Learning to Reconstruct Computed Tomography Images Directly From Sinogram Data Under A Variety of Data Acquisition Conditions

Yinsheng Li, Ke Li, Chengzhu Zhang, Juan Montoya, and Guang-Hong Chen

***Abstract*—Computed tomography (CT) is widely used in medical diagnosis and non-destructive detection. Image reconstruction in CT aims to accurately recover pixel values from measured line integrals, i.e., the summed pixel values along straight lines. Provided that the acquired data satisfy the data sufficiency condition as well as other conditions regarding the view angle sampling interval and the severity of transverse data truncation, researchers have discovered many solutions to accurately reconstruct the image. However, if these conditions are violated, accurate image reconstruction from line integrals remains an intellectual challenge. In this paper, a deep learning method with a common network architecture, termed iCT-Net, was developed and trained to accurately reconstruct images for previously solved and unsolved CT reconstruction problems with high quantitative accuracy. Particularly, accurate reconstructions were achieved for the case when the sparse view reconstruction problem (i.e., compressed sensing problem) is entangled with the classical interior tomographic problems.**

***Index Terms*—Image reconstruction, deep learning, sparse-view, interior tomography.**

I. Introduction

THE reconstruction of a function in N-dimensional space from its integral values over a K-dimensional hyperplane $(1 \leq K < N)$ is a central topic in integral geometry [1], [2]. The importance of integral geometry in our daily life can be appreciated by noting that the data acquired in x-ray medical computed tomography (CT) are essentially line integrals through the human body. These line integral data (i.e., integral values for $K = 1$) are acquired at different view angles as the tube-detector assembly rotates from one angular position to another. Image reconstruction from line integrals is also central to other imaging modalities [3], [4] such as Single Photon Emission Computed Tomography (SPECT) and Positron Emission Tomography (PET).

In an ideal scenario, when acquired line integral data can be converted to properly fill the corresponding Fourier space of the image function, the modern filtered back projection (FBP) [5] solution can be readily derived using the inverse Fourier transform, essentially equivalent to the one discovered by Radon [6], [7] in 1917. However, the Fourier transform related FBP reconstruction method is rather restrictive [8]. Due to the quasi-local nature of the information encoding process (i.e., the acquisition of line integral data only involves the function values along a straight line) as well as the use of divergent beam acquisition geometry in CT, there are many other new solutions [9], [10] to exactly reconstruct the image function. Interestingly, these solutions are not mathematically equivalent to one another and these new solutions even enable one to accurately reconstruct a region of interest (ROI) inside the scan field of view (FOV) [11]–[15] with much more relaxed data acquisition conditions, e.g., the super-short scan problem. In this case, it is important to note that there are missing data in Fourier space and thus the Fourier based FBP methods fail to accurately reconstruct the image. Furthermore, if all of the acquired line integral data are potentially truncated, the intrinsic connection with the Fourier transform of the image object completely fails. In this so-called interior problem [3], [4], it has been mathematically proven [16]–[18] that a stable solution does exist under certain conditions, albeit no analytical inversion formula has been discovered yet for this case.

The reconstruction problem with line integral data becomes even more difficult when data acquisition view angles are sparse. Despite the so-called compressed sensing (CS) theory [19], [20] having provided a mathematical foundation to address this sparse view reconstruction problem, when the super-short scan and interior problems in CT encounter sparse view acquisitions, it remains unknown whether it is possible to accurately reconstruct either the entire image or local ROIs within the FOV. Additionally, the inevitable noise contamination in data acquisition further complicates image reconstruction problems from line integral data.

Inspired by the breakthroughs of deep learning [21]–[24] in computer vision and natural language processing, and its success in computer games [25]–[27], physics [28], [29], chemistry [30], and recently tomographic image reconstruction

Manuscript received January 15, 2019; revised March 19, 2019; accepted April 8, 2019. Date of publication April 11, 2019; date of current version October 1, 2019. This work was supported in part by NIH under Grant U01EB021183. *(Corresponding author: Guang-Hong Chen.)*

Y. Li, C. Zhang, and J. Montoya are with the Department of Medical Physics, University of Wisconsin–Madison, Madison, WI 53706 USA.

