under investigation, primarily the porting of certain code portions to GPU and careful memory-access optimizations.

These investigations provide a basis for continuing work including improvements in both computational performance as well as image quality improvement. Specific future developments will include investigations into the utility of extending the regularization into the third (longitudinal or "z") dimension, which may include incorporating a longitudinal direction penalty term as well as ensuring that interpolated values are aligned in the longitudinal direction.

5. CONCLUSION

This technical note describes a software package dedicated to reconstructing helical, third-generation CT data using an iterative model-based reconstruction method. This method is intended to serve as a complement to the previously released FreeCT_wFBP and offers many of the same advantages: a configurable reconstruction software capable of handling various third-generation helical CT scanner geometries via simple modifications to a human-readable configuration file (e.g., Listing 1). Like FreeCT_wFBP, this package does not represent the exact reconstruction algorithms employed by clinical CT scanners, however, an initial assessment demonstrated that FreeCT_ICD does provide acceptable performance on the ACR phantom as well as accurate reconstruction of attenuation values. While FreeCT_ICD does not represent the exact clinical scanner reconstructions, it represents a key iterative reconstruction extension of the FreeCT project, which is in line with the overarching goal of FreeCT to provide reconstruction research tools for clinical diagnostic CT data. FreeCT_ICD's CPU implementation allows for reasonable speed for reconstruction of clinical CT data and is well suited to large-scale explorations of reconstruction parameter space. Since the software is command line based and configured through parameter files, FreeCT_ICD can easily be automated (e.g., via Bash, Python) to perform large numbers of reconstructions without the need for user intervention or extensive access to clinical scanners.

More information, including documentation, detailed licensing, and source code, can be found via FreeCT's website (http://cvib.ucla.edu/freect), or the FreeCT Github page

(https://github.com/freect). The website also hosts contact information for bug reporting, a list of any known bugs, as well as a list of planned updates, and version history.[25]

ACKNOWLEDGMENTS

This work was partially funded by grants from the California Tobacco Related Disease Research Program (No. 22RT-0131) and the NCI's Quantitative Imaging Network (No. U01 CA181156). Dr. Noo also acknowledges some support from the Siemens Master Research Agreement at the University of Utah. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

CONFLICT OF INTEREST

(Hoffman). Imaging Scientist, Part-time, Toshiba Medical Research Institute, USA (Noo): Institutional research agreement, Siemens Healthcare. (Young): None. (Hsieh): None. (McNitt-Gray): Institutional research agreement, Siemens Healthcare.

APPENDIX (1)

DESCRIPTION OF THE FORWARD PROJECTOR

In this appendix, we provide a detailed mathematical description of the unique forward projector model we have used for our ICD implementation. Let $f(\underline{x})$ be the function to be reconstructed with $\underline{x} = (x,y,z)$. Our approach amounts to first approximating $f(\underline{x})$ by a linear combination of basis functions, which may be described as follows. Let

$$b_1(t) = \begin{cases} 1-|t| & \text{if } |t|<1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

be the B-spline of order 1, which is the basis function for linear interpolation in 1D; and let

$$\psi(\underline{x}) = b_1(x/\Delta x)\, b_1(y/\Delta y)\, b_1(z/\Delta z) \tag{2}$$

where Δx, Δy and Δz are sampling steps in x, y and z, respectively. Thus, $\psi(\underline{x})$ is the basis function corresponding to trilinear interpolation in 3D. To create a forward projector with rotational symmetry, we enforce $\Delta x = \Delta y$; and we introduce a rotated version of ψ with the rotation taking place around the z-axis by an angle ϕ, namely

$$\psi_r(\underline{x};\phi) = \psi(\underline{x}\cdot\underline{\phi}, \underline{x}\cdot\underline{\phi}^{\perp}, z) \tag{3}$$

with $\underline{\phi} = (\cos\phi, \sin\phi, 0)$, $\underline{\phi}^{\perp} = (-\sin\phi, \cos\phi, 0)$. Now, consider the following uniformly-distributed samples in x, y and z:

$$\begin{aligned} x_i &= i\,\Delta x,\ i = -N/2,\ldots,N/2 \\ y_j &= j\,\Delta y,\ j = -N/2,\ldots,N/2 \\ z_k &= z_0 + k\,\Delta z,\ k = 0\ldots,N_z-1 \end{aligned} \tag{4}$$

