

Multi-Slice Fusion for Sparse-View and Limited-Angle 4D CT Reconstruction

Soumendu Majee, *Student Member, IEEE*, Thilo Balke, *Student Member, IEEE*, Craig A. J. Kemp, Gregory T. Buzzard, *Member, IEEE* and Charles A. Bouman *Fellow, IEEE*

Abstract—Inverse problems spanning four or more dimensions such as space, time and other independent parameters have become increasingly important. State-of-the-art 4D reconstruction methods use model based iterative reconstruction (MBIR), but depend critically on the quality of the prior modeling. Recently, plug-and-play (PnP) methods have been shown to be an effective way to incorporate advanced prior models using state-of-the-art denoising algorithms. However, state-of-the-art denoisers such as BM4D and deep convolutional neural networks (CNNs) are primarily available for 2D or 3D images and extending them to higher dimensions is difficult due to algorithmic complexity and the increased difficulty of effective training.

In this paper, we present *multi-slice fusion*, a novel algorithm for 4D reconstruction, based on the fusion of multiple low-dimensional denoisers. Our approach uses multi-agent consensus equilibrium (MACE), an extension of plug-and-play, as a framework for integrating the multiple lower-dimensional models. We apply our method to 4D cone-beam X-ray CT reconstruction for non destructive evaluation (NDE) of samples that are dynamically moving during acquisition. We implement multi-slice fusion on distributed, heterogeneous clusters in order to reconstruct large 4D volumes in reasonable time and demonstrate the inherent parallelizable nature of the algorithm. We present simulated and real experimental results on sparse-view and limited-angle CT data to demonstrate that multi-slice fusion can substantially improve the quality of reconstructions relative to traditional methods, while also being practical to implement and train.

Index Terms—Inverse problems, 4D tomography, Model based reconstruction, Plug-and-play, Deep neural networks

I. INTRODUCTION

Improvements in imaging sensors and computing power have made it possible to solve increasingly difficult reconstruction problems. In particular, the dimensionality of reconstruction problems has increased from the traditional 2D and 3D problems representing space to more difficult 4D or even 5D problems representing space-time and, for example, heart or respiratory phase [1]–[6].

These higher-dimensional reconstruction problems pose surprisingly difficult challenges computationally and perhaps more importantly, in terms of algorithmic design and training due to the curse of dimensionality [7]. However, the high

dimensionality of the reconstruction also presents important opportunities to improve reconstruction quality by exploiting the regularity in the high-dimensional space. In particular, for time-resolved imaging, we can exploit the regularity of the image to reconstruct each frame with fewer measurements and thereby increase temporal resolution. In the case of 4D CT, the contributions of [2], [8], [9] have increased the temporal resolution by an order of magnitude by exploiting the space-time regularity of objects being imaged. These approaches use model-based iterative reconstruction (MBIR) [10], [11] to enforce regularity in 4D using simple space-time prior models. More recently, deep learning based post-processing for 4D reconstruction has been proposed as a method to improve reconstructed image quality [12].

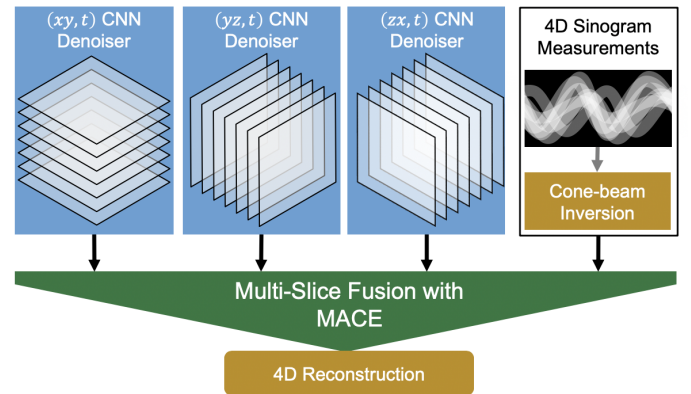


Fig. 1. Illustration of our multi-slice fusion approach. Each CNN denoiser operates along the time direction and two spatial directions. We fuse the CNN denoisers with the measurement model to produce a 4D regularized reconstruction.

Recently, it has been demonstrated that plug-and-play (PnP) priors [13]–[16] can dramatically improve reconstruction quality by enabling the use of state-of-the-art denoisers as prior models in MBIR. So PnP has great potential to improve reconstruction quality in 4D CT imaging problems. However, state-of-the-art denoisers such as deep convolutional neural networks (CNN) and BM4D are primarily available for 2D and sometimes 3D images, and it is difficult to extend them to higher dimensions [7], [17], [18]. In particular, extending CNNs to 4D requires very computationally and memory intensive 4D convolution applied to 5D feature tensor structures. This problem is further compounded by the lack of GPU accelerated routines for 4D convolution from major Deep-

Soumendu Majee, Thilo Balke, and Charles A. Bouman are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA (e-mail: smajee@purdue.edu, tbalke@purdue.edu, bouman@purdue.edu). Craig A.J. Kemp is with Eli Lilly and Company, Indianapolis, IN, USA (email: kemp_craig_a@lilly.com). Gregory T. Buzzard is with the Department of Mathematics, Purdue University, West Lafayette, IN, USA (email: buzzard@purdue.edu).

This work was supported by Eli Lilly and Company under research project funding agreement 17099289. Charles A. Bouman and Gregory T. Buzzard were supported in part by NSF grant CCF-1763896.

Learning frameworks such as Tensorflow, Keras, PyTorch¹. Furthermore, 4D CNNs require 4D ground truth data to train the PnP denoisers, which might be difficult or impossible to obtain.

In this paper, we present a novel 4D X-ray CT reconstruction algorithm that combines multiple low-dimensional CNN denoisers to implement a highly effective 4D prior model. Our approach, *multi-slice fusion*, integrates the multiple low-dimensional priors using multi-agent consensus equilibrium (MACE) [19]. MACE is an extension of the PnP framework that formulates the inversion problem using an equilibrium equation—as opposed to an optimization—and allows for the use of multiple prior models and agents.

Figure 1 illustrates the basic concept of our approach. Multi-slice fusion integrates together three distinct CNN denoisers each of which is trained to remove additive white Gaussian noise along lower dimensional slices (hyperplanes) of the 4D object. When MACE fuses the denoisers it *simultaneously* enforces the constraints of each denoising agent, so that the reconstructions are constrained to be smooth in all four dimensions. Consequently, multi-slice fusion results in high-quality reconstructions that are practical to train and compute even when the dimensionality of the reconstruction is high. In our implementation, one MACE agent estimates the cone-beam tomographic inversion. The remaining 3 agents are CNN denoisers trained to remove additive white Gaussian noise along two spatial directions and the time direction. The CNN agents work along complimentary spatial directions and are designed to take as input a stack of five 2D slices from five neighboring time-points. We refer to this as 2.5D denoising [7], [20]. Further details are given in Section IV.

The MACE solution can be computed using a variety of algorithms, including variants of the plug-and-play algorithm based on ADMM or other approaches [13], [14], [21], [22]. We implement multi-slice fusion on distributed heterogeneous clusters in which different agent updates are distributed onto different cluster nodes. In particular, the cone-beam inversion computations are distributed onto multiple CPU nodes and concurrently, the CNN denoising computations are distributed onto multiple GPU nodes.

We present experiments using both simulated and real data of 4D NDE tomographic imaging from sparse-views, and we compare multi-slice fusion with MBIR using total variation (TV) and 4D Markov random field (MRF) priors. Our results indicate that multi-slice fusion can substantially reduce artifacts and increase resolution relative to these alternative reconstruction methods.

The rest of the paper is organized as follows. In section II, we introduce the problem of 4D CT reconstruction. In section III, we introduce the theory behind MACE model fusion. In section IV, we use the MACE framework to introduce multi-slice fusion. In section V, we describe our training pipeline for training the CNN denoisers. In section VI, we describe our distributed implementation of multi-slice fusion on heterogeneous clusters. Finally, in section VII, we present

results on sparse-view and limited-angle 4D CT using both simulated and real data.

II. PROBLEM FORMULATION

In 4D X-ray CT imaging, a dynamic object is rotated and several 2D projections (radiographs) of the object are measured for different angles as illustrated in Figure 2. The problem is then to reconstruct the 4D array of X-ray attenuation coefficients from these measurements, where three dimensions correspond to the spatial dimensions and the fourth dimension corresponds to time.

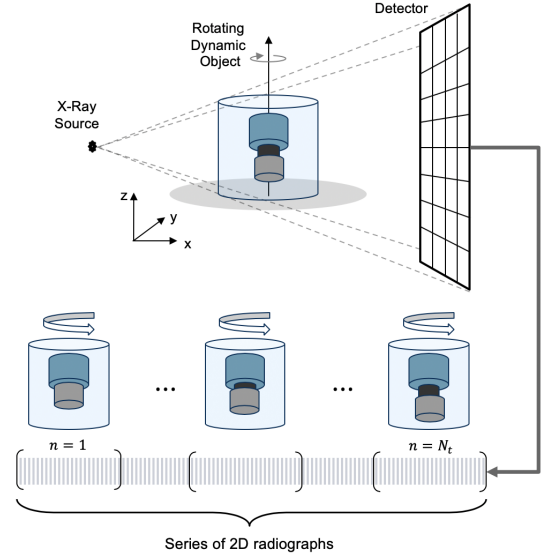


Fig. 2. Illustration of 4D cone-beam X-ray CT imaging. The dynamic object is rotated and several 2D projections (radiographs) of the object are measured for different angles. The projections are divided into N_t disjoint subsets for each of the N_t time-points.