K. Li and G.-H. Chen are with the Department of Medical Physics, University of Wisconsin–Madison, Madison, WI 53706 USA, and also with the Department of Radiology, University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: gchen7@wisc.edu).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2019.2910760

problems in MRI, CT, and other modalities [31], [32], one may wonder whether deep learning may be employed to not only accurately reconstruct images for those line integral reconstruction problems that have already been solved through human knowledge, but also those that have not yet been solved by human knowledge such as the interior tomographic problem with sparse view angles. In this work, we developed a deep neural network, referred to as intelligent CT network (iCT-Net) and demonstrated that this iCT-Net can be trained to reconstruct images with high quantitative accuracy with either complete or incomplete line integral data including problems that have not been solved or have not been satisfactorily solved by human knowledge.

II. Network Architecture and Training Strategies

A. Deep Learning Neural Network Architecture

When x-ray photons interact with an image object to encode the structural information of that object into measured line integral data, quantum noise caused by the intrinsic photon number fluctuations is inherent in the measured data. Therefore, uncertainty is inevitable in the acquired line integral data in x-ray CT and thus it is natural to use a statistical framework to address the image reconstruction problem. In this framework, an image estimate $\hat{x}$ is defined as the image that maximizes the posteriori conditional probability $P(x|y)$ given the measured line integral data $y \in \mathcal{Y}$, where y denotes the individual line integral datum in sinogram space which is denoted as $\mathcal{Y}$. This is accomplished via the Bayesian inference and solving the optimization problem:

$$\hat{x} = \arg\max P(x|y) = \arg\max P(y|x)P(x) \tag{1}$$

This method requires an explicit assumption about the a priori distribution $P(x)$. In statistical machine learning, instead of using an explicit assumption on the prior $P(x)$, the posterior distribution $P(x|y)$ is directly learned from the training data via a supervised learning process [33]. In this process, a sample x_i is drawn from the output training image data set and a sample y_i is drawn from the input training line integral data set. The data pairs (y_i, x_i) are used to train the iCT-Net in this work, to learn a map $f : \mathcal{Y} \mapsto \mathcal{X}$ ($\mathcal{X}$ denotes image space), i.e., a map directly from sinogram space to image space, such that the learned model distribution, $Q(x|y; f)$, can best approximate the underlying posterior distribution, $P(x|y)$. Once the map $f : \mathcal{Y} \mapsto \mathcal{X}$ is learned, it is applied to predict an image output from the input projection data not used in the training process.

The design of our iCT-Net was inspired by the current FBP based CT imaging pipeline which consists of three major cascaded steps: The first step is to correct measured signals to account for erroneous detector counts caused by a variety of physical reasons such as excessive noise and beam hardening, followed by the second step to filter the corrected data with an apodized ramp filter, and the third step to backproject the filtered data to accomplish the domain transform from line integral space to tomographic image space. In the iCT-Net architecture, multi-channel convolutional neural layers were designed to not only maintain the primary functionality of each of the above three steps in the conventional FBP based CT imaging pipeline, but also to enable iCT-Net to address difficult image reconstruction problems such as view angle truncation, the view angle undersampling, and interior problems using the same architecture. Specifically, the design of our iCT-Net consists of four major cascaded components as shown in Figure 1: (1) Convolutional layers (L1-L5) suppress excessive noise in line integral data and convert a sparse view sinogram into a dense view sinogram. These layers accomplish a manifold learning process, i.e., to learn a noise-reduced and complete data manifold from a noise contaminated and sparse view data manifold. This component is analogous to the signal correction step in the conventional FBP based CT imaging pipeline. (2) Convolutional layers (L6-L9) learn high level feature representations from the output data of the L5 layer. This component is analogous to the filtering step in the conventional FBP based CT imaging pipeline. (3) A fully connected layer L10 performs a domain transform from the extracted data feature space to image space. (4) Layers L11 and L12 learn a combination of the partial image from each view angle to generate a final image. These final two components are analogous to the backprojection and summation steps in the conventional FBP based CT imaging pipeline, but with learnable summation weights to account for potential data redundancy and differences caused by the completely different strategies used in iCT-Net to filter data. Parameters in all layers are directly learned from the input data and training images in the training data set. The iCT-Net architecture enables us to reconstruct images with a 512×512 matrix since the number of parameters is on the order of $O(N^2 \times N_c)$, which is in contrast to $O(N^4)$ in other architectures [32]. Here, N denotes the image matrix size and N_c denotes the number of detector elements.