The approximate expansion of $f(\underline{x})$ we seek through iterative reconstruction is

$$f_a(\underline{x}) = \sum_{i,j,k} c_{i,j,k}\, \psi_r(\underline{x} - \underline{x}_{i,j,k}; \phi_k). \tag{5}$$

This expansion corresponds to applying the interpolation process discussed in section 2A. The quantities $c_{i,j,k}$ represent the unknowns, and

$$\underline{x}_{i,j,k} = x_i\,\underline{\phi}_k + y_j\,\underline{\phi}_k^{\perp} + z_k\,\underline{e}_z \tag{6}$$

with $\phi_k = \phi_0 + k\,\Delta\phi$, $k = 0,\ldots,N_z-1$, and $\underline{e}_z = (0,0,1)$. Thus, for each k, the basis functions $\psi_r(\underline{x} - \underline{x}_{i,j,k}; \phi_k)$ are centered on an (x,y) grid that rotates with index k. The angular shift $\Delta\phi$ and the separation between slices Δz are related to the angle $\Delta\lambda$ between the readings by the equations

$$\begin{cases} \Delta z = \dfrac{P}{2\pi}(m\, N_{\text{FFS}}\, \Delta\lambda) \\ \Delta\phi = \dfrac{2\pi}{P}\Delta z \end{cases} \tag{7}$$

where P is the table feed per turn, N_{FFS} is the number of flying focal spot positions, and m is an integer.

Equation (7) for Δz implies that Δz needs to be an integer multiple, m, of $\frac{P}{2\pi}(N_{\text{FFS}}\Delta\lambda)$. In other words, this means that we require that the final reconstructed slice thickness is an integer multiple of the z distance traveled by the x-ray source when moving from one reading to the next reading with the same flying focal spot position (in the specific case of no flying focal spots, this is simply the distance between consecutive source positions; with one flying focal spot direction, e.g., in-plane *or* longitudinal, every other source position; with two flying focal spot directions, e.g. in-plane *and* longitudinal, every fourth). This requirement is essential to create the rotational symmetry leveraged by our system matrix; exploiting this symmetry leads to the substantial size reduction in the system matrix[12].

Because it is typically non-trivial for the user to directly specify such a particular value for Δz, in our program we let the user freely specify an initial Δz (the desired reconstructed slice thickness), then convert this to the nearest Δz value that satisfies the form of Eq. (7) by rounding the potentially non-integer value of m up to the nearest integer value. For example, if a user requests a slice thickness of 1.0 mm and this yields $m = 18.5$, m would be rounded to 19, yielding a final reconstructed slice thickness and spacing of 1.027 mm returned to the user.

At this stage, we are ready to describe the system matrix. Consider the problem of computing the line integral of f along the line of direction $\underline{\alpha}$ through a point $\underline{s}$, i.e.,

$$g(\underline{s},\underline{\alpha}) = \int_{-\infty}^{\infty} f(\underline{s} + t\underline{\alpha})\, dt. \tag{8}$$

We first substitute f_a of Eq. (5) for f in this expression to obtain

$$g(\underline{s},\underline{\alpha}) \simeq \int_{-\infty}^{\infty} f_a(\underline{s}+t\underline{\alpha})\,dt. \tag{9}$$

Next, using the expression of f_a, we get

$$g(\underline{s},\underline{\alpha}) \simeq \sum_{i,j,k} c_{i,j,k} \int_{-\infty}^{\infty} \psi_r(\underline{s}+t\underline{\alpha}-\underline{x}_{i,j,k};\phi_k)\,dt. \tag{10}$$

In this equation, the multiplier of $c_{i,j,k}$ represents one element of the system matrix. By varying $\underline{s}$ and $\underline{\alpha}$ to cover all measured line integrals in the helical geometry, we obtain a set of multipliers that form one column of the system matrix. By enforcing the above relations between $\Delta\phi$, Δz and $\Delta\lambda$, rotational symmetry is induced so that the columns of the system matrix that correspond to the elements defined with $k = 0$ identify all columns of the system matrix.