Let N_t be the number of time-points, M_n be the number of measurements at each time-point, and N_s be the number of voxels at each time-point of the 4D volume. For each time-point $n \in \{1, \dots, N_t\}$, define $y_n \in \mathbb{R}^{M_n}$ to be the vector of sinogram measurements at time n , and $x_n \in \mathbb{R}^{N_s}$ to be the vectorized 3D volume of X-ray attenuation coefficients for that time-point. Let us stack all the measurements to form a measurement vector $y = [y_1^\top, \dots, y_{N_t}^\top]^\top \in \mathbb{R}^M$ where $M = \sum_{n=1}^{N_t} M_n$ is the total number of measurements. Similarly, let us stack the 3D volumes at each time-point to form a vectorized 4D volume $x = [x_1^\top, \dots, x_{N_t}^\top]^\top \in \mathbb{R}^N$, where $N = N_t N_s$ is the total number voxels in the 4D volume. The 4D reconstruction problem then becomes the task of recovering the 4D volume of attenuation coefficients, x , from the series of sinogram measurements, y .

In the traditional maximum a posteriori (MAP) approach, the reconstruction is given by

$$x^* = \arg \min_x \{l(x) + \beta h(x)\}, \quad (1)$$

where $l(x)$ is the data-fidelity or log-likelihood term, $h(x)$ is the 4D regularizer or prior model, and the unit-less parameter β controls the level of regularization in the reconstruction. The

¹Currently only 1D, 2D, and 3D convolutions are supported with GPU acceleration

data-fidelity term, $l(x)$, can be written in a separable fashion as

$$l(x) = \frac{1}{2} \sum_{n=1}^{N_t} \|y_n - A_n x_n\|_{\Lambda_n}^2, \quad (2)$$

where A_n is the system matrix, and Λ_n is the weight matrix for time-point n . The weight matrix accounts for the non-uniform noise variance due to a Gaussian approximation [23] of the underlying Poisson noise. The weight matrix is computed as $\Lambda_n = \text{diag}\{c \exp\{-y_n\}\}$ where the scalar c is empirically chosen [2].

If the prior model, $h(x)$, can be expressed analytically like a 4D Markov random field (MRF) as in [2], [4], then the expression in equation (1) can be minimized iteratively to reconstruct the image. However, in practice, it can be difficult to represent an advanced prior model in the form of a tractable cost function $h(x)$ that can be minimized. Consequently, PnP algorithms have been created as a method for representing prior models as denoising operations [13], [14]. More recently, PnP methods have been generalized to the multi-agent consensus equilibrium (MACE) framework as a way to integrate multiple models in a principled manner [4], [19], [24].

III. MACE MODEL FUSION

In this section, we use the multi-agent consensus equilibrium (MACE) framework to fuse the data-fidelity term and multiple denoisers; these multiple denoisers form a single prior model for reconstruction. This allows us to construct a 4D prior model using low-dimensional CNN denoisers (described in Section IV).

To introduce the concept of consensus equilibrium, let us first consider a variation of the optimization problem in equation (1) with K regularizers $h_k(x)$, $k = 1, \dots, K$. The modified optimization problem can thus be written as

$$x^* = \arg \min_x \left\{ l(x) + \frac{\beta}{K} \sum_{k=1}^K h_k(x) \right\}, \quad (3)$$

where the normalization by K is done to make the regularization strength independent of the number of regularizers.

Now we transform the optimization problem of equation (3) to an equivalent consensus equilibrium formulation. However, in order to do this, we must introduce additional notation. First, we define the proximal maps of each term in equation (3). We define $L(x) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ to be the proximal map of $l(x)$ as

$$L(x) = \arg \min_{z \in \mathbb{R}^N} \left\{ l(z) + \frac{1}{2\sigma^2} \|x - z\|_2^2 \right\}, \quad (4)$$

for some $\sigma > 0$. Similarly, we define $H_k(x) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ to be the proximal map of each $h_k(x)$, $k = 1, \dots, K$ as

$$H_k(x) = \arg \min_{z \in \mathbb{R}^N} \left\{ \frac{1}{2\sigma^2} \|x - z\|_2^2 + h_k(z) \right\}. \quad (5)$$

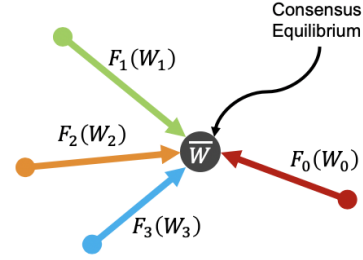


Fig. 3. Illustration of consensus equilibrium as analogous to a force balance equation: each agent pulls the solution toward its manifold and at equilibrium the forces balance each other.

Each of these proximal maps serve as agents in the MACE framework. We stack the agents together to form a stacked operator $F : \mathbb{R}^{(K+1)N} \rightarrow \mathbb{R}^{(K+1)N}$ as

$$F(W) = \begin{bmatrix} L(W_0) \\ H_1(W_1) \\ \vdots \\ H_K(W_K) \end{bmatrix}, \quad (6)$$

where $W \in \mathbb{R}^{(K+1)N}$ is stacked representative variable. The consensus equilibrium is the vector $W^* \in \mathbb{R}^{(K+1)N}$ that satisfies

$$F(W^*) = G(W^*), \quad (7)$$

where G is an averaging operator given as

$$G(W) = \begin{bmatrix} \bar{W} \\ \vdots \\ \bar{W} \end{bmatrix}, \quad (8)$$

and the weighted average is defined as

$$\bar{W} = \frac{1}{1+\beta} W_0 + \frac{\beta}{1+\beta} \left(\frac{1}{K} \sum_{k=1}^K W_k \right). \quad (9)$$

Notice the weighting scheme is chosen to balance the forward and prior models. The unitless parameter β is used to tune the weights given to the prior model and thus the regularization of the reconstruction. Equal weighing of the forward and prior models can be achieved using $\beta = 1$.

If W^* satisfies the consensus equilibrium condition of equation (7), then it can be shown [19] that \bar{W}^* is the solution to the optimization problem in equation (3). Thus if the agents in MACE are true proximal maps then the consensus equilibrium solves an equivalent optimization problem.

However, if the MACE agents are not true proximal maps, then there is no inherent optimization problem to be solved, but the MACE solution still exists. In this case, the MACE solution can be interpreted as the balance point between the forces of each agent as illustrated in Figure 3. Each agent pulls the solution toward its manifold and the consensus equilibrium solution represents a balance point between the forces of each agent. Thus MACE provides a way to incorporate non-optimization based models such as deep neural networks for solving inverse problems.

To see how we can incorporate deep neural network based prior models, first notice that equation (5) can be interpreted

as the MAP estimate for a Gaussian denoising problem with prior model h_k and noise standard deviation σ . Thus we can replace each MACE operator, H_k , for each $k = 1, \dots, K$ in equation (5) with a deep neural network trained to remove additive white Gaussian noise of standard deviation σ .

It is interesting to note that when H_k is implemented with a deep neural network denoiser, then the agent H_k is not, in general, a proximal map and there is no corresponding cost function h_k . We know this because for H_k to be a proximal map, it must satisfy the condition that $\nabla H_k(x) = [\nabla H_k(x)]^\top$ (see [13], [25]), which is equivalent to H_k being a conservative vector function (see for example [26, Theorem 2.6, p. 527]). For a CNN, ∇H_k is a function of the trained weights, and in the general case, the condition will not be met unless the CNN architecture is specifically designed to enforce such a condition.

The consensus equilibrium equation 7 states the condition that the equilibrium solution must satisfy. However, the question remains of how to compute this equilibrium solution. Our approach to solving the consensus equilibrium equations is to first find an operator that has the equilibrium solution as a fixed point, and then use standard fixed point solvers. To do this, we first notice that the averaging operator has the property that $G(G(W)) = G(W)$. Intuitively, this is true because applying averaging twice is the same as applying it once. Using this fact, we see that

$$(2G - I)(2G - I) = 4GG - 4G + I = I, \quad (10)$$

where I is the identity mapping. We then rewrite equation (7) as

$$\begin{aligned} FW^* &= GW^* \\ (2F - I)W^* &= (2G - I)W^* \\ (2G - I)(2F - I)W^* &= W^*. \end{aligned}$$

So from this we see that the following fixed point relationship must hold for the consensus equilibrium solution.

$$(2G - I)(2F - I)W^* = W^*, \quad (11)$$

and the consensus equilibrium solution W^* is a fixed point of the mapping $T = (2G - I)(2F - I)$.

We can apply a variety of iterative fixed point algorithms to equation (11) to compute the equilibrium solution. These algorithms have varying convergence guarantees and convergence speeds [19]. One such algorithm is Mann iteration [19], [24], [27]. Mann iteration performs the following pseudo-code steps until convergence where \leftarrow indicates assignment of a pseudo-code variable.

$$W \leftarrow (1 - \rho)W + \rho TW, \quad (12)$$

where weighing parameter $\rho \in (0, 1)$ is used to control the speed of convergence. In particular, when $\rho = 0.5$, the Mann-iteration solver is equivalent to the consensus-ADMM algorithm [19], [28]. It can be shown that the Mann iteration converges to a fixed point of $T = (2G - I)(2F - I)$ if T is a non-expansive mapping [19].

Note that each Mann iteration update in equation (12) involves performing the minimization in equation (4). This

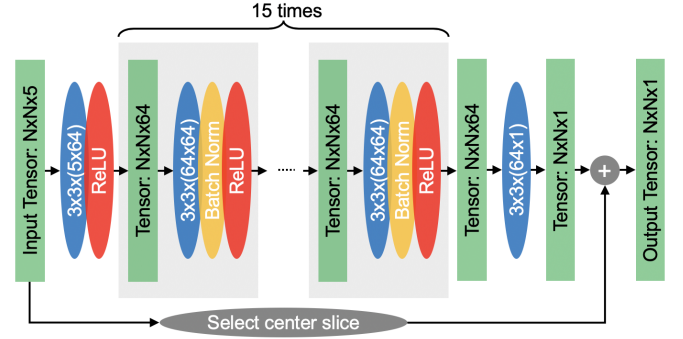


Fig. 4. Architecture of our 2.5D CNN denoiser. Different sizes of input and output necessitate a selection operator for the residual connection. Each green rectangle denotes a tensor, and each ellipse denotes an operation. Blue ellipses specify the shape of the convolution kernel.

nested iteration is computationally expensive and leads to slow convergence. Instead of minimizing equation (4) till convergence, we initialize with the result of the previous Mann iteration and perform only three iterations of iterative coordinate descent (ICD). We denote this partial update operator as $\tilde{L}(W_0, X_0)$ where X_0 is the initial condition to the iterative update. The corresponding new F operator approximation is then given by

$$\tilde{F}(W; X) = \begin{bmatrix} \tilde{L}(W_0; X_0) \\ H_1(W_1) \\ \vdots \\ H_k(W_K) \end{bmatrix}. \quad (13)$$

Algorithm 1 shows a simplified Mann iteration using partial updates. We perform algebraic manipulation of the traditional Mann iterations [24], [27] in order to obtain the simplified but equivalent Algorithm 1. It can be shown that partial update Mann iteration also converges [24], [27] to the fixed point in equation (11). We used a zero initialization, $x^{(0)} = 0$, in all our experiments and continue the partial update Mann iteration until the differences between state vectors X_k become smaller than a fixed threshold.

