Four-dimensional tomographic reconstruction by time domain decomposition

Viktor V. Nikitin, Marcus Carlsson, Fredrik Andersson, and Rajmund Mokso

Abstract—Since the beginnings of tomography, the requirement that the sample does not change during the acquisition of one tomographic rotation is unchanged. We derived and successfully implemented a tomographic reconstruction method which relaxes this decades-old requirement of static samples. In the presented method, dynamic tomographic data sets are decomposed in the temporal domain using basis functions and deploying an L1 regularization technique where the penalty factor is taken for spatial and temporal derivatives. We implemented the iterative algorithm for solving the regularization problem on modern GPU systems to demonstrate its practical use.

Index Terms—Tomographic reconstruction, motion artifacts, function decomposition, Total Variation

I. INTRODUCTION

TIME resolved four-dimensional X-ray computed tomography is widely used in medicine and material sciences where the inner structure of the sample under study is dynamically changing in time. The conventional approach to the dynamic tomography is to acquire measurement data during rotation with a constant angular step size. Then, the reconstruction is performed for each 180 degrees rotation cycle by using reconstruction methods such as Filtered Back-projection (FBP) or Algebraic Reconstruction Technique (ART). The obtained series of recovered objects then form temporal samples representing the object evolution in time. This scheme, however, assumes that the object is static during each 180 degrees interval. Any significant change in the object's structure during a single rotation cycle introduces motion artifacts apparent as blurred and corrupted reconstructions. There is therefore a constant race between the speed of sample rotation and the sample dynamics. Shorter scan times may be achieved by decreasing the detector exposure time or by reducing the number of projections for each 180 degrees. However, in practice, the reduced detector exposure time leads to a lower signal to noise ratio, so as the limited number of projections gives specific incompleteness artifacts.

Algorithms for four-dimensional tomographic reconstruction became of great interest especially with the development of fast detector systems in the last years. With brilliant synchrotron light sources, it is possible to perform continuous data acquisition with more than 8GB/s rate [1] and produce terabytes of acquired three-dimensional data sets and

V.V. Nikitin and R. Mokso are with the MAX IV Laboratory, Lund University, Lund, Sweden, email: (viktor.nikitin@maxiv.lu.se, rajmund.mokso@maxiv.lu.se)

M. Carlsson and F. Andersson are with the Centre for Mathematical Sciences, Lund University, Lund, Sweden, email: (marcus.carlsson@math.lu.se, fredrik.andersson@math.lth.se)

corresponding four-dimensional reconstructions in a single experiment. Processing of such a big data requires fast and optimized reconstruction algorithms and powerful software-hardware systems.

Various methods have been proposed for reconstructing tomography data for continuous dynamic acquisition. The first class of methods is based on estimating a prior information about the actual motion. For instance, space-time Gibbs priors define relationships among neighboring points in space and time by using information about the motion, see [2], [3], [4]. The algorithms proposed in [5], [6] in turn operate with estimated deformation vector fields (DVFs) between different time frames [7]. With a prior information, it is possible to keep reconstruction quality by decreasing the number of projection for a 180 degrees interval. Half-circle intervals representing different time frames can thus be scanned faster, that results in the suppression of motion artifacts. In the method proposed in [8], the authors generalize the usual back-projection along straight lines to dynamically curved paths constructed according to the motion model obtained from projections. Dynamically curved paths are then utilized for compensating deformations.

The second class of methods for suppressing motion artifacts takes into account the regularization in a non-local fashion. The methods analyze similarity between corresponding patches at different time steps, even if the patches have moved to another location. In [9] the non-local regularization penalty is an unweighted sum of distances between patch pairs in the three-dimensional object. Temporal Non-local Means (TNLM) method proposed in [10] utilizes weighting factors defined according to the ground truth objects. Another recent research was carried out by Kazantsev et. al [11], [12]. The authors estimate local structural correlations over multiple time frames in order to find the edges inside the object which remain constant in time, then the patched-based regularization (analog of the non-local regularization) is performed according to the obtained object structure. The authors also propose a sparsity seeking approach that operates with a non-local penalty for collecting only relevant information in the spatial and temporal spaces, see [13]. This approach sufficiently decreases the amount of computations.

The third class of methods for the four-dimensional tomographic reconstruction is built upon the concept of compressed sensing. Compressed sensing employs sparsity promoting algorithms with using the L_1 norm minimization. The spatial-temporal total variation regularization (STTVR) introduced by Wu et. al [14] operates with L_1 penalty factors for the gradient in spatial variables (Total Variation penalty)

and for the temporal variable, independently. The method proposed by Ritschl et.al [15] extends the Total Variation penalty to the temporal dimension by taking the gradient in four variables. Corresponding iterative scheme for solving regularization problem consists of two steps. The first step is fidelity term minimization by ART or CGLS, and the second is Total Variation minimization by the gradient-descent method. Authors note that such a separate minimization is caused by data dimensionality and high computational costs. There also exist methods where the penalty factor is represented by L_p norm with $p \in (0, 1)$, see [16], [17]. It is shown that in some cases the norm L_p is more effective than L_1 norm because it is closer to L_0 - a direct measure of sparsity.

The regularization problem that arises with employing the concept of compressed sensing for four-dimensional tomographic data reconstruction is non-smooth, therefore standard methods such as least-squares, conjugate gradients are not applied directly. For most non-smooth problems, a global optimum cannot be found with a given precision and in a reasonable time. The quality of solutions often entirely depends on the model, initial values, and optimization algorithms. The most common way for solving the regularization problem in four-dimensional tomography is by splitting the minimization function by parts and working with each part independently. Solutions from different parts are combined only after a particular number of iterations. Concerning the compressed sensing in tomography, the minimization is firstly performed for the fidelity term that includes the projection operator, then the contribution from the sparseness terms is taken into account. The primal-dual algorithm proposed by Chambolle et al. [18] allows solving the non-smooth minimization problem directly, i.e. by minimizing the resulting function for all terms together within one iteration. The authors proved that the algorithm has $O(\frac{1}{N})$ convergence rate in finite dimensions, and $O(\frac{1}{N^2})$ if the primal or the dual objective function is uniformly convex. The algorithm is presented as a general framework with established connections to other known algorithms such as Arrow-Hurwicz method [19], Douglas-Rachford splitting algorithm [20], [21], and preconditioned ADMM [22], [23].