Although we could exactly compute the multiplier of $c_{i,j,k}$, we found it advantageous to approximate it using the principle of Joseph's method, as explained hereafter. The sought advantage is equivalent to that observed between Joseph's method and the bilinear interpolation method in [31]: a reduction of about 50% in system matrix size with little impact on image quality (this is because each interpolated value in (x,y,z) is obtained from 4 rather than 8 neighbor voxels, which effectively halves the number of elements on each row of the system matrix; see [31] for more details). The approximation is as follows. First, we test whether $\underline{\alpha}$ is closer to $\underline{\phi}_k$ or $\underline{\phi}_k^{\perp}$. The first case occurs when $|\underline{\alpha}\cdot\underline{\phi}_k| > |\underline{\alpha}\cdot\underline{\phi}_k^{\perp}|$. In this case, we perform a change of variable to replace t by a projected coordinate u along $\underline{\phi}_k$. This corresponds to writing

$$\underline{s}+t\,\underline{\alpha}-\underline{x}_{i,j,k} = u\,\underline{\phi}_k + v\,\underline{\phi}_k^{\perp} + w\,\underline{e}_z, \tag{11}$$

which gives

$$\begin{aligned} u &= (\underline{s}+t\,\underline{\alpha})\cdot\underline{\phi}_k - x_i\,,\\ v &= (\underline{s}+t\,\underline{\alpha})\cdot\underline{\phi}_k^{\perp} - y_j,\\ w &= (\underline{s}+t\,\underline{\alpha})\cdot\underline{e}_z - z_k. \end{aligned} \tag{12}$$

From the first equation, we get the link between u and t. Inserting this link in the other two equations, we get v and w as functions of u. The change of variable followed by a simple rectangular-rule approximation of the integral leads to

$$\begin{aligned} \int_{-\infty}^{\infty} \psi_r(\underline{s}+t\underline{\alpha}-\underline{x}_{i,j,k};\phi_k)\,dt \simeq \frac{\Delta t}{|\underline{\alpha}\cdot\underline{\phi}_k|}\,\psi_r(v\,\underline{\phi}_k^{\perp}\\ + w\,\underline{e}_z;\phi_k)\big|_{u=0}. \end{aligned} \tag{13}$$

In the alternative case, when $|\underline{\alpha}\cdot\underline{\phi}_k| < |\underline{\alpha}\cdot\underline{\phi}_k^{\perp}|$, a similar approach is followed using the link between v and t instead of that between u and t.

*These three authors have made equal intellectual contributions to the manuscript and the associated work described here.

[a]Author to whom correspondence should be addressed. Electronic mail: jmhoffman@mednet.ucla.edu.