Algorithm 1: Partial update Mann iteration for computing the MACE solution

Input: Initial Reconstruction: $x^{(0)} \in \mathbb{R}^N$

Output: Final Reconstruction: x^*

- 1 $X \leftarrow W \leftarrow \begin{bmatrix} x^{(0)} \\ \vdots \\ x^{(0)} \end{bmatrix}$
 - 2 **while not converged do**
 - 3 $X \leftarrow \tilde{F}(W; X)$
 - 4 $Z \leftarrow G(2X - W)$
 - 5 $W \leftarrow W + 2\rho(Z - X)$
 - 6 $x^* \leftarrow X_0$
-

IV. MULTI-SLICE FUSION USING MACE

We use four MACE agents to implement multi-slice fusion. We set $K = 3$ and use the names $H_{xy,t}$, $H_{yz,t}$, $H_{zx,t}$ to denote

the denoising agents H_1, H_2, H_3 in equation (6). The agent L enforces fidelity to the measurement while each of the denoisers $H_{xy,t}, H_{yz,t}, H_{zx,t}$ enforces regularity of the image in orthogonal image planes. MACE imposes a consensus between the operators $L, H_{xy,t}, H_{yz,t}, H_{zx,t}$ to achieve a balanced reconstruction that lies at the intersection of the solution space of the measurement model and each of the prior models. The MACE stacked operator F encompassing all four agents can be written as

$$F(W) = \begin{bmatrix} L(W_0) \\ H_{xy,t}(W_1) \\ H_{yz,t}(W_2) \\ H_{zx,t}(W_3) \end{bmatrix}. \quad (14)$$

Here the representative variable $W \in \mathbb{R}^{4N}$ is formed by stacking four vectorized 4D volumes.

The three denoisers $H_{xy,t}, H_{yz,t}$, and $H_{zx,t}$ share the same architecture and trained model but are applied along different planes of the 4D space. The CNN architecture is shown in Figure 4. We have modified a typical CNN architecture [29] to input information from a third dimension. The channel dimension of a convolution layer is typically used to input multiple color channels for denoising 2D color images using CNNs. We re-purpose the channel dimension to input five adjacent 2D slices of the noisy image to the network and output the denoised center slice. The other slices are being denoised by shifting the 5-slice moving window. We call this 2.5D since the receptive field along the convolution dimensions is large but in the channel dimension is small. It has been shown that this type of 2.5D processing is a computationally efficient way of performing effective 3D denoising with CNNs [7], [20]. We use the notation $H_{xy,t}$ to denote a CNN space-time denoiser that performs convolution in the xy-plane and uses the convolution channels to input slices from neighboring time-points. The denoisers $H_{yz,t}$ and $H_{zx,t}$ are analogous to $H_{xy,t}$ but are applied along the yz and zx-plane, respectively. This orientation of the three denoisers ensures that

- 1) The spatial dimensions x, y, z are treated equivalently. This ensures the regularization to be uniform across all spatial dimensions;
- 2) Each dimension in x, y, z, and t is considered at least once. This ensures that model fusion using MACE incorporates information along all four dimensions.

Since the three denoising operators $H_{xy,t}, H_{yz,t}$, and $H_{zx,t}$ process the 4D volume “slice by slice”, they can be implemented in parallel on large scale parallel computers. Details on distributed implementation are described in section VI.

V. TRAINING OF CNN DENOISERS

All three prior model agents $H_{xy,t}, H_{yz,t}$, and $H_{zx,t}$ in multi-slice fusion share the same 2.5D model shown in Figure 4 but are oriented along different planes. Consequently we train a single 2.5D CNN model using 3D data. Even though the CNN needs to denoise 3D time-space data, we train it using 3D spatial data since 3D volumes are widely available unlike time-space data.

Figure 5 outlines our training data generation. We start with a low-noise 3D CT volume that is representative of the

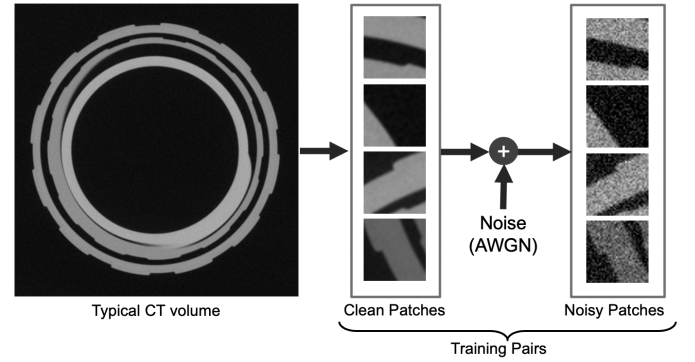


Fig. 5. Illustration of our training data generation. We extract 3D patches from a typical CT volume and add additive white Gaussian noise (AWGN) to generate training pairs. This makes the training process self-supervised.

objects to be reconstructed. We extract 3D patches from the CT volume and add pseudo-random additive white Gaussian noise (AWGN) to the patches to generate the training pairs. We then train the CNN to remove the noise. The use of AWGN is due to the mathematical form of the quadratic norm term in the proximal map in equation 5 and follows from the theory of Plug-and-play [13], [14].

VI. DISTRIBUTED RECONSTRUCTION

The computational structure of multi-slice fusion is well-suited to a highly distributed implementation. The main computational bottleneck in Algorithm 1 is the F operator. Fortunately, F is a parallel operator and thus its individual components $L, H_{xy,t}, H_{yz,t}$, and $H_{zx,t}$ can be executed in parallel. The operators $L, H_{xy,t}, H_{yz,t}$, and $H_{zx,t}$ can themselves be parallelized internally as well. The distributed implementation of multi-slice fusion is illustrated in Figure 6.

The CNN denoisers $H_{xy,t}, H_{yz,t}$, and $H_{zx,t}$ are 2.5D denoisers that denoise the 4D volume by processing it slice by slice and thus can be trivially parallelized leading to a large number of concurrent operations. The concurrent operations for all three denoisers are distributed among multiple GPUs due to the availability of optimized GPU routines in Tensorflow. In our experiments we used a GPU cluster with three Nvidia Tesla P100 GPUs to compute the CNN denoising operators.

The cone-beam inversion operator, L , can also be computed for each time-point independently due to the separable structure in equations (4) and (2). This leads to a large number of concurrent operations which are distributed among multiple CPU nodes. The cone-beam inversion for each time-point is computed using a coordinate-descent minimization with multi-threaded parallelism. Further details about the cone-beam inversion can be found in [3].

VII. EXPERIMENTAL RESULTS

We present experimental results on two simulated and two real 4D X-ray CT data for Non-Destructive Evaluation (NDE) applications to demonstrate the improved reconstruction quality of our method. The four experimental cases are outlined below

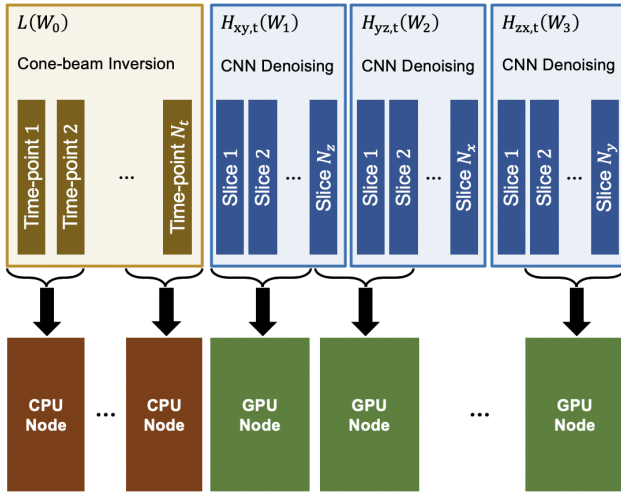


Fig. 6. Illustration of distributed computation of multi-slice fusion. We perform distributed computation of the F operator which is the main computational bottleneck in Algorithm 1. Each operator within F , namely $H_{xy,t}$, $H_{yz,t}$, $H_{zx,t}$, and L can be executed in parallel. Furthermore, operators $H_{xy,t}$, $H_{yz,t}$, $H_{zx,t}$, and L are 3D operators that can process the 4D volume “slice by slice” leading to a large number of concurrent operations that can be distributed among multiple compute nodes.

- 1) **Simulated Data 360°**: Sparse-view results on simulated data with a set of sparse views ranging over 360° at each reconstructed time-point;
- 2) **Simulated Data 90°**: Sparse-view limited-angle results on simulated data with a set of sparse views ranging over 90° at each reconstructed time-point;
- 3) **Real Data 360°**: Sparse-view results on real data with a set of sparse views ranging over 360° at each reconstructed time-point;
- 4) **Real Data 90°**: Sparse-view limited-angle results on real data with a set of sparse views ranging over 90° at each reconstructed time-point.

The selection of the rotation range per time-point is arbitrary and can be chosen after the measurements have been taken. For example, a full rotation with 400 views can be used as a single time-point or as four time-points with 100 views each. The four time-points per rotation can provide extra temporal resolution, however, they require a more difficult reconstruction with incomplete information.