An alternative regularization technique successfully used for the static tomography is a sparse data representation by using appropriate functions. In [24] the function set of wavelets is proposed for decomposition of the projection operator. The reconstruction is then based on thresholding of the noisy wavelet coefficients. Similar procedure of the tomographic data inversion is performed by using other types of functions: curvelets [25], shearlets [26], Gabor frames [27].

In this work, we use the concept of compressed sensing in the way that data in the temporal direction is represented by a linear combination of appropriate basis functions, and the L_1 norm minimization is performed for the gradient in both spatial and temporal variables. The choice of basis functions depends on the motion structure inside the object and can be determined according to the measured data. For solving obtained non-smooth regularization problem, we adopt the primal-dual Chambolle-Pock algorithm [18].

There are two main advantages of the proposed method compared to the methods mentioned above. First, in contrast to other approaches, we address the cases where rapid motions happen during 180 degrees interval. Other methods operate with different strategies for decreasing the number of projections to represent this interval, and, in that way, require small amplitude of structural changes inside the interval. Second, the proposed method allows operating with big real four-dimensional data that has many samples in the temporal direction. This is achieved by the time-domain data decomposition that sufficiently decreases data sizes. Reconstruction, in this case, are reasonably fast, whereas the existing methods are time-consuming to an extension that impairs their practical use for standard-sized data volumes.

The paper is organized as follows. In Section 2, we introduce the projection operator in 4D, and explain the regularization strategy. Section 3 shows how the minimization problem for recovering objects can be solved by the Chambolle-Pock algorithm. How to decrease the number of computation by representing data as a linear combination of appropriate basis functions is discussed in Section 4. In section 5 we validate our approach on synthetic data, while in section 6 we process two experimental data sets. Details for technical algorithm acceleration by utilizing high-performance facilities with CPUs and GPUs are given in Section 7. Conclusions and outlook are given in Section 8.

II. REGULARIZATION PROBLEM

Let $f(x, y, z, t)$ be function which represents a three-dimensional object dynamically changing in time t. The object is rotated continuously and projection data is measured for angles θ and for radial direction s. The projection operator in this case is described by integration over the lines through the object state at time t, which is connected to the rotation angle θ. We assume a linear connection between the angle θ and time t, i.e. $\theta = \alpha t$, where the parameter α in practice is related to the speed of rotation and the detector exposure time. The projection operator $\mathcal{R}_\alpha : \mathbb{R}^3 \times [0, T] \to \mathbb{R}^2 \times S^1$ where S^1 denotes the unit circle, is defined by

$$\mathcal{R}_\alpha f(s, z, \theta) =$$
$$\iiint f(x, y, z, t)\delta(x\cos\theta + y\sin\theta - s)\delta(\theta - \alpha t)dx\, dy\, dt$$
$$(1)$$

The corresponding adjoint operator $\mathcal{R}_\alpha^* : \mathbb{R}^2 \times S^1 \to \mathbb{R}^3 \times [0, T]$ is defined as follows

$$\mathcal{R}_\alpha^* g(x, y, z, t) =$$
$$\iint g(\theta, s, z)\delta(x\cos\theta + y\sin\theta - s)\delta(\theta - \alpha t)ds\, d\theta$$
$$(2)$$

The inverse problem of recovering the function f from the measurements $g = \mathcal{R}_\alpha f$ has plenty of possible solutions. The non-uniqueness is caused by the fact that at each particular time frame t there exist only one projection related to the angle $\theta = \alpha t$, which is surely not enough to recover the unique object structure. In this case, regularization can be used to introduce assumptions on the solution. The traditional

approach to finding a unique solution is by minimizing the data fidelity term as

$$\hat{f} = \operatorname{argmin}_f \left\{ \frac{1}{2} \|\mathcal{R}_\alpha f - g\|_2^2 \right\} \tag{3}$$

This term is commonly used in static tomography where the object does not change during 180 degrees rotation. Since the cost function is quadratic, one can use gradient-based methods such as the standard least-squares iteration scheme, or the conjugate gradient least-squares scheme with a faster rate of convergence. It is also common to use tomography specific methods. Indeed, recovering the object structure from a limited number of the measured projection angles can be done by algebraic reconstruction methods [28], [29], [30], suppressing the Poisson noise in reconstruction is typically done by the EM algorithm [31], [32].

None of the listed methods operate with data along the time axis and, consequently, could produce a big number of possible solutions for (3). Thus, we consider additional penalty factor $J(f)$ and focus on the following regularization problem

$$\hat{f} = \operatorname{argmin}_f \left\{ \frac{1}{2} \|\mathcal{R}_\alpha f - g\|_2^2 + \lambda J(f) \right\}, \tag{4}$$

where λ is a regularization parameter establishing a trade-off between the data fidelity term and the regularization term.

