Deep Learning Computed Tomography: Learning Projection-Domain Weights From Image Domain in Limited Angle Problems

Tobias Würfl, Mathis Hoffmann, Vincent Christlein, Katharina Breininger, Yixin Huang, Mathias Unberath, and Andreas K. Maier

***Abstract*— In this paper, we present a new deep learning framework for 3-D tomographic reconstruction. To this end, we map filtered back-projection-type algorithms to neural networks. However, the back-projection cannot be implemented as a fully connected layer due to its memory requirements. To overcome this problem, we propose a new type of cone-beam back-projection layer, efficiently calculating the forward pass. We derive this layer's backward pass as a projection operation. Unlike most deep learning approaches for reconstruction, our new layer permits joint optimization of correction steps in volume and projection domain. Evaluation is performed numerically on a public data set in a limited angle setting showing a consistent improvement over analytical algorithms while keeping the same computational test-time complexity by design. In the region of interest, the peak signal-to-noise ratio has increased by 23%. In addition, we show that the learned algorithm can be interpreted using known concepts from cone beam reconstruction: the network is able to automatically learn strategies such as compensation weights and apodization windows.**

***Index Terms*— Reconstruction algorithms, neural networks, machine learning.**

I. Introduction

DEEP LEARNING has revolutionized the fields of signal processing and pattern recognition. Many of those advances have been transferred successfully to the field of medical image processing, promising large improvements on image understanding tasks like computer aided diagnosis [1], [2]. Those promising results motivate the question whether similar techniques can be exploited to improve 3-D reconstruction. While 3-D scene retrieval from 2-D observations has been studied in the computer vision context, the problem statements are considerably different than the ones faced in cone-beam computed tomography (CBCT). Consequently, the application of deep learning techniques to this field may require fundamentally new approaches.

A recent perspective article [3] shared a vision of utilizing machine learning to create a new class of data-driven image reconstruction algorithms to improve on traditional analytic and iterative methods. The authors conclude that challenges eluding proper mathematical modelling seem to be particularly promising candidates for machine learning approaches.

One of the most famous challenges of this category is tomographic CBCT reconstruction from incomplete data. In this work, we are particularly interested in the limited angle problem. Emerging artifacts are deterministic suggesting that methods based on machine learning, e. g. using neural networks, may effectively suppress those and thereby outperform traditional approaches. As envisioned by Wang [3], a Neural Network devised for this task must model the complete tomographic reconstruction algorithm. We showed that this is possible for parallel and fan-beam geometry in publication [4]. Recently it has been shown, that such a strategy, incorporating known operators like backprojection instead of relying on a hand-crafted network structure decreases maximal error bounds [5]. An extension of our architecture was subsequently presented in a joint work by Hammernik *et al.* [6] using a variational network performing the non-linear filtering, based on compressed sensing theory.

Despite promising performance, the previously proposed methods are limited to parallel and fan-beam geometries and rather small network sizes. Since practically all available CT scanners acquire projection data in cone-beam geometry, the applicability of these methods to real-world problems and, thus, their attractiveness, is greatly reduced.

In this paper, we extend our method to the clinically relevant cone-beam geometry. We express the widely used Feldkamp-Davis-Kress (FDK) algorithm in terms of a neural network. To this end, we introduce an efficient, differentiable cone-beam back-projection layer. This enables an end-to-end learning of various neural network architectures for reconstruction which refers to the potential for joint optimization of correction steps

Manuscript received February 14, 2018; revised March 31, 2018 and April 25, 2018; accepted April 26, 2018. Date of publication May 7, 2018; date of current version May 31, 2018. This work was supported by the National Institute of Biomedical Imaging and Bioengineering under Grant EB017095 and Grant EB017185. *(Corresponding author: Tobias Würfl.)*

T. Würfl, M. Hoffmann, V. Christlein, K. Breininger, Y. Huang, and A. K. Maier are with the Pattern Recognition Lab, Department Informatik, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany (e-mail: tobias.wuerfl@fau.de).

M. Unberath is with the Computer Aided Medical Procedures, Johns Hopkins University, Baltimore, MD 21218 USA.

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

Digital Object Identifier 10.1109/TMI.2018.2833499

both in volume and in projection domain. Using this a whole new class of reconstruction algorithms, replacing heuristic compensation methods in the reconstruction pipeline with data-driven solutions becomes available. Additionally, because of the strong link to reconstruction theory, we benefit from very good initializations and keep the interpretation of the original algorithm. We apply our method to a simulated limited angle reconstruction problem, based on the Low-Dose CT Grand Challenge data. Our Tensorflow GPU implementation of the proposed method and example data can be found at: https://github.com/ma0ho/Deep-Learning-Cone-Beam-CT.

This work is structured as follows: in Section II, we review deep learning applied to CBCT reconstruction and briefly mention analytic approaches that address the limited angle problem. Section III introduces the proposed method where we develop a reconstruction network for the parallel beam geometry. We extended this approach first to fan-beam and then to the clinically relevant cone-beam geometry. The evaluation is carried out in Section IV after the dataset and evaluation metrics are outlined. The paper is concluded in Section VI.

II. Related Work

Wang [3] argues that learning-based approaches may be particularly promising for applications that cannot be solved analytically as they lack proper mathematical formulation. This assumption seems justified when considering previous work on machine learning in CBCT reconstruction.

Our method considers mapping an analytic reconstruction pipeline to a neural network as a one-to-one correspondence. As a consequence prior to training, our networks represent an implementation of this pipeline. This presents the advantage, that they can be adapted to specific settings violating the requirements for exact reconstruction by optimization. Because the learned steps still correspond to their analytic counterparts, they can be readily interpreted after training. We are not aware of a similar method to adapt analytic algorithms simultaneously in projection and volume domain to training data.

Claus *et al.* [7] devise a method that addresses metal artifact reduction. They train a neural network that artificially removes metal implants from projection images via inpainting to reduce metal artifacts occurring during reconstruction.

Chengl *et al.* [8] present a deep learning method that predicts detailed versions of crudely reconstructed volumes to speed up convergence in iterative reconstruction algorithms, a method referred to as leapfrogging. In this approach, the output of the neural network is further refined using traditional iterative reconstruction methods.

Kang *et al.* [9] applied neural networks to a denoising task. Their network operates on the contourlet domain and predicts noise coefficients, which are subsequently subtracted from the original image to receive a denoised image. They present evidence that learning a mapping from a corrupted image to a corruption map is more stable than directly learning a mapping to an uncorrupted image.

This approach showed great promise for denoising low dose images, by scoring the second place in the "2016 NIH-AAPM-Mayo Clinic Low Dose CT Grand Challenge". Since then, additional denoising approaches have been proposed that either use a different network architecture [10] or introduce a novel perceptual loss function [11].

Apart from the denoising problem, neural networks have been applied to sparse view reconstruction [12], [13] applying variants of the popular U-net architecture [14] and the idea of learning the difference to an uncorrupted image.

Within this manuscript, we are most concerned with challenges of CBCT reconstruction in the limited angle case where data is acquired with an insufficient angular coverage of 180°. In the same context, Gu and Ye [15] trained a neural network to remove artifacts as a post reconstruction method. They use a U-net type architecture and perform training in the wavelet domain. Similar to Kang *et al.* [9], they report that the prediction of artifact images is more stable compared to directly estimating a restored image.