We compare multi-slice fusion with several other methods outlined below

- FBP: Conventional 3D filtered back projection reconstruction;
- MBIR+TV: MBIR reconstruction using a total variation (TV) prior [30] in the spatial dimensions;
- MBIR+4D-MRF: MBIR reconstruction using 4D Markov random field prior [2] with $q = 2.2$, $p = 1.1$, 26 spatial neighbors and 2 temporal neighbors;
- MBIR+ $H_{xy,t}$: MBIR using the CNN $H_{xy,t}$ as a PnP prior;
- MBIR+ $H_{yz,t}$: MBIR using the CNN $H_{yz,t}$ as a PnP prior;
- MBIR+ $H_{zx,t}$: MBIR using the CNN $H_{zx,t}$ as a PnP prior.

We used two CPU cluster nodes, each with 20 Kaby Lake CPU cores and 96 GB system memory to compute the cone-

beam inversion. We used three GPU nodes, each with a Nvidia Tesla P100 GPU (16 GB GPU-memory) and 192 GB system memory to compute the CNN denoisers. To compute the multi-slice fusion reconstruction, we run Algorithm 1 for 10 Mann iterations, with 3 iterations of cone-beam inversion per Mann iteration. The total reconstruction time of multi-slice fusion for each experimental case are given in Table I.

Experimental Case	Reconstruction size (x,y,z,t)	Reconstruction Time
Simulated Data 360°	$240 \times 240 \times 28 \times 8$	8 mins
Simulated Data 90°	$240 \times 240 \times 28 \times 8$	8 mins
Real Data 360°	$731 \times 731 \times 91 \times 16$	133 mins
Real Data 90°	$263 \times 263 \times 778 \times 12$	46 mins

TABLE I
TOTAL RECONSTRUCTION TIME OF MULTI-SLICE FUSION FOR EACH EXPERIMENTAL CASE

The 2.5D CNN denoiser model used in the reconstructions was trained using a low-noise 3D CT reconstruction of a bottle and screw cap made from different plastics. The object is representative of a variety of Non-Destructive Evaluation (NDE) problems in which the objects to be imaged are constructed from a relatively small number of distinct materials. The extracted patches were normalized to $[0, 1]$ and random rotation, mirroring, intensity shift were applied. The standard deviation of the additive white Gaussian noise added during training was 0.1.

A. Simulated Data 360°

Magnification	5.57
Number of Views per Time-point	75
Rotation per Time-point	360°
Cropped Detector Array	240×28 , (0.95 mm) ²
Voxel Size	(0.17 mm) ³
Reconstruction Size (x,y,z,t)	$240 \times 240 \times 28 \times 8$

TABLE II
EXPERIMENTAL SPECIFICATIONS FOR SIMULATED DATA 360°

In this section we present results on simulated data to evaluate our method in a sparse-view setting. Each time-point is reconstructed from a sparse set of views spanning 360°. We take a low-noise CT reconstruction of a bottle and screw cap and denoise it further using BM4D [18] to generate a clean 3D volume to be used as a 3D phantom. We then vertically translate the 3D phantom by one pixel per time-point to generate a 4D phantom x^0 . We generate simulated sinogram measurements as $\mathcal{N}(Ax^0, \Lambda^{-1})$ where A is the projection matrix and the inverse covariance matrix $\Lambda = \text{diag}\{c \exp\{-Ax^0\}\}$ accounts for the non-uniform noise variance due to a Gaussian approximation [23] of the underlying Poisson noise. We then perform a 4D reconstruction from the simulated sinogram data and compare with the 4D phantom. The experimental specifications are summarized in Table II.

Figure 7 compares reconstructions using multi-slice fusion with several other methods. Each image is a slice through the

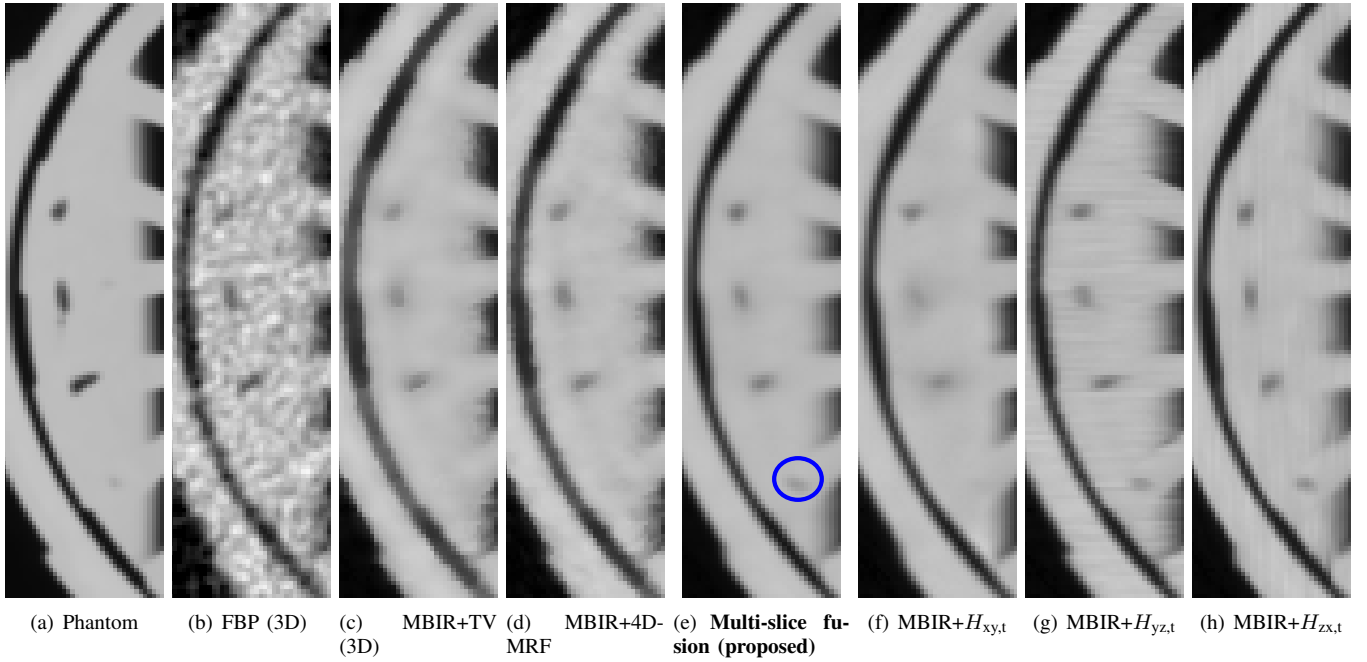
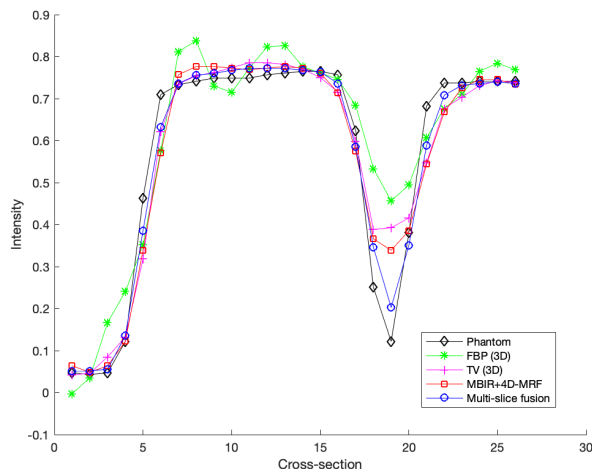
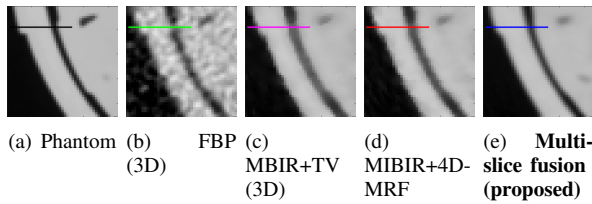


Fig. 7. Comparison of different methods for simulated data 360° . Each image is a slice through the reconstructed object for one time-point along the spatial xy-plane. The reconstruction using FBP suffers from high noise and fails to recover the small hole in the bottom of the image. MBIR+TV and MBIR+4D-MRF suffer from jagged edges and fail to recover the small hole in the bottom of the image. MBIR+ $H_{xy,t}$ and MBIR+ $H_{zx,t}$ suffer from horizontal and vertical streaks, respectively, since the denoisers were applied in those planes. MBIR+ $H_{yz,t}$ cannot reconstruct the small hole in the bottom of the image since the xy-plane does not contain sufficient information.



(f) Plot of cross-section

Fig. 8. Plot of cross-section through the phantom and reconstructions from simulated data 360° . Multi-slice fusion results in the most accurate reconstruction of the gap between materials.

reconstructed object for one time-point along the spatial xy-plane. The reconstruction using FBP suffers from high noise and fails to recover the small hole in the bottom of the image. The reconstructions using MBIR+TV and MBIR+4D-MRF suffer from jagged edges and fail to recover the small hole in the bottom of the image. MBIR+ $H_{yz,t}$ and MBIR+ $H_{zx,t}$ suffer from horizontal and vertical streaks, respectively, since the denoisers were applied in those planes. MBIR+ $H_{xy,t}$ does not suffer from streaks in the figure since we are viewing a slice along the xy-plane, but it suffers from other artifacts. MBIR+ $H_{xy,t}$ cannot reconstruct the small hole in the bottom of the image since the xy-plane does not contain sufficient information. It is to be noted that multi-slice fusion enhances the size and contrast of the small hole highlighted by the blue circle relative to the phantom. This can cause deviations when measuring the size of small features in the reconstruction.

Next we plot a cross-section through the object for multi-slice fusion, MBIR+4D-MRF, MBIR+TV, FBP, and the phantom in Figure 8. Multi-slice fusion results in the most accurate reconstruction of the gap between materials.