How to solve the minimization problem (4) depends on the structure of the penalty term $J(f)$. The gradient-based algorithms can be used for the quadratic penalty $J(f) = \|Kf\|_2^2$, where K is a linear operator such as the gradient ∇. With $J(f) = \|Kf\|_1$ the resulting function becomes non-differentiable and more complicated methods have to be used. The methods assume computation of the proximal operator which is used to make an approximation to a value while making a compromise between the accuracy of the approximation and a cost associated to the new value. There exist proximal splitting methods that are based on the fact that the functions $\frac{1}{2}\|\mathcal{R}_\alpha f - g\|_2^2$ and $\lambda J(f)$ are used individually yielding an easily implementable algorithm. These methods include, for instance, Forward-Backward splitting [33], DouglasRachford splitting [20], [21], and Alternating-Direction Method of Multipliers (ADMM) [22], [23]. The primal-dual algorithm proposed by Chambolle et al. [18] also operates with proximal operators, however, minimization of the resulting function is performed for all terms together. Non-smooth penalties such as $J(f) = \|Kf\|_p$ for $p \in (0,1)$ are beneficial for even more sparse results, cf. [16], [17]. However, the developed algorithms are more complicated and additional assumptions are required to guarantee convergence.

In this work we introduce the regularization term described with respect to four variables as

$$\lambda_1 \big\| |\nabla_{\lambda_2} f| \big\|_1 =$$
$$\lambda_1 \left\| \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial z}\right)^2 + \left(\lambda_2 \frac{\partial f}{\partial t}\right)^2} \right\|_1, \tag{5}$$

where constant λ_1 denotes a trade-off between data fidelity and the regularization term, and constant λ_2 controls the level of sparseness in the temporal direction. The inclusion of the temporal derivative in the regularization term results in preserving rapid data changes and diminishes small data changes in the temporal direction. This assumes that the object under study does not rapidly change the whole structure, i.e., most object parts keep constant between adjacent time frames. The inclusion of derivatives in spatial directions, in turn, preserve sharp edges of the object inner structure and minimize noise components. Moreover, spatial derivatives allow controlling possible artifacts coming from the temporal derivative since they penalize rapid temporal data change of adjacent spatial points.

As a result, with regularization term (5) we establish a connection between spatial and temporal variables. Alternatively, one can consider spatial and temporal penalty factors independently. However, in this case the temporal sparseness may not be controlled by the object inner structure. Finally, we propose to use the following model for recovering function f from the measurements g

$$\hat{f} = \operatorname{argmin}_f \left\{ \frac{1}{2} \|\mathcal{R}_\alpha f - g\|_2^2 + \lambda_1 \big\| |\nabla_{\lambda_2} f| \big\|_1 \right\} \tag{6}$$

For solving this regularization problem we decided to use the first-order, primal-dual algorithm of Chambolle and Pock because of its general formulation. In [18] the authors show that most popular algorithms, including DouglasRachford splitting and ADMM, are particular cases of the Chambolle-Pock algorithm.

III. CHAMBOLLE-POCK ALGORITHM FOR DYNAMIC DATA RECONSTRUCTION

In this section, we will recapitulate main structure of the Chambolle-Pock algorithm and show how to apply it to the regularization scheme (6).

The algorithm operate with proximal operators defined for a function $F(f)$ as follows

$$\operatorname{prox}_{\sigma F}(h) = \operatorname{argmin}_f \left\{ F(f) + \frac{1}{2\sigma} \|h - f\|_2^2 \right\},$$

where σ defines a trade-off between two minimization terms. The proximal operator can be interpreted as an approximation to a value, while making a compromise between the accuracy of the approximation $\|h - f\|_2^2$ and a cost associated to the new value (function $F(f)$). We refer to [34] for more details about proximal operators and applications. The algorithm is designed for solving the saddle-point problem described by

$$\hat{f} = \operatorname{argmin}_f \left\{ \max_h \left(\langle Kf, h \rangle + G(f) + F^*(h) \right) \right\}, \tag{7}$$

which is the dual formulation of the problem $\hat{f} = \operatorname{argmin}_f \{F(Kf) + G(f)\}$. Here K is a linear map, F^*, G are proper, convex, lower-semicontinuous (l.s.c.) functions, F^* being itself the convex conjugate of a convex l.s.c. function F. The algorithm in Figure 1 solves the problem (7) with $1/N$ rate of convergence.

The algorithm can be adapted for solving (6) with $F(Kf) = F_1(K_1 f) + F_2(K_2 f)$ ($G(f) = 0$), where linear operator K_1

1: Initialize: $\theta \in [0,1]$, $\tau\sigma\|K\|^2 < 1$, (f_0, h_0) is some initial
   guess, $\tilde{f}^0 = f^0$
2: **repeat**
3:      $h^{n+1} = \text{prox}_{\sigma F^*}(h^n + \sigma K \tilde{f}^n)$
4:      $f^{n+1} = \text{prox}_{\tau G}(f^n - \tau K^* h^{n+1})$
5:      $\tilde{f}^{n+1} = f^{n+1} - \theta(f^n - f^{n+1})$
6: **until** convergence criteria met

Fig. 1. Chambolle-Pock algorithm. General formulation.

acting on f produce functions p_1 which then are used as input
for functional F_1,

$$p_1 = K_1 f = \mathcal{R}_\alpha f, \qquad F_1(p_1) = \frac{1}{2}\|p_1 - g\|_2^2,$$

and where linear operator K_2 acting on f produces a spatial
vector field $\vec{p}_2$ that is used as an input for functional F_2,

$$\vec{p}_2 = K_2 f = \nabla_{\lambda_2} f, \qquad F_2(\vec{p}_2) = \lambda_2\big\|\|\vec{p}_2\|\big\|_1.$$

The corresponding dual problem is obtained by adding three
dual variables $(h_1, \vec{h}_2)$ with respect to $(p_1, \vec{p}_2)$. It has the
following form

$$\hat{f} = \text{argmin}_f \max_{h_1, \vec{h}_2} \Big\{ \langle K_1 f, h_1\rangle + \langle K_2 f, \vec{h}_2\rangle + F_1^*(h_1) + F_2^*(\vec{h}_2)\Big\}$$