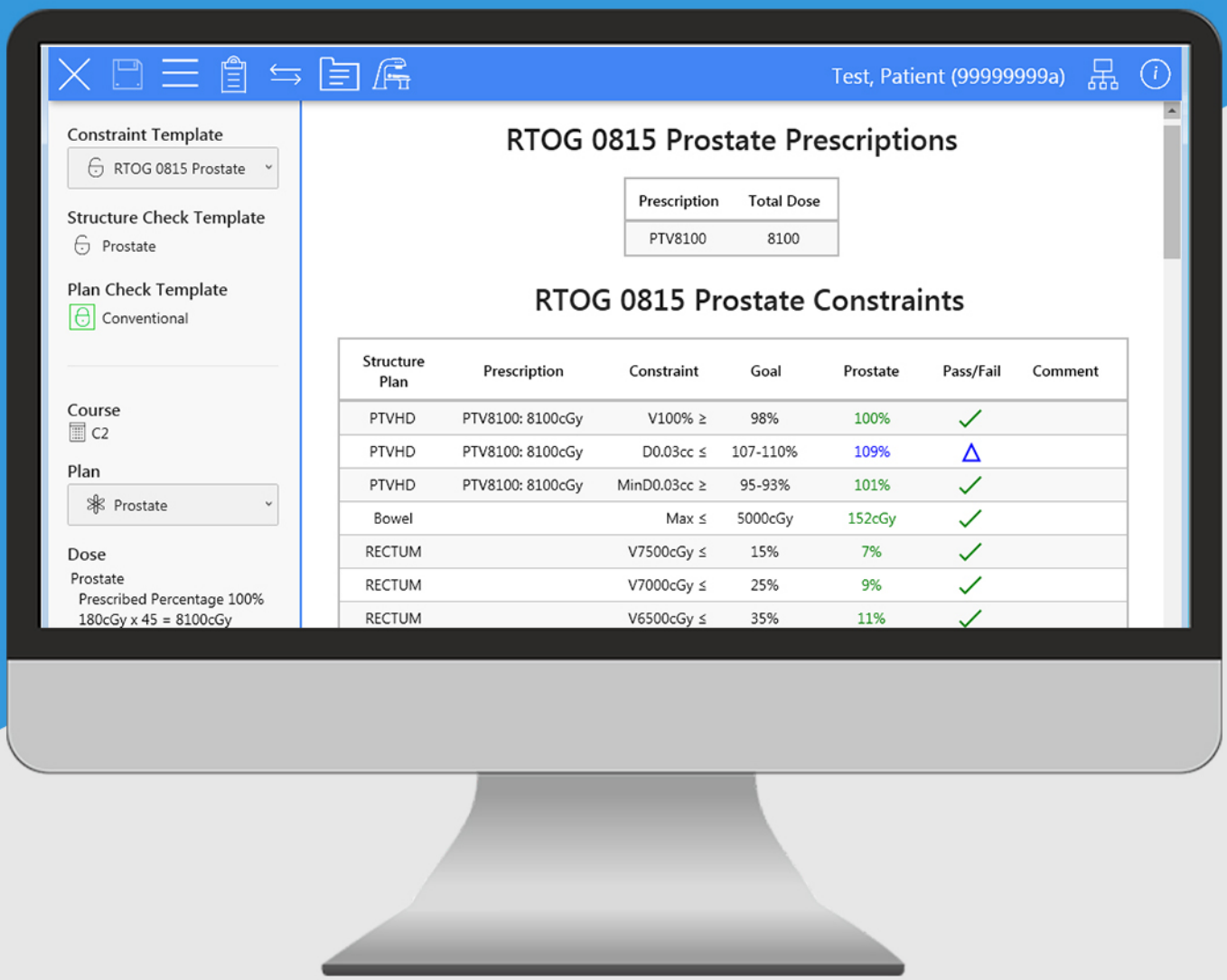
REFERENCES

1. 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.
2. 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.
3. 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.
4. Lo P, Young S, Kim HJ, Brown MS, McNitt MF. Variability in CT lung-nodule quantification: effects of dose reduction and reconstruction methods on density and texture based features. *Med Phys*. 2016;43:4854–4865.
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. Zhao B, Tan Y, Bell DJ, et al. Exploring intra- and inter-reader variability in uni-dimensional, bi-dimensional, and volumetric measurements of solid tumors on CT scans reconstructed at different slice intervals. *Eur J Radiol*. 2013;82:959–968.
7. Zhao B, James LP, Moskowitz CS, et al. Evaluating variability in tumor measurements from same-day repeat CT scans of patients with non-small cell lung cancer. *Radiology*. 2009;252:263–272.
8. Zhao B, Tan Y, Tsai WY, Schwartz LH, Lu L. Exploring variability in CT characterization of tumors: a preliminary phantom study. *Transl. Oncol*. 2014;7:88–93.
9. Zheng Y, Solomon J, Choudhury K, Marin D, Samei E. Accuracy and variability of texture-based radiomics features of lung lesions across CT imaging conditions, in SPIE Med. Imaging (2017), p. 101325F.
10. 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.
11. Thibault J-B, Sauer KD, Bouman CA, Hsieh J. A three-dimensional statistical approach to improved image quality for multislice helical CT. *Med Phys*. 2007;34:4526–4544.
12. Xu J, Tsui BMW. Iterative image reconstruction in helical cone-beam x-ray CT using a stored system matrix approach. *Phys Med Biol*. 2012;57:3477–3497.
13. Joseph PM. An improved algorithm for reprojection rays through pixel images. *IEEE Trans Med Imaging*. 1982;1:192–196.
14. Guo M, Gao H. Memory-efficient algorithm for stored projection and backprojection matrix in helical CT. *Med Phys*. 2017;44:1287–1300.
15. Cho JH, Ramani S, Fessler JA. Motion-compensated image reconstruction with alternating minimization, Proc. 2nd Intl. Mtg. image Form. X-ray CT. 2012;330–333.
16. Pickhardt PJ, Lubner MG, Kim DH, et al. Abdominal CT with model-based iterative reconstruction (MBIR): initial results of a prospective trial comparing ultralow-dose with standard-dose imaging. *AJR Am J Roentgenol*. 2012;199:1266–1274.
17. Steckmann S, Knaup M, Kachelrieß M. Algorithm for hyperfast cone-beam spiral backprojection. *Comput Methods Progr Biomed*. 2010;98:253–260.