Finally we report the peak signal to noise ratio (PSNR) and the structural similarity index measure (SSIM) [31] with respect to the phantom for each method in Table III to objectively measure image quality. We define the PSNR for a given 4D reconstruction x with a phantom x^0 as

$$\text{PSNR}(x) = 20 \log_{10} \left(\frac{\text{Range}(x^0)}{\text{RMSE}(x, x^0)} \right), \quad (15)$$

where range is computed from the 0.1st and 99.9th percentiles of the phantom. As can be seen from Table III, multi-slice fusion results in the highest PSNR and SSIM scores.

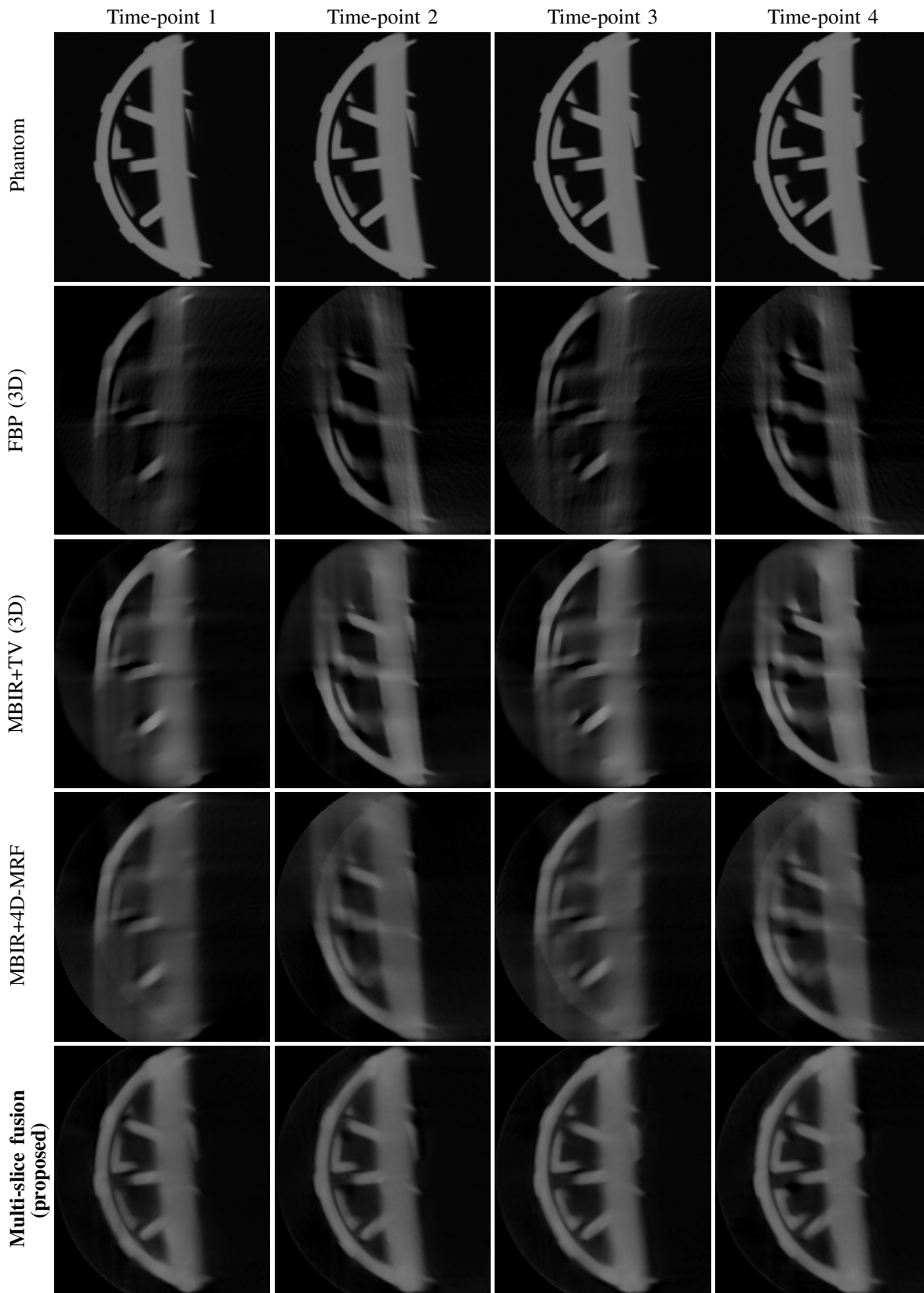


Fig. 9. Comparison of different methods for simulated data with 90° rotation of object per time-point. The FBP reconstruction has severe limited-angle artifacts. MBIR+TV improves the reconstruction in some regions but it suffers in areas affected by limited angular information. MBIR+4D-MRF reduces limited-angle artifacts, but allows severe artifacts to form that are not necessarily consistent with real 4D image sequences. In contrast, the multi-slice fusion result does not suffer from major limited-angle artifacts.

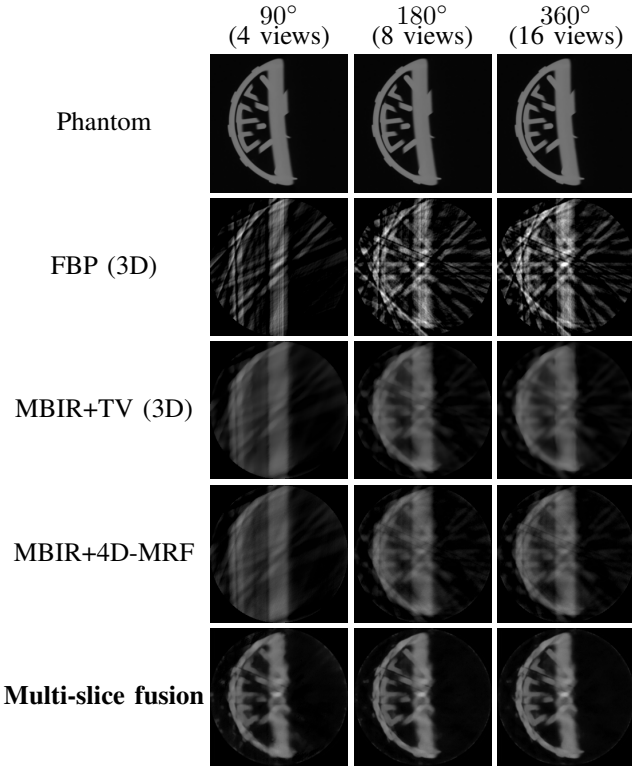
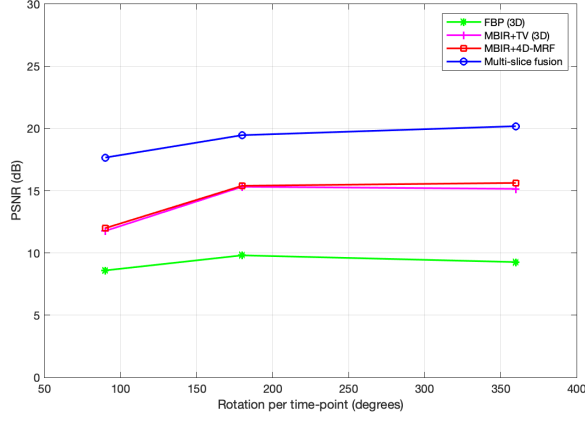


Fig. 10. Illustration of the reconstruction quality obtained for extreme sparse-view data with different levels of limited angle per time-point. FBP results in strong artifacts due to sparse-views and limited angles. MBIR+TV and MBIR+4D-MRF mitigates most of the major sparse-view artifacts but suffers from limited angle artifacts in the 90° limited angle case. Multi-slice fusion results in fewer limited-angle and sparse-view artifacts and an improved PSNR metric. Moreover, multi-slice fusion results in reduced artifacts compared to MBIR+TV and MBIR+4D-MRF as the rotation per time point is decreased.

B. Simulated Data 90°

In this section we present results on simulated data to evaluate our method in a sparse-view and limited-angle setting. Each time-point is reconstructed from a sparse set of views spanning 90°. The simulated measurement data is generated in a similar fashion as Section VII-A using the experimental specifications summarized in Table IV.

Figure 9 shows a comparison of different methods for simulated data with 90° rotation of object per time-point. The FBP reconstruction has severe limited-angle artifacts.

Method	PSNR(dB)	SSIM
FBP	19.69	0.609
MBIR+TV	26.63	0.860
MBIR+4D-MRF	25.84	0.787
Multi-slice fusion	29.07	0.943
MBIR+ $H_{xy,t}$	29.03	0.922
MBIR+ $H_{yz,t}$	28.04	0.932
MBIR+ $H_{zx,t}$	28.31	0.926

TABLE III
QUANTITATIVE EVALUATION FOR SIMULATED DATA 360°. MULTI-SLICE FUSION HAS THE HIGHEST PSNR AND SSIM METRIC AMONG ALL THE METHODS.

Magnification	5.57
Number of Views per Time-point	36
Rotation per Time-point	90°
Cropped Detector Array	240 × 28 , (0.95 mm) ²
Voxel Size	(0.17 mm) ³
Reconstruction Size (x,y,z,t)	240 × 240 × 28 × 8

TABLE IV
EXPERIMENTAL SPECIFICATIONS FOR SIMULATED DATA 90°

MBIR+TV improves the reconstruction in some regions but it suffers in areas affected by limited angular information. MBIR+4D-MRF reduces limited-angle artifacts, but allows severe artifacts to form that are not necessarily consistent with real 4D image sequences. In contrast, the multi-slice fusion result does not suffer from major limited-angle artifacts.

Table V shows peak signal to noise ratio (PSNR) and structural similarity index measure (SSIM) with respect to the phantom for each method. Multi-slice fusion results in the highest PSNR and SSIM scores.

Method	PSNR(dB)	SSIM
FBP	10.86	0.467
MBIR+TV	15.35	0.801
MBIR+4D-MRF	14.25	0.742
Multi-slice fusion	19.44	0.875

TABLE V
QUANTITATIVE EVALUATION FOR SIMULATED DATA 90°. MULTI-SLICE FUSION HAS THE HIGHEST PSNR AND SSIM METRIC AMONG ALL THE METHODS.