Corresponding convex conjugate functions for F_1, F_2 and
adjoint operators for K_1, K_2 are computed as follows

$$K_1^*(p_1) = \mathcal{R}_\alpha^* p_1, \qquad K_2^*(\vec{p}_2) = -\text{div}_{\alpha_2}\vec{p}_2,$$

$$F_1^*(h_1) = \max_{h_1'}\{\langle h_1, h_1'\rangle - \frac{1}{2}\|h_1' - g\|_2^2\} = \langle h_1, g\rangle + \frac{\|h_1\|_2^2}{2},$$

$$F_2^*(\vec{h}_2) = \max_{\vec{h}_2'}\{\langle \vec{h}_2, \vec{h}_2'\rangle - \lambda_1\big\|\|\vec{h}_2'\|\big\|_1\} = \kappa(\vec{h}_2),$$

where $-\text{div}_{\lambda_2}$ is the adjoint operator to $-\nabla_{\lambda_2}$, and $\kappa(\vec{h}_2) = 0$
if $\lambda\big\|\|\vec{h}_2\|\big\|_1 \leq 1$, otherwise $\kappa(\vec{h}_2) = \infty$. Now it is straightfor-
ward to compute corresponding proximal operators:

$$\text{prox}_{\sigma F_1^*}(h_1) = \text{argmin}_{h_1'}\big\{\langle h_1', g\rangle + \frac{\|h_1'\|_2^2}{2} + \frac{1}{2\sigma}\|h_1 - h_1'\|_2^2\big\}$$
$$= (h_1 - \sigma g)/(1 + \sigma),$$

$$\text{prox}_{\sigma F_2^*}(\vec{h}_2) = \text{argmin}_{\vec{h}_2'}\big\{\kappa(\vec{h}_2') + \frac{1}{2\sigma}\|\vec{h}_2 - \vec{h}_2'\|_2^2\big\}$$
$$= \vec{h}_2 / \max(1, \big\|\|\vec{h}_2\|\big\|_1/\lambda_1),$$

$$\text{prox}_{\tau G}(f) = \text{argmin}_{f'}\big\{0 + \frac{1}{2\tau}\|f - f'\|_2^2\big\} = f.$$

We result in the algorithm shown in Figure 2 that summa-
rizes main steps for solving (6).

IV. PROBLEM OPTIMIZATION

The regularization problem (6) becomes computation and
memory intensive if one aims to work with the object repre-
sented by several hundred or thousands of samples in each
spatial and temporal variables. The most computationally
intensive parts in the algorithm involve evaluating the projec-
tion operator $\mathcal{R}_\alpha$ and the corresponding adjoint operator $\mathcal{R}_\alpha^*$

1: Initialize: $\theta \in [0,1]$, $\tau\sigma < 1^*$, $f^0, h_1^0, \vec{h}_2^0$ is some initial
   guess, $\tilde{f}^0 = f^0$
2: **repeat**   //description
3:    $h_1^{n+1} = \frac{h_1^n + \sigma\mathcal{R}_\alpha \tilde{f}^n - \sigma g}{1 + \sigma}$,
             $//h_1^{n+1} = \text{prox}_{\sigma F_1^*}(h_1^n + \sigma K_1 \tilde{f}^n)$
4:    $\vec{h}_2^{n+1} = \frac{\vec{h}_2^n + \sigma\nabla_{\lambda_2}\tilde{f}^n}{\max\big(1, \big\|\|\vec{h}_2^n + \sigma\nabla_{\lambda_2}\tilde{f}^n\|\big\|_1/\lambda_1\big)}$,
             $//\vec{h}_2^{n+1} = \text{prox}_{\sigma F_2^*}(\vec{h}_2^n + \sigma K_2 \tilde{f}^n)$
5:    $f^{n+1} = f^n - \tau\mathcal{R}_\alpha^* h_1^{n+1} + \tau\text{div}_{\lambda_2}\vec{h}_2^{n+1}$,
             $//f^{n+1} = \text{prox}_{\tau G}(f^n - \tau K_1^* h_1^{n+1} - \tau K_2^* \vec{h}_2^{n+1}))$
6:    $\tilde{f}^{n+1} = f^{n+1} + \theta(f^{n+1} - f^n)$.
7: **until** convergence criteria met
* linear operators ∇_{λ_2}, $\mathcal{R}_\alpha$ are normalized after a constant
multiplication.

Fig. 2. Chambolle-Pock algorithm for dynamic tomography.

operator many times. If we assume that the number of samples
in each spatial and in the temporal (or angular) direction is
of the order of N, then the complexity of computing the
operators $\mathcal{R}_\alpha$ and $\mathcal{R}_\alpha^*$ by directly discretizing the integrals in
formulas (1) and (2) is $\mathcal{O}(N^4)$. In static tomography there exist
several approaches how to decrease computational complexity
to $\mathcal{O}(N^3 \log N)$, where all angles are used for reconstruction
simultaneously. The methods include Fourier-based methods
[35], [36], the log-polar-based method [37], [38], hierarchical
decomposition [39], [40]. These methods are not directly used
in the dynamic tomography problem since each projection is
related to a particular object state in time. However, in what
follows we will show that the methods are still in use for
dynamic tomography.