Floyd [16] presented a method to learn the reconstruction filter of an analytic reconstruction algorithm. While this similar to our work, it is not as general, as our framework allows to learn every other possible step in a reconstruction algorithm jointly in projection and volume domain. Another algorithm capable of learning filters was presented by Pelt and Batenburg [17], [18]. They learn a non-linear combination of multiple reconstructions. Within the framework proposed in our work, any general element of a single reconstruction can be learned. Their method increases the complexity of the algorithm proportional to the number of reconstructions used, while the test-time complexity of our method is the same as for analytic algorithms. Additionally, our methods can be extended to enable learning any step of the reconstruction algorithm.

Another approach to apply deep learning to the reconstruction problem uses neural networks to learn optimization methods. Examples for this strategy are given by [19], [20], and [21]. A general downside of those methods is their iterative solution that renders them computationally costly.

An alternative is unrolling an iterative algorithm to a fixed N-step iterative algorithm [22], [23], [24]. However, the complexity of such an algorithm, while being fixed is still much higher than the complexity of analytical algorithms our method is focussed on.

Finally, many traditional approaches to the limited angle problem exist. Noo *et al.* [25] devised a special purpose reconstruction formula which is capable of exact reconstruction but is limited to a reconstruction within a small region of interest. Riess *et al.* [26] proposed a heuristic combination of filters that are applied in projection and reconstruction domain, respectively. Their method modifies the well known short scan weights by Parker [27] to cope with the missing rays in projection domain. Subsequently, they apply a bilateral filter to remove remaining streak artifacts in the reconstructed volume. Recently, both strategies have been combined by Schäfer *et al.* [28] to yield a solution capable of an exact reconstruction in a region of interest, which additionally benefits from the heuristic compensation weights outside this region. They argue that a weakness of the heuristic weights lies in their non-smooth transitions. They compensate for this by means of apodization. Another well known approach to limited angle

reconstruction are regularized iterative algorithms. An algorithm specifically designed for limited angle tomography was presented by Huang *et al.* [29]. Downsides of these approaches is their high computational complexity and their assumption of a piecewise constant object.

III. Methodology

In Section III-A, we convert the filtered back-projection (FBP) algorithm in parallel-beam geometry to a neural network. Subsequently, Section III-B shows the extension to fan-beam geometry. Next, we present our new extension to cone-beam geometry in Section III-C by introducing the FDK algorithm and its mapping to a neural network. Finally, in Section III-D, we discuss regularization methods suitable for our model.

A. Parallel-Beam Geometry

We start our derivation using a formulation of a reconstruction as a least-squares minimization problem of the following objective function [30]:

$$L(\mathbf{f}) = \frac{1}{2}\|\mathbf{Af} - \mathbf{p}\|_2^2, \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^{M \cdot P}$ denotes the projection data, $\mathbf{A} \in \mathbb{R}^{N \times M \cdot P}$ denotes the forward projection model, $\mathbf{f} \in \mathbb{R}^N$ denotes the reconstruction for which the function is minimized, M the number of pixels in one projection, P the number of projections, and N the number of pixels/voxels in the reconstructed image. Two solutions to the problem are found considering the gradient with respect to $\mathbf{f}$:

$$\frac{\delta L(\mathbf{f})}{\delta \mathbf{f}} = \mathbf{A}^T(\mathbf{Af} - \mathbf{p}). \tag{2}$$

Following the negative gradient direction gives rise to iterative solution schemes. A direct solution is found at the point, where the gradient vanishes $\frac{\delta L(\mathbf{f})}{\delta \mathbf{f}} \stackrel{!}{=} \mathbf{0}$:

$$\mathbf{f} = \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{p}. \tag{3}$$

In parallel-beam geometry, the operator $\mathbf{A}_{\text{pb}}$ is also known as the Radon transform. Its adjoint $\mathbf{A}_{\text{pb}}^T$ is readily available as the back-projection operator. These operators are also the basis for analytical reconstruction, which can be expressed in continuous domain as:

$$f(x, y) = \int_0^\pi p(u, \theta) * h(u)|_{u = x\cos\theta + y\sin\theta} d\theta, \tag{4}$$

where x, y denote a coordinate of a point in the reconstruction, i.e. one index in $\mathbf{f}$. Thus, we can rewrite this equation using the operator notation from before to:

$$\mathbf{f} = \mathbf{A}_{\text{pb}}^T \mathbf{Cp}, \tag{5}$$

where $\mathbf{C}$ denotes the convolution of the projection data $p(u, \theta)$ with the discrete ramp filter $h(u)$. This corresponds to the $(\mathbf{AA}^T)^{-1}$ in the general case of Eq. 3. Note that for this algorithm it is sufficient to acquire data with an angular coverage of 180°, since $p(u, \theta) = p(-u, \theta + 180°)$ [30] with u denoting the spatial coordinate of the projection data.

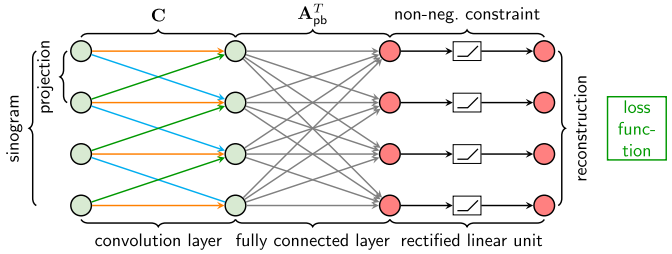


Fig. 1. Parallel-beam architecture. Green nodes represent intermediate results in volume domain, red nodes intermediate results in projection domain. The color of arrows denotes weight sharing.

The mapping to a neural network is straightforward by specifying layers that implement the operators $\mathbf{A}_{\text{pb}}^T$ and $\mathbf{C}$. Operator $\mathbf{C}$ is readily implemented as a convolutional layer with a single one-dimensional filter with its size equal to the spatial dimension of the projection. Note that unlike conventional filters employed in deep learning, the ramp filter has an infinite impulse response. This means that it is conventionally implemented as a filter with the same extent as a single acquired projection. Consequently, this operation is usually implemented in Fourier rather than in spatial domain.

In theory, $\mathbf{A}_{\text{pb}}^T$ can easily be mapped to a neural network as it represents a matrix multiplication and, therefore, can be implemented as a fully connected layer. In terms of a neural network, the operation of such a layer can be expressed as:

$$\mathbf{x}_{l+1} = \mathbf{W}\mathbf{x}_l, \tag{6}$$

where $\mathbf{W}$ denotes the matrix multiplication with the weights. For the reconstruction, the vector $\mathbf{x}_{l+1}$ can be regarded as the reconstruction $\mathbf{f}$, while the input $\mathbf{x}_l$ is the filtered projection data $\mathbf{Cp}$. However, as known from iterative reconstruction, the number of weights of this matrix is $N \cdot M \cdot P$ which can easily amount to several terabytes for modest problem sizes. This renders a straightforward implementation as fully connected layer infeasible. However, we can adopt a solution similar to iterative reconstruction. To this end, we construct a new back-projection layer without adjustable parameters. This new layer can compute its forward pass identifying $\mathbf{W}$ with the computation of $\mathbf{A}_{\text{pb}}^T$. To be able to adjust parameters of the neural network in projection domain, we have to calculate the gradient of this layer with respect to its inputs. Using backpropagation, the derivative of a fully connected layer with respect to its inputs is computed as:

$$\mathbf{e}_{l-1} = \mathbf{W}^T\mathbf{e}_l, \tag{7}$$

where $\mathbf{e_l}$ denotes the intermediate term that is commonly known as the error in deep learning. Similar to the derivation of the forward pass, we replace $\mathbf{W}$ with the back-projection operation $\mathbf{A}_{\text{pb}}^T$ and obtain:

$$\mathbf{e}_{l-1} = (\mathbf{A}_{\text{pb}}^T)^T\mathbf{e}_l = \mathbf{A}_{\text{pb}}\mathbf{e}_l. \tag{8}$$

This result enables the back-projection layer to efficiently calculate its forward and backward pass as fixed function without ever storing the complete matrix in memory.