In order to determine the effectiveness of our method for more challenging data, we generate extreme sparse-view simulated data with different angle of rotation per time-point while keeping the rest of the experimental setup the same as Table IV. Figure 10 illustrates the reconstruction quality obtained for the extreme sparse-view data with different levels of limited angle. FBP results in strong artifacts due to sparse-views and limited angles. MBIR+TV and MBIR+4D-MRF mitigates most of the major sparse-view artifacts but suffers from limited angle artifacts in the 90° limited angle case. Multi-slice fusion results in fewer limited-angle and sparse-view artifacts and an improved PSNR metric. Moreover, multi-slice fusion results in a reduced motion and sparse view artifacts as compared to MBIR+TV and MBIR+4D-MRF as the rotation per time point is decreased.

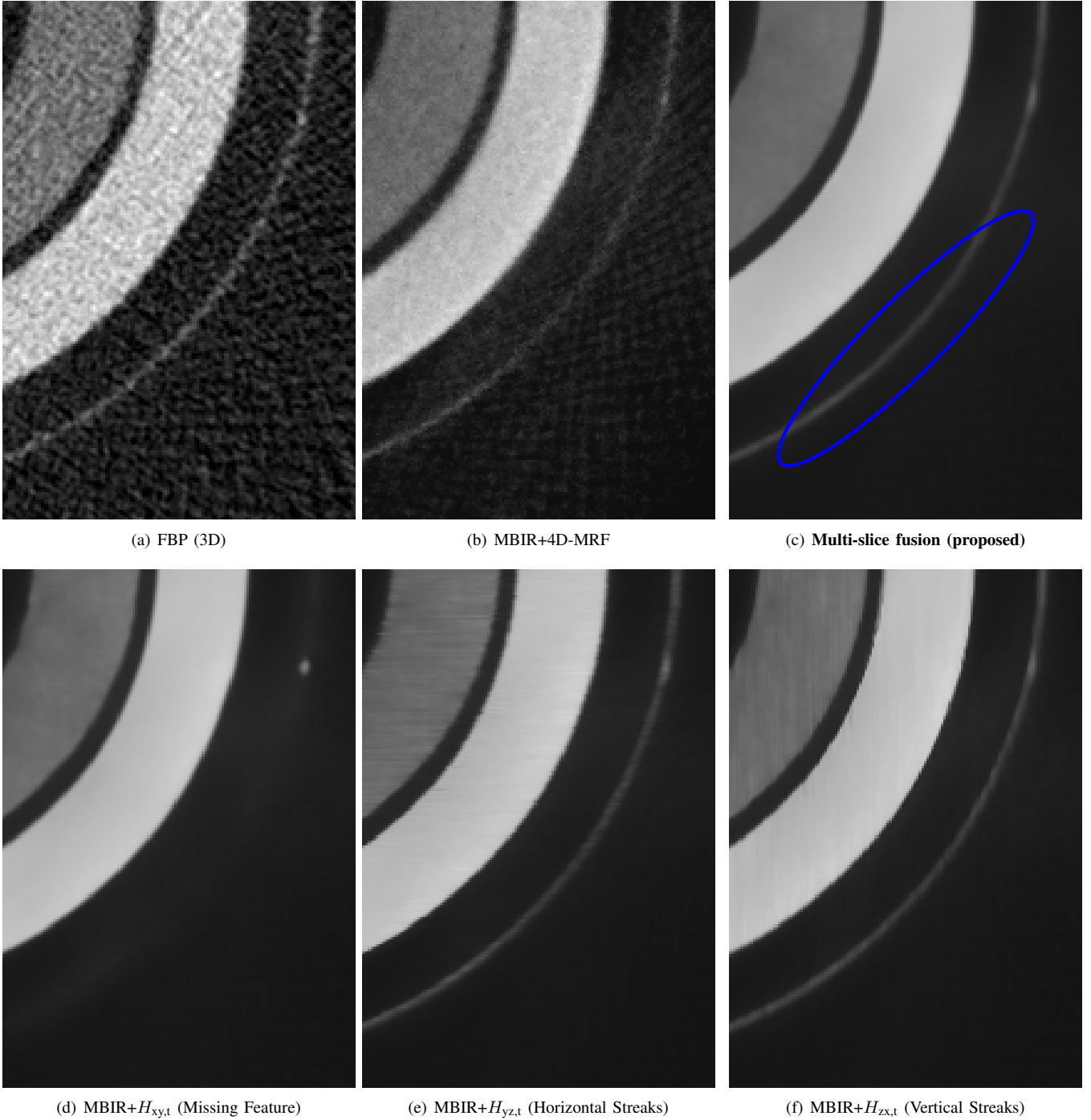


Fig. 11. Comparison of different methods for Real Data 360°: vial. Each image is a slice through the reconstructed vial for one time-point along the spatial xy-plane. Both FBP and MBIR+4D-MRF suffer from obvious windmill artifacts, higher noise and blurred edges. In contrast to that, the multi-slice fusion reconstruction has smooth and uniform textures while preserving edge definition. MBIR+ $H_{yz,t}$ and MBIR+ $H_{zx,t}$ suffer from horizontal and vertical streaks. MBIR+ $H_{xy,t}$ cannot reconstruct the outer ring since the slice displayed is at the edge of the aluminum seal and the xy-plane does not contain sufficient information. Multi-slice fusion can resolve the edges of the rings better than either of MBIR+ $H_{xy,t}$, MBIR+ $H_{yz,t}$, and MBIR+ $H_{zx,t}$ since it has information from all the spatial coordinates.

C. Real Data 360°: Vial Compression

In this section we present results on real data to evaluate our method in a sparse-view setting. The data is from a dynamic cone-beam X-ray scan of a glass vial, with elastomeric stopper and aluminum crimp-seal, using a North Star Imaging X50 X-ray CT system. The experimental specifications are

summarized in Table VI.

The vial is undergoing dynamic compression during the scan, to capture the mechanical response of the components as shown in Figure 15. Of particular interest is the moment when the aluminum seal is no longer in contact with the underside of the glass neck finish. This indicates the moment when the

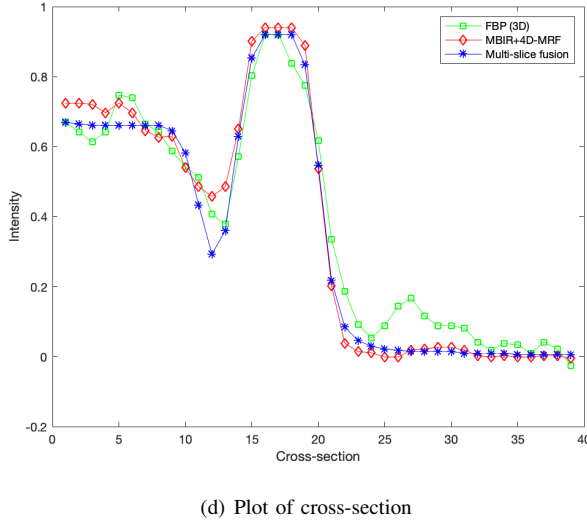
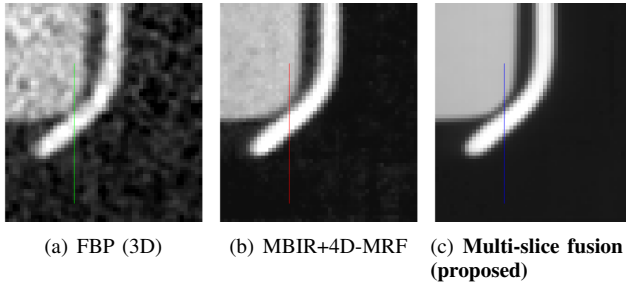


Fig. 12. Plot of cross-section through the vial at a time when the aluminum and glass have physically separated. Multi-slice fusion is able to resolve the junction between materials better while simultaneously producing a smoother reconstruction within materials compared to MBIR+4D-MRF and FBP.

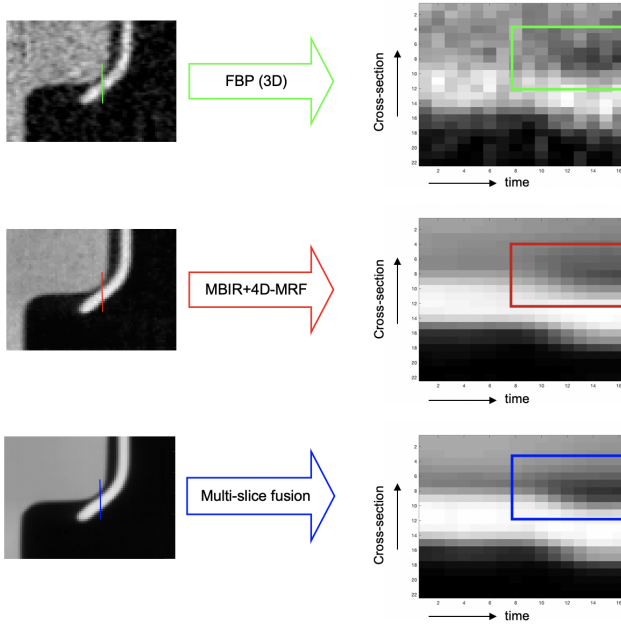


Fig. 13. Illustration of temporal resolution for real data 360° : vial. We plot a cross-section through the vial with time for each method: multi-slice fusion, MBIR+4D-MRF, FBP. Multi-slice fusion results in improved space-time resolution of the separation of aluminum and glass.

Scanner Model	North Star Imaging X50
Voltage	140 kV
Current	500 μ A
Exposure	20 ms
Source-Detector Distance	839 mm
Magnification	5.57
Number of Views per Time-point	150
Rotation per Time-point	360°
Cropped Detector Array	731 \times 91, (0.25 mm) ²
Voxel Size	(0.0456 mm) ³
Reconstruction Size (x,y,z,t)	731 \times 731 \times 91 \times 16

TABLE VI
EXPERIMENTAL SPECIFICATIONS FOR REAL DATA 360°: VIAL COMPRESSION

force applied exceeds that exerted by the rubber on the glass; this is known as the “residual seal force” [32].