One way of how to reduce resources for computations is by
introducing an additional assumption on the object movement.
Let us assume the motion at each concrete space sample
(x, y, z) can be approximated by a linear combination of basis
functions $\{\varphi_j(t)\}_{j=0}^{M-1}$, i.e.,

$$f(x, y, z, t) \approx \sum_{j=0}^{M-1} f_j(x, y, z)\varphi_j(t), \qquad (8)$$

where $\{f_j(x, y, z)\}_{j=0}^{M-1}$ are decomposition coefficients.
Choice of basis functions φ_j for better approximation de-
pends on the motion structure. As a straightforward example,
one can choose the Fourier basis with a low number of
coefficients to represent slow motions, and a high number
of coefficients to represent rapid motions. Other possible
functions for representation include Haar wavelets, Heaviside
step functions, as well as their smooth approximations (Gram-
Schmidt Orthonormalization [41] can be used to make an
orthonormal basis if necessary). It should be also noted that
typically the object needs to be reconstructed only with some
particular step in time so that the total number of time frames
is sufficiently smaller than the total number of angles. This
fact potentially allows decreasing the number of coefficients
for representation.

Now by making use of representation (8) and exploiting the linearity property of the projection operator we have

$$\mathcal{R}_\alpha f(s,z,\theta) = \sum_{j=0}^{M-1} \mathcal{R}f_j(\theta,s,z)\varphi_j\left(\frac{\theta}{\alpha}\right), \qquad (9)$$

where the operator $\mathcal{R}f(\theta,s,z) = \iint f(x,y,z)\delta(x\cos\theta + y\sin\theta - s)dx\,dy$ denotes the standard Radon transform computed for the whole set of angles θ. For the constructed algorithm (Figure 2) we also need the representation of the adjoint operator $\mathcal{R}_\alpha^*$. It can be found through the adjoint equality

$$\langle \mathcal{R}_\alpha f, g\rangle \approx \left\langle \sum_{j=0}^{M-1} \mathcal{R}f_j(\theta,s,z)\varphi_j\left(\frac{\theta}{\alpha}\right), g(\theta,s,z)\right\rangle =$$

$$\sum_{j=0}^{M-1} \langle f_j(\theta,s,z), \mathcal{R}^*(g\hat{\varphi}_j)(x,y,z)\rangle =$$

$$\left\langle f, \sum_{j=0}^{M-1} \varphi_j(t)\mathcal{R}^*(g\hat{\varphi}_j)(x,y,z)\right\rangle,$$

which gives the following approximation for the adjoint operator

$$\mathcal{R}_\alpha^* g(x,y,z,t) \approx \sum_{j=0}^{M-1} \varphi_j(t)\mathcal{R}^*(g\hat{\varphi}_j)(x,y,z) \qquad (10)$$

Computing the projection operator $\mathcal{R}_\alpha$ and its adjoint $\mathcal{R}_\alpha^*$ with respect to formulas (9) and (10) can sufficiently decrease the number of computations and allocated memory because of two facts. First, as mentioned above, the number of needed coefficients M is typically much smaller than the number of samples in time. Second, the Radon transform for each coefficient, $\mathcal{R}f_j$ in formula (9), as well as the adjoint operator $\mathcal{R}^*(g\hat{\varphi}_j)$ in formula (10), are computed independently on the temporal variable. With this fact, $\mathcal{R}f_j$ becomes periodic with a period 2π. Also, the Radon transform for the interval $[\pi, 2\pi)$ is the same as the Radon transform for the interval $[0,\pi)$ after changing the sign of variable s. So instead of computing the Radon transform $\mathcal{R}f_j$ for the whole set of angles θ it is enough to compute the transform only for the interval $[0,\pi)$ and simply distribute the result to other angles. In what follows we give a detailed description of how it works, as well as how this idea is adapted for computing $\mathcal{R}^*(g\hat{\varphi}_j)$.

For simplicity we can assume that the object rotation includes exactly N_π angular intervals of size π, i.e. $\theta \in [0, \pi N_\pi)$. In other cases, one can subtract this interval from the whole set of angles, and work with the remaining part separately. For the chosen interval we have $\theta = k\pi + \theta_0$, where $\theta_0 \in [0,\pi)$ and k is an integer value. Then we have the following expression for the Radon transform $\mathcal{R}f_j(\theta,s,z)$ from formula (9),

$$\mathcal{R}f_j(\theta,s,z) = \mathcal{R}f_j(k\pi + \theta_0, s, z) = \mathcal{R}f_j(\theta_0, (-1)^k s, z)$$

In turn, the adjoint operator $\mathcal{R}^*(g\hat{\varphi}_j)$ from formula (10) can be rewritten as follows

$$\mathcal{R}^*(g(\theta,s,z)\hat{\varphi}_j(\theta,s,z)) =$$

$$\sum_{k=0}^{N_\pi-1} \mathcal{R}^*\left(g(k\pi+\theta_0, (-1)^k s, z)\hat{\varphi}_j(k\pi+\theta_0, (-1)^k s, z)\right) =$$

$$\mathcal{R}^*\left(\sum_{k=0}^{N_\pi-1} g(k\pi+\theta_0, (-1)^k s, z)\hat{\varphi}_j(k\pi+\theta_0, (-1)^k s, z)\right) =$$

$$\mathcal{R}^*(\tilde{g}_j(\theta_0, s, z))$$

where $\tilde{g}_j(\theta_0,s,z) = \sum_{k=0}^{N_\pi-1} g(k\pi+\theta_0, (-1)^k s, z)\hat{\varphi}_j(k\pi+\theta_0, (-1)^k s, z)$. Here instead of computing the adjoint Radon transform for the whole set of angles it is enough to sum up data over angular intervals of size π and then compute the adjoint operator only for the angular interval $[0,\pi)$.