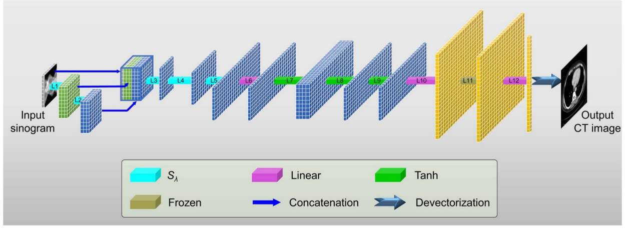


Fig. 1. Architecture of iCT-Net. The proposed deep neural network consists of a total of 12 layers (L1-L12). The L11 layer is a frozen layer, which means that parameters in this layer are not updated in the training process. Both linear and nonlinear activations are used as indicated in the graphics. S_λ is a hard thresholding activation function defined in Eq. (2).

As shown in Figure 1, iCT-Net takes an acquired sinogram with dimensions of $N_c \times N_v$, to generate a CT image with a matrix size of $N \times N$ ($N = 512$), via a twelve-layer deep neural network. Here N_v denotes the number of view angles. Specifics of each of the twelve layers in iCT-Net are described as follows.

L1-L5 are five convolutional layers. L1-L3 operate along the dimension of detector elements while L4 and L5 operate along the dimension of view angles. The L1 layer has 64 convolutional kernels, each with a dimension of $3 \times 1 \times 1$, followed by a hard shrinkage operator (S_λ) as the activation function, which is defined as:

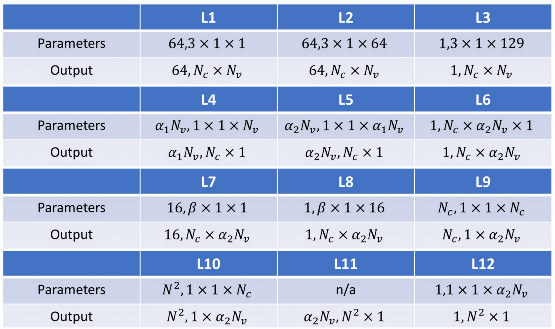
$$S_\lambda(\text{output}) = \begin{cases} \text{output}, & |\text{output}| > \lambda \\ 0, & |\text{output}| \leq \lambda, \end{cases} \tag{2}$$

where λ is the threshold value. The L2 layer has another 64 convolution kernels with a dimension of $3 \times 1 \times 64$, followed by S_λ as the activation. In order to learn new features from the output of the L1 and L2 layers, the original input and the feature outputs of the first two layers were concatenated to form inputs for the L3 layer, the L3 layer has a single channel convolution kernel with a dimension of $3 \times 1 \times 129$, followed by S_λ as the activation. The hyper-parameter λ was empirically selected to be $\lambda = 1 \times 10^{-5}$ for L1-L3 layers. In the L4 layer, there are $\alpha_1\ N_v$ convolutional kernels with the dimension of $1 \times 1 \times N_v$, followed by an activation S_λ. In the L5 layer, there are $\alpha_2\ N_v$ convolutional kernels with the dimension of $1 \times 1 \times \alpha_1\ N_v$, followed by another activation S_λ. A hyperparameter value of $\lambda = 1 \times 10^{-8}$ in L4 and L5 layers and a hyperparameter value of $\alpha_1 = \alpha_2 = 1$ was selected for the dense view reconstruction problem while $\alpha_1 = 2, \alpha_2 = 4$ were empirically selected for the sparse view reconstruction problem with a factor of four view angle undersampling.

L6-L10 are another five convolutional layers. In the L6 layer, there is one kernel with a dimension of $N_c \times \alpha_2\ N_v \times\ 1$, followed by a linear activation. In the L7 layer, there are sixteen kernels with a dimension of $\beta \times 1 \times 1$, followed by a hyperbolic tangent activation, i.e., the operation of the function $tanh(x)$. There is one kernel with dimensions of $\beta \times 1 \times 16$ followed by a hyperbolic tangent activation in the L8 layer. There are N_c kernels with dimensions of $1 \times 1 \times N_c$ followed by a hyperbolic tangent activation in the L9 layer. Finally, there are N^2 kernels with dimensions of $1 \times 1 \times N_c$ followed by a linear activation in the L10 layer. Hyperparameters $N = 512$ and $N_c = 888$ were selected for the non-interior reconstruction problem while $N_c = 222$ was selected for the interior problem with $\varnothing = 12.5$ cm FOV.

Kernels with stride one were used for all convolutional layers. All layers were designed with bias terms except for the L6, L10, and L12 layers. Convolution operations in all convolutional layers were performed with padding to maintain the dimensionality before and after the convolution operations.