During the scan, the vial was held in place by fixtures that were placed out of the field of view as shown in Figure 15. As the object rotated, the fixtures periodically intercepted the path of the X-rays resulting in corrupted measurements and consequently artifacts in the reconstruction. To mitigate this problem, we incorporate additional corrections that are described in Appendix A.

Figure 11 compares multi-slice fusion with several other methods. Each image is a slice through the reconstructed vial for one time-point along the spatial xy-plane. Both FBP and MBIR+4D-MRF suffer from obvious artifacts, higher noise and blurred edges. In contrast to that, the multi-slice fusion reconstruction has smooth and uniform textures while preserving edge definition. Figure 11 also illustrates the effect of model fusion by comparing multi-slice fusion with MBIR+ $H_{xy,t}$, MBIR+ $H_{yz,t}$, and MBIR+ $H_{zx,t}$. MBIR+ $H_{yz,t}$ and MBIR+ $H_{zx,t}$ suffer from horizontal and vertical streaks respectively since the denoisers were applied in those planes. MBIR+ $H_{xy,t}$ does not suffer from streaks in the figure since we are viewing a slice along the xy-plane, but it suffers from other artifacts. MBIR+ $H_{xy,t}$ cannot reconstruct the outer ring since the slice displayed is at the edge of the aluminum seal and the xy-plane does not contain sufficient information. In contrast, multi-slice fusion can resolve the edges of the rings better than either of MBIR+ $H_{xy,t}$, MBIR+ $H_{yz,t}$, and MBIR+ $H_{zx,t}$ since it uses information from all the spatial coordinates.

Next, we plot a cross-section through the object for multi-slice fusion, MBIR+4D-MRF and FBP in Figure 12. For this, we choose a time-point where we know the aluminum and glass have separated spatially, thus creating an air-gap. Multi-slice fusion results in a deeper and more defined reconstruction of the gap between materials. This supports that multi-slice fusion is able to preserve fine details in spite of producing a smooth regularized image.

Finally in Figure 13 we plot a cross-section through the object with respect to time to show the improved space-time resolution of our method. We do this for FBP, MBIR+4D-MRF and multi-slice fusion. Multi-slice fusion results in improved space-time resolution of the separation of aluminum and glass.

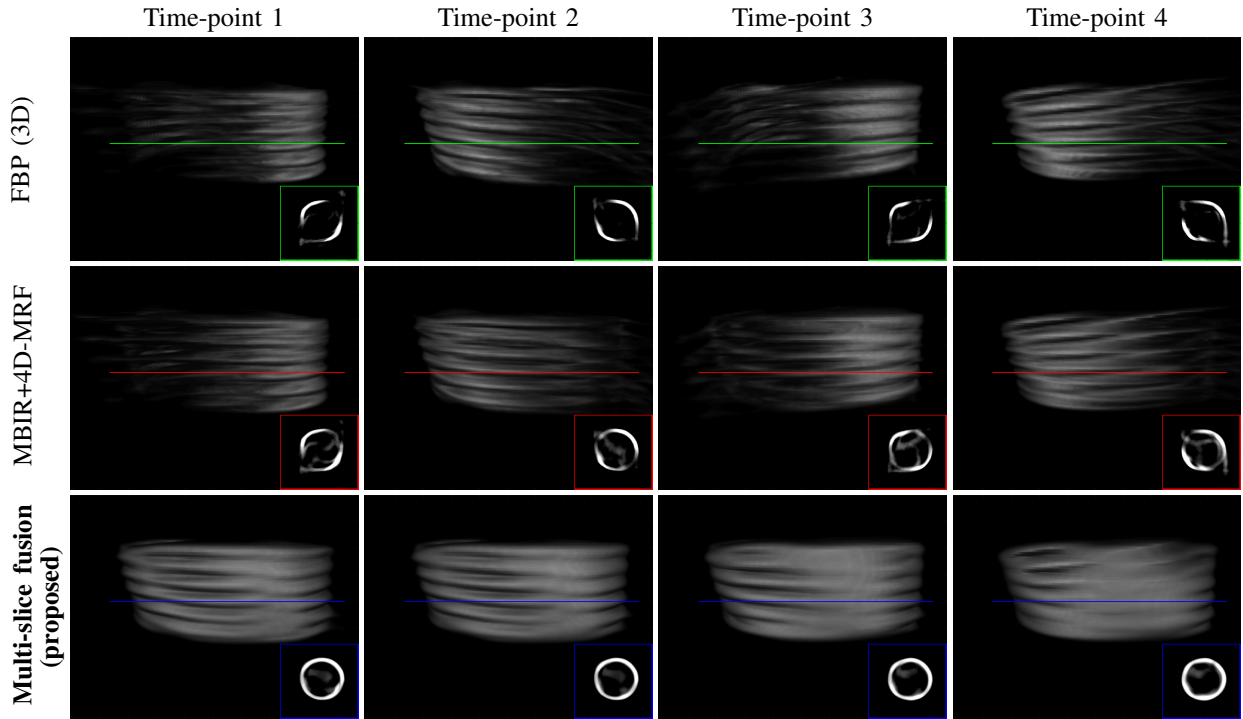


Fig. 14. Volume rendering of the reconstructed spring and its cross-section for four time-points. A 90° limited set of views is used to reconstruct each time-point. The FBP reconstruction contains severe limited-angle artifacts. MBIR+4D-MRF mitigates some limited-angle artifacts but some artifacts remain.

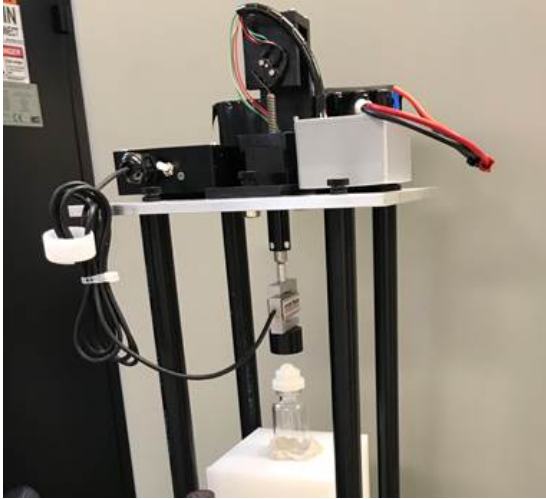


Fig. 15. Experimental setup for Real Data 360° : Vial Compression. The vial is undergoing dynamic compression during the scan, to capture the mechanical response of the components. The glass vial (center) and the actuator (top) are held together by a frame constructed of tubes and plates. The tubes were placed outside the field of view of the CT scanner, thus causing artifacts in the reconstruction. We describe a correction for this in Appendix A.

D. Real Data 90° : Injector Pen

In this section we present results on real data to evaluate our method in a sparse-view and limited-angle setting. The data is from a dynamic cone-beam X-ray scan of an injector pen using a North Star Imaging X50 X-ray CT system. The experimental specifications are summarized in Table VII.

The injection device is initiated before the dynamic scan starts and completes a full injection during the duration of the scan. We are interested in observing the motion of a particular

Scanner Model	North Star Imaging X50
Voltage	165 kV
Current	550 μ A
Exposure	12.5 ms
Source-Detector Distance	694 mm
Magnification	2.83
Number of Views per Time-point	36
Rotation per Time-point	90°
Cropped Detector Array	263×768 , $(0.254 \text{ mm})^2$
Voxel Size	$(0.089 \text{ mm})^3$
Reconstruction Size (x,y,z,t)	$263 \times 263 \times 778 \times 12$

TABLE VII
EXPERIMENTAL SPECIFICATIONS FOR REAL DATA 90° : INJECTOR PEN

spring within the injector pen in order to determine whether it is working as expected. The spring in question is a non-helical wave-spring [33] that is constructed out of circular rings that are joined together. The spring exhibits a fast motion and as a result we need a high temporal resolution to observe the motion of the spring. To have sufficient temporal resolution we reconstruct one frame for every 90° rotation of the object instead of the conventional 360° rotation.

Figure 14 shows a volume rendering of the reconstructed spring and a cross-section through it for four time-points and reconstruction methods FBP, MBIR+4D-MRF, and multi-slice fusion. The FBP reconstruction contains severe limited-angle artifacts. MBIR+4D-MRF mitigates some limited-angle artifacts but some artifacts remain. In contrast, multi-slice fusion mitigates most limited-angle artifacts. The cross-sections of the spring in the multi-slice fusion reconstruction are more circular than the other methods, which align with our prior

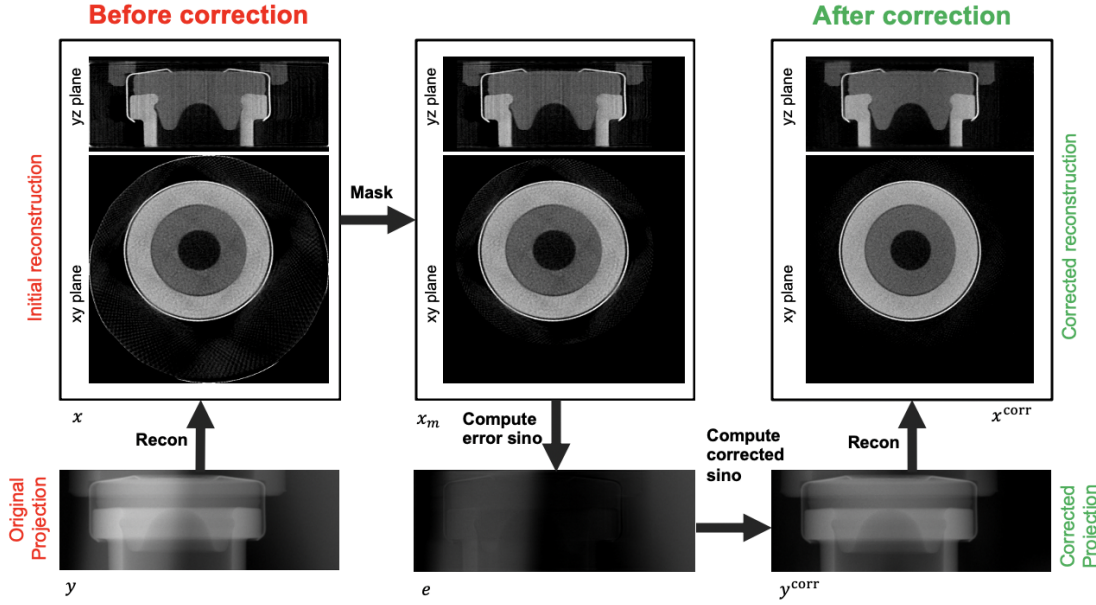


Fig. 16. Pipeline of the blind fixture correction in Algorithm 2. The vertical stripes in the yz-plane of the reconstruction and the ring at the edge of the field of view in the xy-plane of the reconstruction have been rectified after performing the correction.

knowledge about the spring. The fast compression of the spring causes the rings within the spring to move significantly within a time-point, resulting in the observed blur in the multi-slice fusion reconstruction. Strong limited-angle artifacts in the other reconstructions mask this effect.