It is straightforward to adapt steps of the algorithm in Figure (2) with corresponding decompositions (9-10) for discrete spatial variables, discretization in temporal variable in turn needs to be explained. In what follows for simplicity we ignore discretization with respect to spatial variables x, y, z by introducing the following simplified notation

$$\mathsf{f}(t) = f(x,y,z,t),$$
$$\mathsf{Rf}(\theta) = \mathcal{R}f(\theta,s,z).$$

Now let us assume that the total number of samples in the angular direction is N_θ, and the total number of reconstructed time frames is N_t, where in practice we are interested in $N_t \ll N_\theta$. For instance, the first time frame is associated with projection angle $0°$, the second - to $180°$, the third - to $360°$, etc. We let m be the amount of angles per time frame, i.e. we connect samples in the temporal direction $\{t\}_{k=0}^{N_t-1}$ to the angular samples $\{\theta_k\}_{k=0}^{N_\theta-1}$ in the following way

$$\theta_k = k\Delta_\theta, \quad k = 0\ldots N_\theta - 1$$
$$t_k = \frac{\theta_{km}}{\alpha} = \frac{km\Delta_\theta}{\alpha}, \quad k = 0,\ldots N_t - 1 \qquad (11)$$

where the parameter α shows a linear connection between temporal and angular samples as in the definition (1) of the projection operator. With discretization (11) the derivative operator $\frac{\partial f}{\partial t}$ in the algorithm from Figure 2 at discrete samples reads as

$$\frac{\mathsf{f}(t_{k+1}) - \mathsf{f}(t_k)}{\Delta t} = \frac{\alpha(\mathsf{f}(\theta_{km+m}) - \mathsf{f}(\theta_{km}))}{\Delta_\theta}, \quad k = 0\ldots N_t - 1$$

The discrete version of the projection operator $\mathcal{R}_\alpha$ having the approximation (9) is written as

$$\mathsf{R}_\alpha \mathsf{f}(\theta_k) = \sum_{j=0}^{M-1} \mathsf{Rf}_j(\theta_k)\varphi_j\left(\frac{\theta_k}{\alpha}\right), \quad k = 0\ldots N_\theta - 1$$

where the coefficients f_j are computed as follows

$$\mathsf{f}_j = \sum_{k=0}^{N_t-1} \mathsf{f}(t_k)\varphi_j(t_k) = \sum_{k=0}^{N_t-1} \mathsf{f}\left(\frac{\theta_{km}}{\alpha}\right)\varphi_j\left(\frac{\theta_{km}}{\alpha}\right),$$
$$j = 0,\ldots M - 1$$

V. Validation

In this section we validate our approach through simulations.

For the validation we use a synthetic model considering only two spatial variables x, y and consisting of circles with varying spatial distribution in time (see Figure 3). The displacement of the spheres is captured in 8 intervals of each consisting by angular views across π. The two upper panels of the left column show two sequential time frames 3 and 4. The plots also contain circle border marks of the next object state. For instance, the circle border masks of time frame 3 correspond to circle borders of the time frame 4. Note that the displacement of circles between time frames 3 and 4 is smaller than between time frames 4 and 5. The model has sizes $(N, N, N_t) = (256, 256, 8)$, and the projection data has sizes $(N_\theta, N) = (128 \times 8, 256) = (1024, 256)$, where 128 is the number of projections for covering the interval $[0, \pi)$. Two upper panels of the middle column in Figure 3 show reconstruction by using standard FBP method where the Shepp-Logan filter is utilized for suppressing high frequencies in projections. Reconstruction of time frames 3,4 is performed by using projections from the intervals $[3\pi, 4\pi)$ and $[4\pi, 5\pi)$, respectively. Motion artifacts in the recovered time frame 3 are relatively small since the object inner structure has not rapid changes between time frames 3 and 4, whereas the recovered time frame 4 has clearly visible motion artifacts due to fast movement of circles between time frames 4 and 5. The same behavior demonstrates the regularized algebraic reconstruction (two upper panels in the right column of Figure 3) where the penalty factor is taken only for spatial derivatives in x and y, without any penalty for the temporal derivative. This method suppressed the noise in data and diminished limited angles artifacts, however, it does not deal with motion artifacts.

The bottom six panels in Figure 3 demonstrate results of the proposed method, with a different number of basis function used for data representation in the temporal direction. In this example, we consider the Fourier basis with $M = 8, 16, 32$ elements defined by

$$\varphi_j(t) = e^{2\pi i t(j - M/2)}, \quad j = 0 \dots M - 1.$$

Recall that the Fourier basis with a higher number of coefficients is used to recover more rapid motions. The results show that some motion artifacts are suppressed with $M = 8$ basis functions, and almost all artifacts are suppressed for the cases $M = 32$ basis functions. Selecting the number of basis functions is the main free parameter that the user have to adjust in the proposed method.

VI. Applications to experimental data

In the following we show the reconstruction of experimental data. We investigate the rheology of liquid foams by fast synchrotron X-ray tomographic microscopy [42]. Foams are complex cellular systems which require artifact free tomographic reconstruction for a reliable quantification of their time-dependent properties such as deformation fields of bubbles. In our example we acquire X-ray projections of the liquid foam flowing through a constriction and being rotated around the tomographic axis at a rate of 840 deg/s. Each X-ray exposure takes 0.7 ms and in total we acquire 130 tomographic data sets.

The experiment was performed at the TOMCAT beamline of the Swiss Light Source using the fast acquisition setup [1]. The data size is $(N, N_\theta, N_z) = (2016, 300 \times 130, 1800)$, where 300 is the number of projection inside the interval $[0, \pi)$, and 130 is the total number of time frames. Prior to tomographic reconstruction the raw projections were by phase retrieval algorithm to account for the interference occuring due to the partial coherence properties of the synchrotron beam [43], [44]. As an example of reconstruction we considered the data that corresponds to vertical slices $z \in [1200, 1328)$. Two time frames recovered by using static FBP method are shown in Figure 4, left. The tomogram at time frame 94 does not contain motion artifacts because the amplitude of the sample motion during the acquisition of this time-frame is small enough to meet the static sample assumption. The tomogram at time frame 95 shows motion artifacts due to the rapid movement of bubbles between time frames 95 and 96. As seen in Figure 4, right, by applying our new method (algorithm in Figure 2) we overcome the motion artifacts in time frame 95, but also maintain the image quality in the static case (frame 94). For reconstruction we use $M = 16$ basis functions, parameters λ_1, λ_2 showing trade-off between data fidelity and the derivative parts (see formulation (6)) have been experimentally chosen as 2^{-12} and 2^2, respectively. The iterative scheme in the algorithm is performed for 512 iterations, the result of reconstruction by the FBP is chosen as an initial guess.