For the reconstruction of attenuation values, it is a general requirement that solutions are non-negative, as negative values

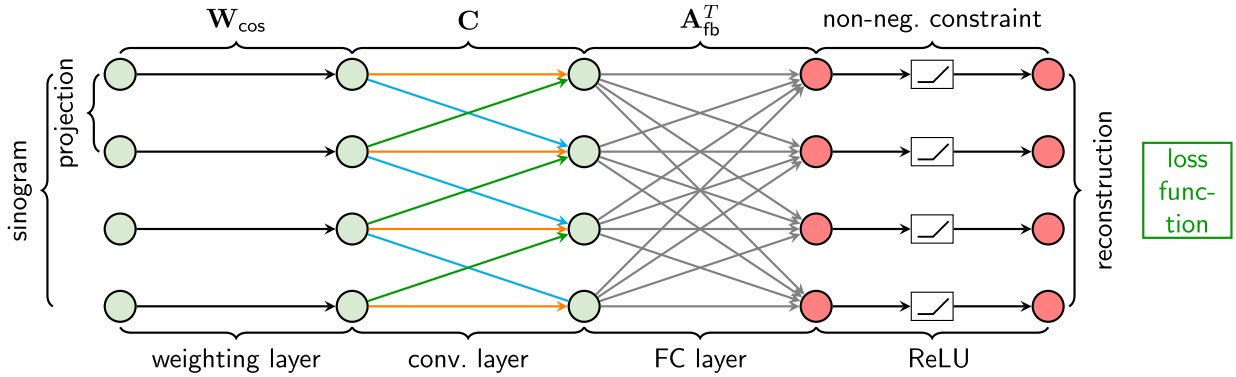


Fig. 2. Fan-beam architecture. Green nodes represent intermediate results in volume domain, red nodes intermediate results in projection domain. The color of arrows denotes weight sharing.

would indicate a source embedded in the imaged object. However, many artifacts in CT imaging can cause negative attenuation values in the reconstruction. It is sensible to enforce this constraint in a neural network model of reconstruction. The non-negativity constraint $f_i \geq 0 \, \forall f_i \in \mathbf{f}$ on the reconstruction can be enforced as:

$$\hat{\mathbf{f}} = \max(0, \mathbf{f}). \tag{9}$$

In deep learning, this constraint is known as the Rectified Linear Unit (ReLU) activation function. Since we do not expect negative values for any reconstruction, ReLU can be employed without loss of generality. The resulting network architecture is displayed in Figure 1.

B. Fan-Beam Geometry

The first generation of CT imaging system acquired data in parallel-beam geometry. This generation of scanners required translating the source only allowing the measurement of a single line at each position, which is a very time consuming procedure. Thus, the following generation of systems introduced an acquisition of whole detector rows at once in a fan-beam geometry. This requires a change of variables in the parallel-beam FBP formula. The non-trivial Jacobian of the change in variables introduces different operators. These consist of an element-wise cosine weighting of the projection data and a change of the back-projection operator to the fan-beam geometry. Additionally a distance weight becomes necessary which is conveniently incorporated in the back-projection operator. In operator notation, the FBP algorithm for fan-beam geometry can be expressed as:

$$\mathbf{x} = \mathbf{A}_{\text{fb}}^T \mathbf{C}\mathbf{W}_{\cos}\mathbf{p}, \tag{10}$$

where $\mathbf{W}_{\cos}$ denotes the pixel-wise independent weighting of the projection data with cosine weights. Thus $\mathbf{W}_{\cos}$ is a diagonal matrix and $\mathbf{C}\mathbf{W}_{\cos}$ corresponds to $(\mathbf{A}\mathbf{A}^T)^{-1}$ in the general case of Eq. 3. The resulting network architecture is displayed in Figure 2. Equation 10 is only valid for a full 360° angular coverage, a trajectory that is over-complete as every line through the object is measured twice. The minimal complete angular coverage in fan-beam geometry consists of $180° + \theta$, where θ denotes the fan-angle. However, this trajectory captures some redundant measurements. If not properly addressed, they lead to severe artifacts in the reconstructed image. The standard solution to this problem in analytic reconstruction is to weigh the data appropriately, e. g. using the scheme proposed by Parker [27]. We augment equation 10 to include redundancy weights, denoted as $\mathbf{W}_{\text{red}}$:

$$\mathbf{x} = \mathbf{A}_{\text{fb}}^T \mathbf{C}\mathbf{W}_{\text{red}}\mathbf{W}_{\cos}\mathbf{p}. \tag{11}$$

The mapping of this reconstruction algorithm to a neural network requires a layer representing these element-wise weightings using a diagonal matrix. This enables us to immediately derive the backward-pass of this operation as a multiplication of the error with the same matrix because for diagonal matrices $\mathbf{W} = \mathbf{W}^T$. Due to the diagonal structure, the number of weights is $M \cdot P$ suggesting that these weights can be stored in memory at all times and, thus, be learned. In order to do this we have to provide the gradient with respect to the weights $\frac{\delta \mathbf{w}_l}{\delta x_{l-1}}$ of this layer. Applying this to the well known matrix formulation of a fully connected layer, it follows:

$$\frac{\delta \mathbf{w}_l}{\delta x_{l-1}} = \mathbf{e}_l \mathbf{x}_{l-1}, \tag{12}$$

where $\mathbf{e}_l$ denotes the error with respect to this layer and $\mathbf{x}_{l-1}$ denotes the activation of the layer prior to layer l. This shows that the update is again an element-wise multiplication.