VIII. CONCLUSION

In this paper, we proposed a novel 4D X-ray CT reconstruction algorithm, multi-slice fusion, that combines multiple low-dimensional denoisers to form a 4D prior. Our method allows the formation of an advanced 4D prior using state-of-the-art CNN denoisers without needing to train on 4D data. Furthermore, it allows for multiple levels of parallelism, thus enabling reconstruction of large volumes in a reasonable time. Although we focused on 4D X-ray CT reconstruction for NDE applications, our method can be used for any reconstruction problem involving multiple dimensions.

ACKNOWLEDGMENT

The authors would like to acknowledge support from Eli Lilly and Company under research project funding agreement 17099289. Charles A. Bouman and Gregory T. Buzzard were supported in part by NSF grant CCF-1763896. We also thank M. Cory Victor and Dr. Coralie Richard from Eli Lilly and Company for their assistance and guidance in setting up the residual seal force test experiment.

APPENDIX A

CORRECTION FOR FIXTURES OUTSIDE THE FIELD OF VIEW

Here we describe our correction for fixtures placed out of the field of view of the scanner. As shown in Figure 15, the setup is held together by a fixture constructed of tubes and plates. The tubes were placed outside the field of view of the CT scanner, thus causing artifacts in the reconstruction. Our

Algorithm 2: Blind fixture correction

Inputs: Original Sinogram: y ,
System Matrix: A ,
Output: Corrected Sinogram: y^{corr}

- 1 $x \leftarrow \text{recon}(y, A)$
- 2 $x^m \leftarrow \text{mask}(x)$
- 3 $e \leftarrow y - Ax^m$
- 4 $p \leftarrow \text{blur}(e)$
- 5 $c \leftarrow \arg \min_{c \in \mathbb{R}} \|e - cp\|^2$
- 6 $y^{\text{corr}} \leftarrow y - cp$

method performs a blind source separation of the projection of the object from that of the tubes. Our blind separation relies on the fact that the projection of the tubes is spatially smooth. This is true since the tubes themselves do not have sharp features and there is motion blur due to the large distance of the tubes from the rotation axis.

Algorithm 2 shows our correction algorithm for the fixtures. Figure 16 illustrates the algorithm pictorially. The initial reconstruction x suffers from artifacts within the image and at the edge of the field of view. We mask x using a cylindrical mask slightly smaller than the field of view to obtain the masked image x^m . This is done so that the majority of the artifacts at the edge of the field of view are masked but the object remains unchanged in x^m . Consequently the error sinogram $e = y - Ax^m$ primarily contains the projection of the tubes with some residual projection of the object. The blurring of e filters out the residual object projection but preserves the spatially smooth projection of the tubes. The corrected measurements y^{corr} are found after performing a least squares fit. The correction can be repeated in order to get an improved reconstruction x and consequently an improved correction y^{corr} .

Figure 16 shows the sinogram and reconstruction both before and after performing the blind correction. Not only does the reconstruction after fixture correction remove the artifacts in the air region, but it also improves the image quality inside the object. It can be seen that the vertical stripes in the object in the yz view of the reconstruction have been eliminated after performing the correction.

REFERENCES

- [1] C. Huang, J. L. Ackerman, Y. Petibon, T. J. Brady, G. El Fakhri, and J. Ouyang, "MR-based motion correction for PET imaging using wired active MR microcoils in simultaneous PET-MR: Phantom study," *Medical physics*, vol. 41, no. 4, 2014.
- [2] K. A. Mohan, S. Venkatakrishnan, J. W. Gibbs, E. B. Gulsoy, X. Xiao, M. De Graef, P. W. Voorhees, and C. A. Bouman, "TIMBIR: A method for time-space reconstruction from interlaced views," *IEEE Trans. Computational Imaging*, vol. 1, no. 2, pp. 96–111, 2015.
- [3] T. Balke, S. Majee, G. T. Buzzard, S. Poveromo, P. Howard, M. A. Groeber, J. McClure, and C. A. Bouman, "Separable models for cone-beam MBIR reconstruction," *Electronic Imaging*, vol. 2018, no. 15, pp. 181–1, 2018.
- [4] S. Majee, T. Balke, C. A. Kemp, G. T. Buzzard, and C. A. Bouman, "4D X-ray CT reconstruction using Multi-Slice Fusion," in *2019 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2019, pp. 1–8.
- [5] Z. Nadir, M. S. Brown, M. L. Comer, and C. A. Bouman, "A model-based iterative reconstruction approach to tunable diode laser absorption tomography," *IEEE Transactions on Computational Imaging*, vol. 3, no. 4, pp. 876–890, 2017.
- [6] S. Majee, D. H. Ye, G. T. Buzzard, and C. A. Bouman, "A model based neuron detection approach using sparse location priors," *Electronic Imaging*, vol. 2017, no. 17, pp. 10–17, 2017.
- [7] A. Ziabari, D. H. Ye, K. D. Sauer, J. Thibault, and C. A. Bouman, "2.5D deep learning for CT image reconstruction using a multi-GPU implementation," in *Signals, Systems, and Computers, 2018 52nd Asilomar Conference on*. IEEE, 2018.
- [8] J. Gibbs, K. A. Mohan, E. Gulsoy, A. Shahani, X. Xiao, C. Bouman, M. De Graef, and P. Voorhees, "The three-dimensional morphology of growing dendrites," *Scientific reports*, vol. 5, p. 11824, 2015.
- [9] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich, "Space-time tomography for continuously deforming objects," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [10] S. J. Kisner, E. Haneda, C. A. Bouman, S. Skatter, M. Kourinny, and S. Bedford, "Model-based CT reconstruction from sparse views," in *Second International Conference on Image Formation in X-Ray Computed Tomography*, 2012, pp. 444–447.
- [11] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 534–548, 1993.
- [12] D. Clark and C. Badea, "Convolutional regularization methods for 4D, x-ray CT reconstruction," in *Medical Imaging 2019: Physics of Medical Imaging*, vol. 10948. International Society for Optics and Photonics, 2019, p. 109482A.
- [13] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 408–423, 2016.
- [14] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 945–948.
- [15] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *arXiv preprint arXiv:1809.04693*, 2018.
- [16] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1872–1876, 2017.
- [17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [18] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4D nonlocal spatiotemporal transforms," in *Image Processing: Algorithms and Systems IX*, vol. 7870. International Society for Optics and Photonics, 2011, p. 787003.
- [19] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," *SIAM Journal on Imaging Sciences*, vol. 11, no. 3, pp. 2001–2020, 2018.
- [20] D. Jiang, W. Dou, L. Vosters, X. Xu, Y. Sun, and T. Tan, "Denoising of 3D magnetic resonance images with multi-channel residual learning of convolutional neural network," *Japanese journal of radiology*, vol. 36, no. 9, pp. 566–574, 2018.
- [21] Y. Sun, B. Wohlberg, and U. S. Kamilov, "Plug-in stochastic gradient method," *arXiv preprint arXiv:1811.03659*, 2018.
- [22] Y. Sun, S. Xu, Y. Li, L. Tian, B. Wohlberg, and U. S. Kamilov, "Regularized fourier ptychography using an online plug-and-play algorithm," *arXiv preprint arXiv:1811.00120*, 2018.
- [23] C. A. Bouman and K. Sauer, "A unified approach to statistical tomography using coordinate descent optimization," *IEEE Transactions on image processing*, vol. 5, no. 3, pp. 480–492, 1996.
- [24] V. Sridhar, G. T. Buzzard, and C. A. Bouman, "Distributed framework for fast iterative CT reconstruction from view-subsets," *Electronic Imaging*, vol. 2018, no. 15, pp. 102–1, 2018.
- [25] J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la Société mathématique de France*, vol. 93, pp. 273–299, 1965.
- [26] R. E. Williamson, R. H. Crowell, and H. F. Trotter, *Calculus of vector functions*. Prentice-Hall, 1972.
- [27] V. Sridhar, X. Wang, G. T. Buzzard, and C. A. Bouman, "Distributed memory framework for fast iterative CT reconstruction from view-subsets using multi-agent consensus equilibrium," *Manuscript in preparation for IEEE Transactions on Computational Imaging*, 2018.
- [28] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [29] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [30] P. Getreuer, "Rudin-osher-fatemi total variation denoising using split bregman," *Image Processing On Line*, vol. 2, pp. 74–95, 2012.
- [31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [32] R. Mathaes, H.-C. Mahler, J.-P. Buettiker, H. Roehl, P. Lam, H. Brown, J. Luemkemann, M. Adler, J. Huwyler, A. Streubel *et al.*, "The pharmaceutical vial capping process: Container closure systems, capping equipment, regulatory framework, and seal quality tests," *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 99, pp. 54–64, 2016.
- [33] S. Komura and H. Toyofuku, "Wave spring," Apr. 22 1997, US Patent 5,622,358.