18. Steckmann S, Knaup M, Kachelrieß M. High performance cone-beam spiral backprojection with voxel-specific weighting. *Phys Med Biol*. 2009;54:3691–3708.
19. OpenMP Architecture Review Boards. OpenMP API specification for parallel programming. November, 2015. www.openmp.org/specifications (Accessed January 23, 2018).
20. Dagum L, Menon R. OpenMP : an industry-standard API for shared-memory programming. *IEEE Comput. Sci. Eng*. 1998;5:46–55.
21. Guo M, Gao H. Memory-efficient algorithm for stored projection and backprojection matrix in helical CT. *Med Phys*. 2017;44:1287–1300.
22. Siddon RL. Fast calculation of the exact radiological path for a three-dimensional CT array. *Med Phys*. 1985;12:252–255.
23. Zbijewski W, Beekman FJ. Characterization and suppression of edge and aliasing artefacts in iterative x-ray CT reconstruction. *Phys Med Biol*. 2004;49:149–157.
24. Zbijewski W, Beekman FJ. Comparison of methods for suppressing edge and aliasing artefacts in iterative x-ray CT reconstruction. *Phys Med Biol*. 2006;51:1877–1889.

25. UCLA CT Physics and Reconstruction Group. *FreeCT Homepage*. www.cvib.ucla.edu/freect (Accessed January 2018).
26. Hahn K, Schöndube H, Stierstorfer K, Hornegger J, Noo F. A comparison of linear interpolation models for iterative CT reconstruction. *Med Phys*. 2016;43:6455–6473.
27. Sargent RWH, Sebastian DJ. On the convergence of sequential minimization algorithms. *J Optim Theory Appl*. 1973;12:567–575.
28. CT Accreditation Phantom Instructions, American College of Radiology; 2013:1–14. http://www.acr.org/~/media/ACR/Documents/Accreditation/CT/PhantomTestingInstructions.pdf.
29. Matenine D, Côté G, Mascolo-Fortin J, Goussard Y, Després P. System matrix computation vs storage on GPU: a comparative study in cone beam CT. *Med Phys*. 2017;45:579–588.
30. de Man B, Basu S. Distance-driven projection and backprojection in three dimensions. *Phys Med Biol*. 2004;49:2463–2475.
31. Long Y, Fessler J, Balter J. 3D forward and back-projection for X-Ray CT using separable footprints. *IEEE Trans Med Imaging*. 2010;29:1839–1850.
32. Lewitt RM. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Phys Med Biol*. 1992;37:705–716.
33. Hsieh SS, Pelc NJ. Limits to dose reduction from iterative reconstruction and the effect of through-slice blurring. *Proc. SPIE Med. Imaging 2016 Phys. Med Imaging*. 2017;97831C:9783.

ClearCheck

One-click plan evaluation



Dose Constraints ◦ Plan Checks

Structure Checks ◦ Collision Checks

RADformation.com