Since the convolution with the Ramp filter remains completely unchanged, the last thing we need to construct is a fan-beam back-projection layer. This can be achieved once more by calculating the fan-beam back-projection in the forward pass of the layer and using fan-beam projection to calculate the derivatives with respect to the input. The fan-beam back-projection operator A_{fb}^T usually includes the distance weighting resulting from the change of variables:

$$\left(\frac{D}{D + r \sin(\beta - \phi)} \right)^2, \tag{13}$$

where D denotes the focal length, r, ϕ denote the coordinates of the point on the detector to reconstruct, and β denotes the current projection angle of the system. The incorporation of this distance weights in the back-projector is straightforward. However, their inclusion means that they have to be also considered during the backward pass of the network when calculating A_{fb}. This becomes a distance weighted forward

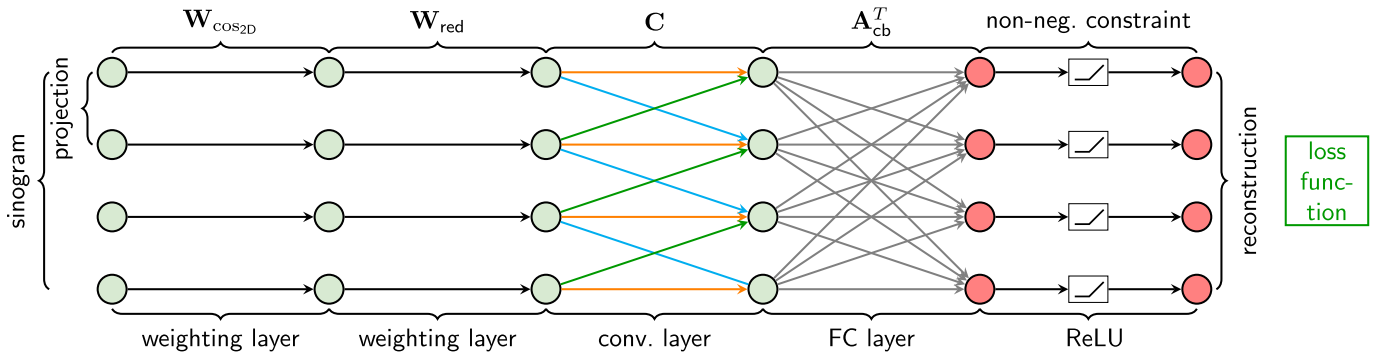


Fig. 3. Short scan cone-beam architecture. Green nodes represent intermediate results in volume domain, red nodes intermediate results in projection domain. The color of arrows denotes weight sharing.

projection that is different from iterative algorithms and can be addressed with a matched projection strategy.

C. Cone-Beam Geometry

We now turn to the problem of extending the previously presented methods to the problem of reconstruction from projections acquired in the clinically relevant cone-beam geometry. This problem is substantially more challenging than reconstruction from parallel or fan-beam projections. The reason is that the dimensionality of the problem is drastically increased as considering cone-beam geometry is associated with a transition to a 3-D, rather than a 2-D, reconstruction problem. A broad range of 3-D scan trajectories and a multitude of algorithms capable of exact reconstruction and many approximative methods exist. The choice between these algorithms is constrained by the requirements on the acquisition trajectory of source and detector.

We focus on incomplete circular trajectories as they are the de facto standard in flat-panel CBCT applications. A popular algorithm for the reconstruction of those is the FDK algorithm. This algorithm extends the fan-beam algorithm by assuming the object to be invariant in z-direction. It can be expressed conveniently in operator notation and reads:

$$\mathbf{f} = \mathbf{A}_{\text{cb}}^T \mathbf{C}_{\text{2D}} \mathbf{W}_{\cos_{\text{2D}}} \mathbf{p}. \tag{14}$$

Here $\mathbf{W}_{cos_{\text{2D}}}$ denotes a two-dimensional cosine weighting of the projection data $\mathbf{p}(\beta, u, v)$ according to the angle between the current and the principal ray. For a particular detector pixel, the weight is computed as:

$$\frac{D}{\sqrt{D^2 + \hat{u}^2 + \hat{v}^2}}, \tag{15}$$

where D again denotes the focal length, while $\hat{u}$ and $\hat{v}$ denote the coordinates of the considered detector pixel in horizontal and vertical direction, respectively. Weights obtained in this way can be incorporated in our new cone-beam reconstruction network by using yet another weighting layer. The convolution operator $\mathbf{C}$ still denotes a one-dimensional row-wise convolution of the data along the u-direction with the ramp-filter $h(u)$.

Equation 14 also contains a new operator $\mathbf{A}_{\text{cb}}^T$ which is the three-dimensional back-projection with a distance weighting. The transpose of this back-projection is, again, the cone-beam projection operator. With its help, we can map operator $\mathbf{A}_{\text{cb}}^T$ to a layer in a neural network by computing the cone-beam back-projection in the forward-pass of the network and using the projection to calculate the derivatives with respect to the input. However, we need to incorporate the distance weights from the change of coordinates, similar to the fan-beam back-projection layer. These can be expressed as:

$$\left(\frac{D}{D-s}\right)^2, \tag{16}$$

where s denotes the distance of the reconstruction domain point to the detector plane. To effectively address this issue, we propose to implement a matched projector $\mathbf{A}_{\text{cb}}$.

It is known that data acquired on a circular source and detector trajectory does not satisfy Tuy's condition and is, hence, incomplete. Still, the heuristic adaptation of Parker weights [27] is a popular choice. We follow the literature and extend the short-scan weights to cone beam geometry:

$$\mathbf{f} = \mathbf{A}_{\text{cb}}^T \mathbf{C}_{\text{2D}} \mathbf{W}_{\text{red}_{\text{2D}}} \mathbf{W}_{\cos_{\text{2D}}} \mathbf{p}. \tag{17}$$

We display our resulting network architecture in Figure 3.

D. Regularization

Pre-training enabled the first successful deep learning models. Since then, new methods like the ReLU activation function and transfer-learning have made pre-training less common. However, since our models are constructed after one-to-one correspondences of analytical algorithms we can initialize our layers with the exact solutions given by reconstruction theory. This yields neural networks capable of performing analytical reconstruction after they have been initialized. This provides a very strong starting point for learning data-optimal algorithms which cannot be formulated analytically.

IV. Evaluation

We apply our model to the problem of limited angle tomography. In Section IV-A, we describe the data that we have used for our experiments. In Section IV-B, we focus on the implementation of our model and list its hyperparameters. Then, in Section IV-C, we briefly discuss the evaluation metrics before we show our results in Section IV-E.