We have also tested our method for reconstructing dynamic tomography data recently acquired at 2-BM beamline at Advanced Photon Source. The sample was prepared as the slurry of ceramic particles in alcohol. Along with alcohol evaporation, the ceramic particles aggregated to form clusters. In total, 12 tomographic data sets were continuously measured with 6 deg/s rotation rate and 30 ms exposure time. The data size is $(N, N_\theta, N_z) = (2560, 900 \times 12, 700)$, where 900 is the number of projection inside the interval $[0, \pi)$. The left part of Figure 5 shows two time frames corresponding to angles $\theta \in [7\pi, 8\pi]$ and vertical slices $z \in (250, 314)$ recovered by using static FBP method. Both reconstructed time frames contain motions artifacts that are mostly seen at the regions where ceramic particles are moving in the direction of the vertical central axis. The proposed method substantially suppress motion artifacts due to the rapid movement of particles, see Figure 5, right. It should be noted, that the method also improves the resolution level since the structure of particles becomes more reliable for segmentation. For reconstruction we use $M = 24$ basis functions, parameters λ_1, λ_2 have been experimentally chosen as 2^{-9} and 2^2, respectively. The iterative scheme in the algorithm from Figure 2 is performed for 512 iterations, the result of reconstruction by FBP is chosen as an initial guess. Note the noise reduction effect of the regularized reconstruction as compared to FBP. This is more apparent in this dataset than in the case of the foam data because the phase retrieval used in the foam example acts similar to a low pass filter.
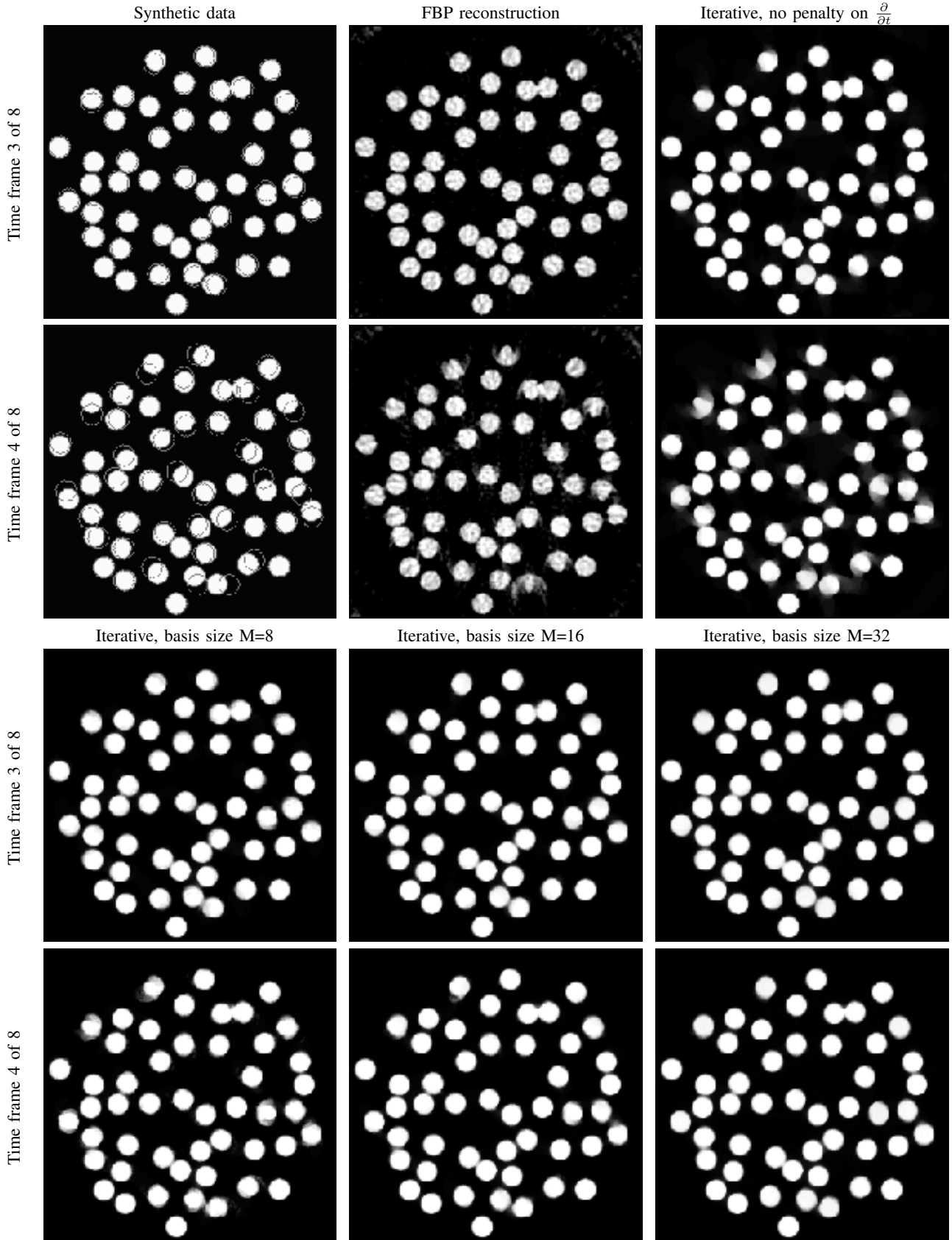
Fig. 3. Two time frames of the synthetic data and corresponding reconstruction by FBP method and by using the iterative scheme with penalty on the gradient in spatial variables (two top rows), reconstruction by the proposed iterative scheme with penalty on the gradient in spatial and temporal variables, for $M = 8, 16, 32$ basis functions (two bottom rows).
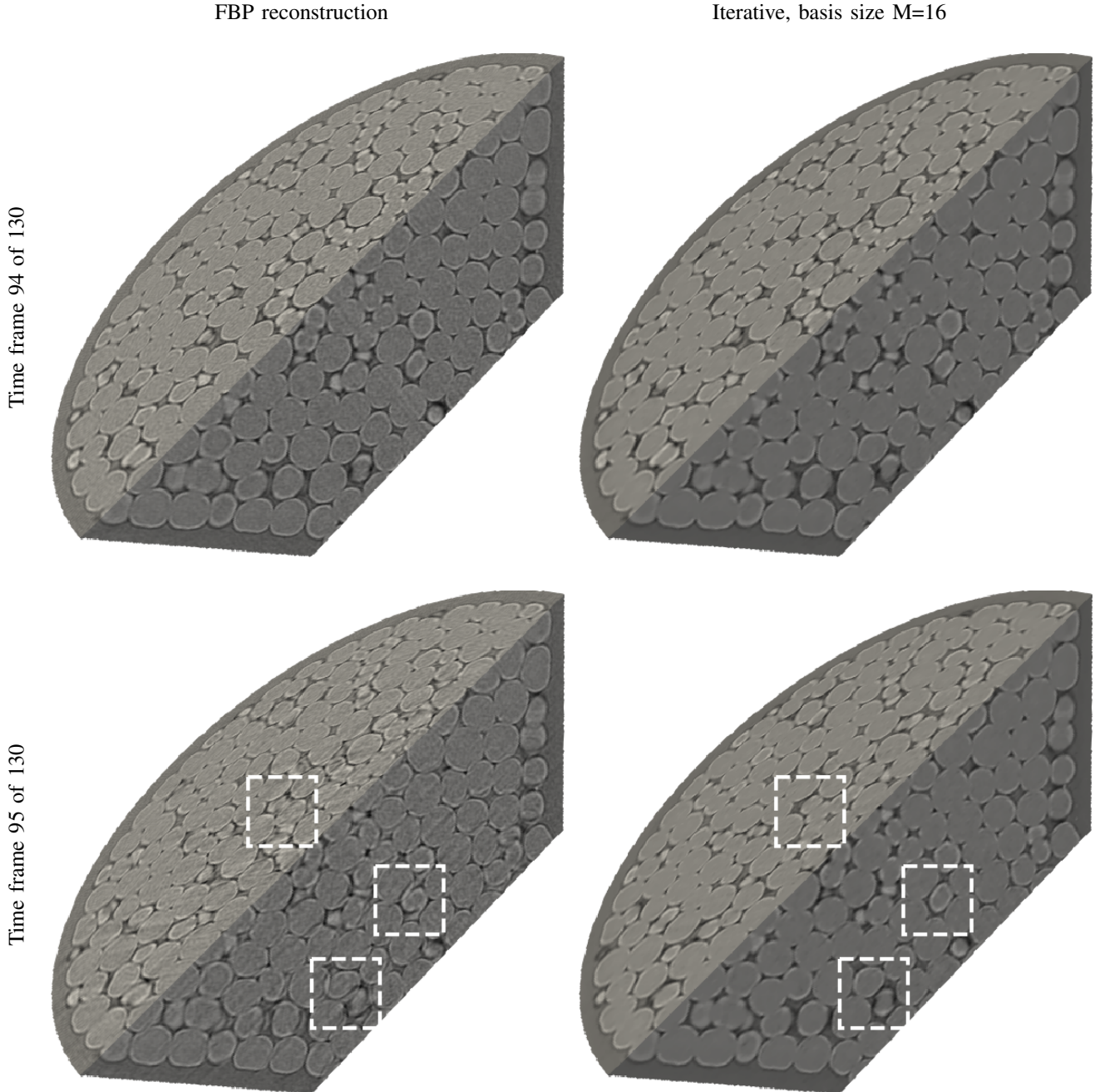
FBP reconstruction   Iterative, basis size M=16