L11-L12 layers generate the final image. The dimensions of the output of the L10 layer are $\alpha_2\ N_v \times N^2$. For each of the $\alpha_2\ N_v$ channels, the N^2 values were reshaped into a matrix with a size of $N \times N$. The matrix was then rotated around its center by an increment angle $\phi_i = (\alpha_2\ N_v - i)\Delta\phi, (i = 1, 2, \cdots, \alpha_2\ N_v)$ followed by a bilinear interpolation to make sure the rotated matrix stays on a Cartesian grid. Hyperparameter $\Delta\phi = \frac{\pi}{492}$ was selected in this work. The rotated matrix was then reshaped back to a column vector with dimension of N^2. The L12 layer combines the contribution from each of the $\alpha_2\ N_v$ channels via a convolution kernel with dimension $1 \times 1 \times \alpha_2\ N_v$ followed by a linear activation to generate the final image with size of N^2. Note that the introduction of a separated rotation layer (L11) reduces the dimension of learnable parameters in L10 from $\alpha_2\ N_v N_c N^2$ to $N_c N^2$ and makes L10 trainable using limited GPU memory designed for personal computers.



	L1	**L2**	**L3**
Parameters	$64{,}3 \times 1 \times 1$	$64{,}3 \times 1 \times 64$	$1{,}3 \times 1 \times 129$
Output	$64, N_c \times N_v$	$64, N_c \times N_v$	$1, N_c \times N_v$
	L4	**L5**	**L6**
Parameters	$\alpha_1 N_v, 1 \times 1 \times N_v$	$\alpha_2 N_v, 1 \times 1 \times \alpha_1 N_v$	$1, N_c \times \alpha_2 N_v \times 1$
Output	$\alpha_1 N_v, N_c \times 1$	$\alpha_2 N_v, N_c \times 1$	$1, N_c \times \alpha_2 N_v$
	L7	**L8**	**L9**
Parameters	$16, \beta \times 1 \times 1$	$1, \beta \times 1 \times 16$	$N_c, 1 \times 1 \times N_c$
Output	$16, N_c \times \alpha_2 N_v$	$1, N_c \times \alpha_2 N_v$	$N_c, 1 \times \alpha_2 N_v$
	L10	**L11**	**L12**
Parameters	$N^2, 1 \times 1 \times N_c$	n/a	$1{,}1 \times 1 \times \alpha_2 N_v$
Output	$N^2, 1 \times \alpha_2 N_v$	$\alpha_2 N_v, N^2 \times 1$	$1, N^2 \times 1$

Fig. 2. Number of kernels and kernel dimensions as well as the corresponding output in all twelve iCT-Net layers.

To help keep track the number of training parameters and the dimension of each layer, these parameters are summarized in Figure 2. Each entry in this table consists of the first number to denote the number of kernels and the tuple followed by the comma denotes the dimension of the used kernel in each layer. For example, $(64, 3 \times 1 \times 1)$ in L1 layer means that there are 64 kernels with dimensions $3 \times 1 \times 1$.

B. Training Strategies

To maximize the potential generalizability of the trained iCT-Net, training datasets should be maximally expanded to include a wide variety of human anatomy at a wide variety of x-ray exposure levels. Although it is possible to access the anonymized clinical CT image data with a variety of human anatomy and other animal anatomy, it is very difficult to obtain data with a wide variety of radiation dose levels. Additionally, the quality of training data acquired from real CT scanners may be compromised due to physical confounding factors such as beam hardening, scatter, the x-ray tube heel effect, and the limited dynamic range of x-ray detectors. To minimize the impact of these confounding factors without compromising the applicability of the trained iCT-Net in experimental evaluations, a two-stage training strategy was used in this study. The first training stage was performed using numerical simulation data and the second training stage was performed using experimental data acquired from a 64-slice MDCT scanner (Discovery CT750 HD, GE Healthcare, Waukesha, WI).

1) Stage-1 Training: This stage includes both a segment-by-segment pre-training phase followed by an end-to-end training phase. The pre-training for the segment L1-L3 was performed using paired training data with low dose (high noise) projection data as input and high dose (low noise) projection data as output. The segment L4-L5 was pre-trained using sinograms with sparse view angles as input and sinograms with dense view angles as output. The segment L7-L9 was pre-trained using sinogram data with dense view angles as input and the corresponding sinograms filtered with a conventional Ram-Lak filter as output. Note that for the interior problem, the input sinogram data are truncated, but the output data used in pre-training are a correspondingly truncated portion of the filtered data generated by applying the Ram-Lak filter to the non-truncated data. In the segment-by-segment pre-training stage,