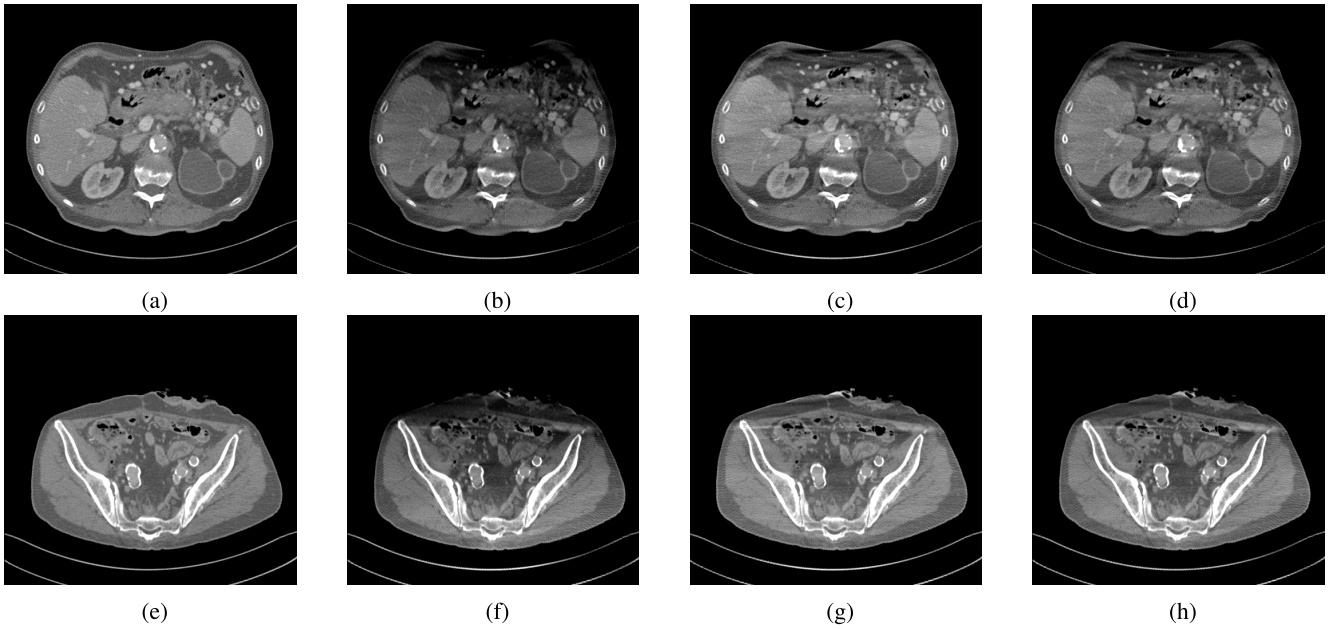


Fig. 4. Figures 4a and 4e show examples of ground truth slices. Using only half of the projections causes loss of mass as shown in Figures 4b and 4f. Our trained model can compensate for the loss of mass. The result is shown in Figures 4d and 4h. A similar result is achieved by the heuristic compensation weights proposed by Riess *et al.* [26] (Figures 4c and 4g).

A. Data

For our experiments, we use data that was released as part of the Low Dose CT Grand Challenge. As the provided projection data was acquired using a helix trajectory, we cannot use it for the circular trajectory. Instead, we perform a forward projection of the provided reference volumes that were reconstructed from the full dose projections using a slice thickness of 1mm. For the forward projection, we perform 360 projections onto a detector of 720 pixels width and 880 pixels height and reconstruct cubic volumes of 512^3 voxels. For our cross-validation, we calculate reconstructions with a reduced resolution of 256^3 voxels. The size of each detector pixel is $1 \times 1\text{mm}^2$ and the angular increment between every projection is 1Â°. The distance between radiation source and detector is 1200mm. Training took approximately five hours for each fold on a machine equipped with an NVIDIA GTX 1080, an Intel Xeon E5-1630 v3 @ 3.70GHz and 64 Gb RAM. The volume was placed in the origin of the world coordinate system which also represents the axis of rotation. We simulate the limited angle problem by limiting the angular range to 180°.

B. Implementation Details

We chose to implement the model in Tensorflow [31] which, except for the back-projection layer, readily provides all required functionality. We implemented the back-projection layer as a custom Tensorflow operation and use CUDA to speed up the computation. The implementation relies on a formulation using projection matrices, which is the conventional format for flat-panel cone-beam CT data. As detailed in section III-C, the gradient of the back-projection operation can be calculated using the projection operator $\mathbf{A}_{\text{cb}}$ with incorporated distance weights. To realize this, we decided to implement a matched projector, calculating the weights of the forward projection for every voxel and distributing the error according to those. Hence, we have an exact implementation of the transpose of $\mathbf{A}_{\text{cb}}^T$.

For training the network, the loss L of the network output $\mathbf{f}_m$ (the reconstructed volume) needs to be computed with respect to the ground truth $\mathbf{f}_r$. $\mathbf{f}_r$ follows by feeding all 360 projection images through a reference network that has been initialized using the analytic weights of the FDK algorithm. Using the Euclidean loss function we receive:

$$L = \|\mathbf{f}_r - \mathbf{f}_m\|_2^2. \tag{18}$$

We use L to train the parameters of the redundancy weighting layer since these are the only adjustable parameters able to compensate for the missing data. For the parameter update, we use a simple gradient descent scheme with a learning rate of 0.2×10^{-8}. We deliberately set the learning rate for the filter to 0 as we do not expect dominant changes.

We perform a leave-one-out cross-validation on this data set, using each volume once for evaluation. To determine the point for early stopping, for each fold we randomly select one dataset as validation set and use the remaining eight volumes for training. The trained algorithm is hereby limited to this specific acquisition geometry. In Section IV-E, we report the test results for each fold of this cross-validation.

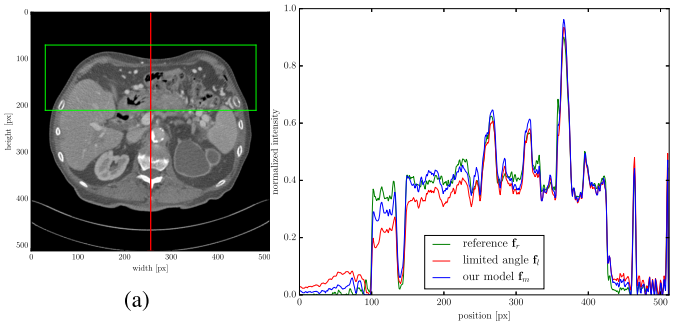
C. Evaluation Metrics

Two common metrics to predict the perceived quality of images are structural similarity (SSIM) [32] and peak signal-to-noise ratio (PSNR). SSIM combines a luminance measure, a contrast measure and a structure measure. It is computed over a multi-scale representation of the volumes and is averaged to yield a final score. We used an implementation of SSIM that matches the description of Wang *et al.* [32]. Therefore, we use Gaussian weights instead of sharp windows and set the algorithmic parameters accordingly. The PSNR is

TABLE I

SSIM AND PSNR FOR THE RECONSTRUCTION USING PARKER WEIGHTS, THE PROPOSED APPROACH, A U-NET BASED RECONSTRUCTION [15], AND THE ITERATIVE WTV METHOD [29] FOR EACH FOLD OF A 10-FOLD CROSS VALIDATION

	SSIM				PSNR			
	Parker	Proposed	U-net	wTV	Parker	Proposed	U-net	wTV
fold 1	0.675	**0.770**	0.539	0.582	22.54 dB	28.45 dB	26.26 dB	**33.12** dB
fold 2	0.652	**0.736**	0.548	0.555	25.37 dB	29.78 dB	27.84 dB	**35.24** dB
fold 3	0.637	**0.704**	0.490	0.524	23.79 dB	27.10 dB	19.18 dB	**30.03** dB
fold 4	0.683	**0.795**	0.572	0.584	19.83 dB	25.74 dB	23.22 dB	**30.33** dB
fold 5	0.611	**0.691**	0.604	0.537	26.18 dB	30.83 dB	30.63 dB	**36.59** dB
fold 6	0.639	**0.769**	0.482	0.551	23.52 dB	31.11 dB	26.75 dB	**34.05** dB
fold 7	0.676	**0.789**	0.579	0.597	20.66 dB	27.28 dB	23.40 dB	**30.68** dB
fold 8	0.679	**0.738**	0.505	0.557	20.88 dB	28.93 dB	22.61 dB	**32.43** dB
fold 9	0.637	**0.744**	0.680	0.565	22.64 dB	31.28 dB	26.64 dB	**32.18** dB
fold 10	0.630	**0.727**	0.645	0.530	26.32 dB	35.26 dB	26.76 dB	**35.32** dB
average	0.652	**0.746**	0.564	0.558	23.17 dB	29.57 dB	25.33 dB	**33.00** dB



(a) (b)

Fig. 5. Figure 5a shows the line along which the intensity profile (figure 5b) has been computed in red. Green: Region of interest that has been used to compute SSIM and PSNR (Tables I and II). Figure 5b depicts the intensity profiles along the line shown in Figure 5a.

defined as

$$\mathrm{PSNR}(\mathbf{f}_r, \mathbf{f}_m) = 10 \log_{10} \left(\frac{\max (\mathbf{f}_r \odot \mathbf{f}_r)}{\|\mathbf{f}_r - \mathbf{f}_m\|_2^2} \right), \tag{19}$$

where $\odot$ denotes element-wise multiplication. We compute both evaluation measures over a volumetric region of interest, in order to preserve the sensitivity of our evaluation measures. This is needed since only parts of the volume are affected by limited angle artifacts. The selected 3D region intersecting a volume slice is shown in Figure 5a. We report results for two other methods on the same data and geometry.

D. Reference Methods

We compare our method to an iterative reconstruction method proposed by Huang *et al.* [29]. We performed 200 iterations of the weighted total variation (wTV) method using the same parameters as in [29]. Additionally, we report results of a method using a multi-resolution network that has previously been used as a baseline in [15]. Similar to [15], we trained the U-net on individual slices instead of volumes. The U-net method is optimized using the Adam optimizer with a learning rate of 10^{-3} the decay for the first moment estimate $\beta_1 = 0.9$, the decay for the second moment estimate $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

E. Results

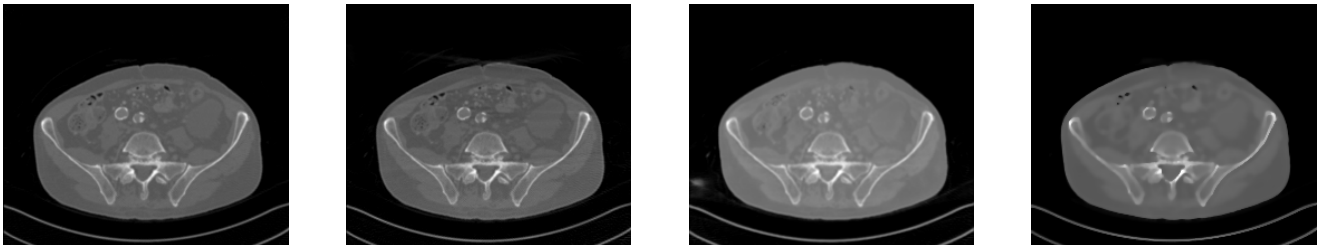
It turns out that the training process is very stable without any further adjustments. This has multiple reasons. Our model is very wide, but quite shallow enabling easy learning compared to deep architectures which suffer from the vanishing gradient effect. Furthermore, our weighting layers are very easy to learn because the adjustable parameters are disentangled from each other.

The impact of adjusting the Parker weights by training as described in Section IV-B is readily visible in Figure 4. Figure 5b shows that the loss of mass, caused by missing data, is effectively compensated. However, Figures 4d and 4h also show that streak artifacts are still visible which are not compensated by this basic architecture.

In order to investigate the training results in greater depth, we show the learned weights in Figure 8a and compare them to the analytic parker weights [27] in Figure 8b. Note that we have smoothed the learned weights with a gaussian filter ($\sigma = 3$) for illustration purposes. In addition, we show the heuristic weights proposed by Riess *et al.* [26] (Figure 8c) that have been specifically designed for limited angle reconstruction. This comparison shows that our method learned similar boosted regions as shown in Figure 8c. Also note that the decay that has been learned at the begin of the scan between 700px and 800px as well as between 0px and 100px at the end of the scan (see Figure 8a) has recently been proposed as an improvement over these heuristic weights by Schäfer *et al.* [28]. This illustrates that our neural network model is not as opaque as conventional models. Because every fundamental stage of the architecture corresponds to a well understood stage of the analytic reconstruction, practitioners can readily interpret them.

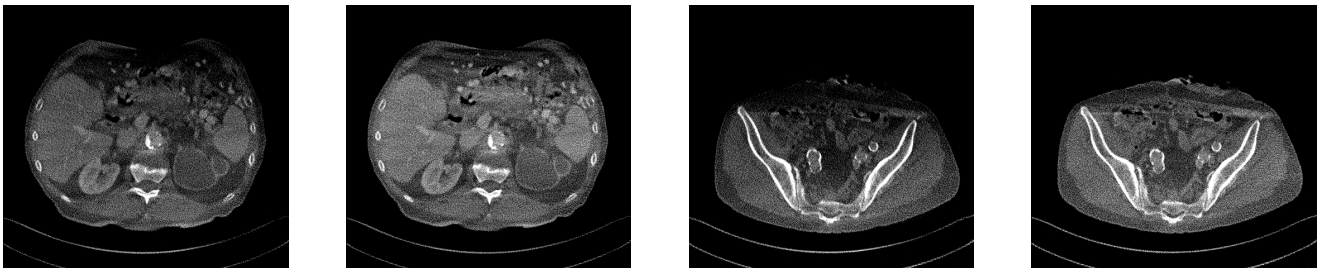
In Figure 4, we compare the results of a reconstruction using the weights proposed by Schäfer et. al (Figures 4c and 4g) with the reconstruction results using our trained weights (Figures 4d and 4h). The results are quite similar from a visual point of view. To investigate this in greater depth, we also compare the SSIM and PSNR for both weights (Table II). These results strengthen the assumption that the heuristic and learned weightsT perform similar. In addition, they show that the learned weights perform better than the heuristic weights in a noisy setting in terms of PSNR. However, the SSIM metric favours the reconstruction result using Parker weights, despite the prominent shading artifact. A reconstruction with noise is shown in Figure 7.

Table I compares the SSIM and PSNR of the limited angle reconstruction using Parker weights, the iterative method and



(a) (b) (c) (d)

Fig. 6. Exemplary results from the cross-validation: (a) Groundtruth, (b) result using the proposed method, (c) using U-net based approach, (d) using the wTV method.



(a) (b) (c) (d)

Fig. 7. To show the robustness of the learned weights to noise, we added Gaussian noise to the projection data. (a) and (c) show the reconstruction results using Parker weights while (b) and (d) show the reconstruction results using the learned weights.

TABLE II
SSIM AND PSNR IN COMPARISON TO ANALYTIC PARKER WEIGHTS AND WEIGHTS PROPOSED BY SCHÄFER *et al.* [28]. ADDITIONAL NOISE HAS BEEN ADDED TO THE PROJECTIONS WITH A STANDARD DEVIATION OF 0.5% OF THE MAXIMUM ATTENUATION

	SSIM	PSNR
Parker	0.849	27.07 dB
Parker + noise	0.628	26.13 dB
Schäfer	0.865	33.34 dB
Schäfer + noise	0.573	29.37 dB
Learned	0.886	33.17 dB
Learned + noise	0.604	29.67 dB

the U-net based method with the reconstruction using our trained model. While the wTV method performs best in terms of PSNR, the proposed method generally performs better in terms of SSIM. The reason for this is that the strong smoothing introduced by the total variation prior is strongly penalized by SSIM while it is less well reflected in PSNR. The reconstructions using the U-net method suffers from spurious objects it adds to the reconstruction, as seen on the left side of 6c as well as less obvious mistakes like reconstructed object boundaries that are reconstructed sharply but with an incorrect curvature.

V. DISCUSSION

In our method, we propose to learn reconstruction algorithms for limited angle data for parallel, fan-, and cone-beam projection geometries. The network architectures are derived from analytical reconstruction formulas. In fact, the pre-trained networks compute exactly the analytical reconstruction formula. By training, we adapt these algorithms to cases for which no analytical closed-form solution is known. The algorithm itself, however, is still of the same class and test-time complexity as the original reconstruction formula. Therefore, the degrees of freedom are limited and we are able to map the trained network again into an analytical reconstruction formula. Thus, we are able to interpret the learned adaptation in the context of classical analytic reconstruction theory. In our results, we observe that the learned weights mimic heuristic extensions of classical reconstruction formulas such as the Rieß' weights and their extension by Schäfer at al. This is in particular interesting as analytical reconstruction formulas are typically derived in continuous form. The implementation in a discrete computer system typically causes additional efforts in order to handle the discretization correctly. As such the ramp filter needs to be implemented according to the Ram-Lak convolver and appropriate apodization is required. For iterative methods, these problems do not emerge as the complete reconstruction formula is derived in discrete form. In our method, we observe that the network adapts to the discrete nature of the problem automatically. We believe that this is the result of our solution being optimal in an L2-sense with respect to the given training data. In conclusion our method enables to improve analytic algorithms in a data-driven fashion by adapting to specific scenarios while not relying on heuristics and staying very close to the properties of analytic algorithms.

At present, we only investigated the FDK algorithm for circular trajectories. However, the presented method could be easily applied to any other trajectory and reconstruction formula. In particular, we deem the investigation of exact reconstruction methods as Defrise-Clack [33] and Katsevich-type reconstruction formulas [34] promising as the proposed approach could mend problems as unused data or extend the formulas to new reconstruction orbits easily.

Many calibration and physical correction steps such as scatter correction and beam-hardening correction involve operations that could be easily embedded in a neural network as data correction steps de-coupled from back-projection.

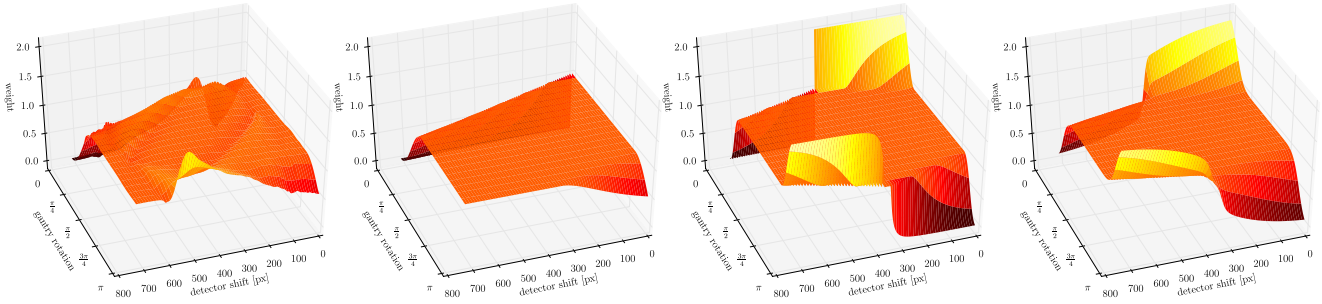


Fig. 8. Presentation of weights: (a) weights that have been learned by our model, (b) Parker weights, (c) weights as proposed by Riess *et al.* [26] without Gaussian smoothing, (d) smooth compensation weights proposed by Schäfer *et al.* [28].

Most of these correction methods estimate each correction independently and are not able to model inter-dependencies. Our method can be used to find optimal end-to-end learned parameters for all steps jointly in projection and volume domain. In our framework, we could reuse the idea of pre-training the net using a previous calibration and perform end-to-end training to establish interrelations between the different correction steps. Of course, our method can also be combined with reconstruction-domain non-linear neural-network-based processing, e.g. the method of Gu and Ye [15], which demonstrated that noise artifacts are easier removed in wavelet domain. Learning those compensation steps provides a basis to replace heuristic algorithms with end-to-end optimized data-optimal solutions.

There is direct similarity of our approach to iterative algorithms. The key difference to those methods is that our neural network does not use these equations to iteratively compute a single solution for $\mathbf{f}$. Instead, it iteratively learns parameters that describe the optimized mapping from $\mathbf{p}$ to $\mathbf{f}$ in a single step. Since we fixed the back-projection layer, the only adjustable parameters in this basic network structure are the weights of the projection-domain layers. To learn more flexible mappings, we have to introduce additional layers that provide complimentary degrees of freedom. This reveals a second difference to iterative algorithms. The derivation of back-projection as a layer solely uses the linearity of the projection transform. This means that the whole transform can include arbitrary differentiable non-linearities and still use backpropagation to compute the necessary derivatives.

There are different approaches to implement $\mathbf{A}$ and $\mathbf{A}^T$. Typically, the forward projector is implemented pixel-driven and the back-projector voxel-driven for optimized execution time. As this does not correspond to the correct gradient for an iterative procedure, we call such an implementation unmatched, as opposed to the matched projector. Thus, we have the choice to perform the unmatched projection and apply the distance weights or implement the matched projector as the exact transpose of the forward operation. The unmatched projector computes the per-pixel value as the integral over the ray connecting the radiation source with that pixel. The main advantage of this implementation is, that it benefits from the texture memory in modern GPUs that provides hardware accelerated bilinear and trilinar interpolation. However, one still has to apply the distance weighting. It is unclear whether it is beneficial to implement the operator $\mathbf{A}$ as the exact transpose of $\mathbf{A}^T$ or as direct forward projection. This has been analyzed extensively in iterative reconstruction. An extensive analysis on the implications of such a choice was conducted by Zeng and Gullberg [35]. He showed that an unmatched projector solves a different optimization problem. However, in real datasets other effects such as noise and beam-hardening dominate the image quality and the choice of matched or unmatched projectors is not critical. Contrary to intuition, unmatched projectors have been shown to increase convergence speed in some cases. Mathematically our formulation is very similar. Thus, we expect these observations to hold for the back-projection layer, too.

Due to the non-convexity of the underlying optimization problem, well adapted regularization strategies are a key factor for a successful application of deep learning algorithms. Big advances have been gained by pre-training [36], Dropout [37] and Batch-normalization [38]. The motivation behind Dropout is to promote independent features. However, the independence assumption does not apply to the large scale regression problems of reconstruction. Both the weights and the elements of the filter can be regarded as discrete versions of smooth functions. Therefore, randomly setting elements to zero will only introduce a residual error without enforcing any sensible prior knowledge. Thus, we found Dropout unsuitable for our models. The internal covariate shift, which occurs with typical convolutional neural networks, also does not apply to our layers since they preserve a specific scale representing attenuation values. Thus, batch normalisation is also not appropriate for our models. Note that these regularization methods are not by definition useless for the application of deep learning to reconstruction. However, they only make sense if an intermediate representation using learned features is constructed. This is not the case in our model.

VI. Conclusion

We propose a deep learning method for cone-beam reconstruction, enabling joint learning of compensation steps in projection and volume domain. This is achieved by mapping the popular FDK algorithm to a neural network. Since the straightforward implementation is impossible, we introduce a novel cone-beam back-projection layer. We show that the derivative with respect to the inputs can be calculated efficiently using a weighted projection. This enables to expand

current architectures only acting as post-processing methods in volume domain to also include correction in projection domain. Many well researched artifacts in CT are typically accounted for in projection domain. Examples for this are physical effects such as beam hardening, scatter and metal artifacts. Other examples include missing data problems such as truncation correction or limited angle reconstruction.

We applied our method to a limited angle problem. By learning compensation weights we showed that we can correct the loss of mass typically caused by missing data. This improvement comes at no additional computational effort since we use the exact same operations as the analytical algorithm, with different weights. For future work, we would like to address the remaining streak artifacts, which are also caused by the missing data. They could be compensated by an additional non-linear filtering in volume domain.

Acknowledgment

The authors would like to thank Dr. C. McCollough, the Mayo Clinic, the American Association of Physicists in Medicine for providing the used data.

References

[1] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[2] F. C. Ghesu *et al.*, "Marginal space deep learning: Efficient architecture for volumetric image parsing," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1217–1228, May 2016.

[3] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, pp. 8914–8924, 2016.

[4] T. Würfl, F. C. Ghesu, V. Christlein, and A. Maier, "Deep learning computed tomography," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2016, pp. 432–440.

[5] A. K. Maier *et al.*, "Precision learning: Towards use of known operators in neural networks," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2018.

[6] K. Hammernik, T. Würfl, T. Pock, and A. Maier, "A deep learning architecture for limited-angle computed tomography reconstruction," in *Bildverarbeitung für die Medizin*. Berlin, Germany: Springer, 2017, pp. 92–97.

[7] B. E. H. Claus, Y. Jin, L. A. Gjesteby, G. Wang, and B. De Man, "Metal-artifact reduction using deep-learning based Sinogram completion: Initial results," in *Proc. 14th Int. Meeting Fully Three-Dimensional Image Reconstruction Radiol. Nucl. Med.*, 2017, pp. 631–634.

[8] L. Cheng, S. Ahn, S. G. Ross, H. Qian, and B. De Man, "Accelerated iterative image reconstruction using a deep learning based leapfrogging strategy," in *Proc. 14th Int. Meeting Fully Three-Dimensional Image Reconstruction Radiol. Nucl. Med.*, 2017, pp. 715–719.

[9] E. Kang, J. Min, and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction," *Med. Phys.*, vol. 44, no. 10, pp. e360–e375, 2017.

[10] H. Chen *et al.*, "Low-dose CT restoration with deep neural network," in *Proc. 14th Int. Meeting Fully Three-Dimensional Image Reconstruction Radiol. Nucl. Med.*, 2017, pp. 25–28.

[11] Q. Yang, P. Yan, M. K. Kalra, and G. Wang. (Feb. 2017). "CT image denoising with perceptive deep neural networks." [Online]. Available: https://arxiv.org/abs/1702.07019

[12] H. Li and K. Mueller, "Low-Dose CT streak artifacts removal using deep residual neural network," in *Proc. 14th Int. Meeting Fully Three-Dimensional Image Reconstruction Radiol. Nucl. Med.*, 2017, pp. 191–194.

[13] H. Y. Seob and J. C. Ye, "Deep Residual Learning Approach for Sparse-view CT Reconstruction," in *Proc. 14th Int. Meeting Fully Three-Dimensional Image Reconstruction Radiol. Nucl. Med.*, 2017, pp. 217–220.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[15] J. Gu and J. C. Ye, "Multi-scale wavelet domain residual learning for limited-angle CT reconstruction," in *Proc. 14th Int. Meeting Fully Three-Dimensional Image Reconstruction Radiol. Nucl. Med.*, 2017, pp. 443–447.

[16] C. E. Floyd, Jr., "An artificial neural network for SPECT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 10, no. 3, pp. 485–487, Sep. 1991.

[17] D. M. Pelt and K. J. Batenburg, "Fast tomographic reconstruction from limited data using artificial neural networks," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5238–5251, Dec. 2013.

[18] D. M. Pelt and K. J. Batenburg, "Improving filtered backprojection reconstruction by data-dependent filtering," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4750–4762, Nov. 2014.

[19] K. Kunisch and T. Pock, "A bilevel optimization approach for parameter learning in variational models," *SIAM J. Imag. Sci.*, vol. 6, no. 2, pp. 938–983, 2013.

[20] L. Calatroni, C. Cao, J. C. De Los Reyes, C.-B. Schönlieb, and T. Valkonen, "Bilevel approaches for learning of variational imaging models," in *Variational Methods*. Berlin, Germany: Walter de Gruyter GmbH, 2017, pp. 252–290.

[21] J. C. De Los Reyes, C.-B. Schönlieb, and T. Valkonen, "The structure of optimal parameters for image restoration problems," *J. Math. Anal. Appl.*, vol. 434, no. 1, pp. 464–500, 2016.

[22] J. Adler and O. Öktem, "Solving ill-posed inverse problems using iterative deep neural networks," *Inverse Problems*, vol. 33, no. 12, p. 124007, 2017.

[23] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE Trans. Med. Imag.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8271999/

[24] P. Putzky and M. Welling. (Jun. 2017). "Recurrent inference machines for solving inverse problems." [Online]. Available: https://arxiv.org/abs/1706.04008

[25] F. Noo, M. Defrise, R. Clackdoyle, and H. Kudo, "Image reconstruction from fan-beam projections on less than a short scan," *Phys. Med. Biol.*, vol. 47, no. 14, p. 2525, 2002.

[26] C. Riess, M. Berger, H. Wu, M. Manhart, R. Fahrig, and A. K. Maier, "TV or not TV—That is the question," in *Proc. 12th Int. Meeting Fully Three-Dimensional Image Reconstruction Radiol. Nucl. Med.*, 2013, pp. 341–344.

[27] D. L. Parker, "Optimal short scan convolution reconstruction for fan beam CT," *Med. Phys.*, vol. 9, no. 2, pp. 254–257, 1982.

[28] D. Schäfer, P. van de Haar, and M. Grass, "Modified Parker weights for super short scan cone beam CT," in *Proc. 14th Int. Meeting Fully Three-Dimensional Image Reconstruction Radiol. Nucl. Med.*, 2017, pp. 49–52.

[29] Y. Huang, O. Taubmann, X. Huang, V. Haase, G. Lauritsch, and A. Maier, "Scale-space anisotropic total variation for limited angle tomography," *IEEE Trans. Radiation Plasma Med. Sci.*, to be published.

[30] G. L. Zeng, *Medical Image Reconstruction*. Berlin, Germany: Springer-Verlag, 2009.

[31] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: http://tensorflow.org/

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] M. Defrise and R. Clack, "A cone-beam reconstruction algorithm using shift-variant filtering and cone-beam backprojection," *IEEE Trans. Med. Imag.*, vol. 13, no. 1, pp. 186–195, Mar. 1994.

[34] A. Katsevich, "Theoretically exact filtered backprojection-type inversion algorithm for spiral CT," *SIAM J. Appl. Math.*, vol. 62, no. 6, pp. 2012–2026, 2002.

[35] G. L. Zeng and G. T. Gullberg, "Unmatched projector/backprojector pairs in an iterative reconstruction algorithm," *IEEE Trans. Med. Imag.*, vol. 19, no. 5, pp. 548–555, May 2000.

[36] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 153–160.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.