Fig. 4. Two time frames of the foam in a glass (half-cropped to show vertical slices) reconstructed by FBP method (left) and by the proposed iterative scheme with penalty on the gradient in spatial and temporal variables (right).

VII. PARALLEL IMPLEMENTATION

The reconstruction problem (6) remains resource intensive even after considering decomposition techniques described in Section IV. Besides the mathematical improvement of the algorithm, we propose its technical acceleration by using parallel computing on CPUs and GPUs. Parallel computations on CPUs are typically done with making use of OpenMP and MPI technologies. NVidia CUDA technology in turn provides an interface to accelerate computations on GPUs. The main difference between these two platforms is that the number of computational cores on GPUs is more than 100 times bigger than the number of cores on CPU. However, GPU cores are much slower than CPU cores, moreover, the memory handling mechanism on GPU is not so advanced as the mechanism on CPU. Nevertheless, GPUs have shown themselves as a powerful tool to accelerate algorithms in tomography [38], [45], [46], [47]. This is because most reconstruction algorithms contain a lot of small identical operations that can be computed independently. We will give some details of how the algorithm from Figure 2 is parallelized for computing on several GPUs, and compare its parallel implementation to the OpenMP implementation on CPU.

The most resource intensive parts of the algorithm are the Radon transform $\mathcal{R}_\alpha$ in Step 3, and the corresponding
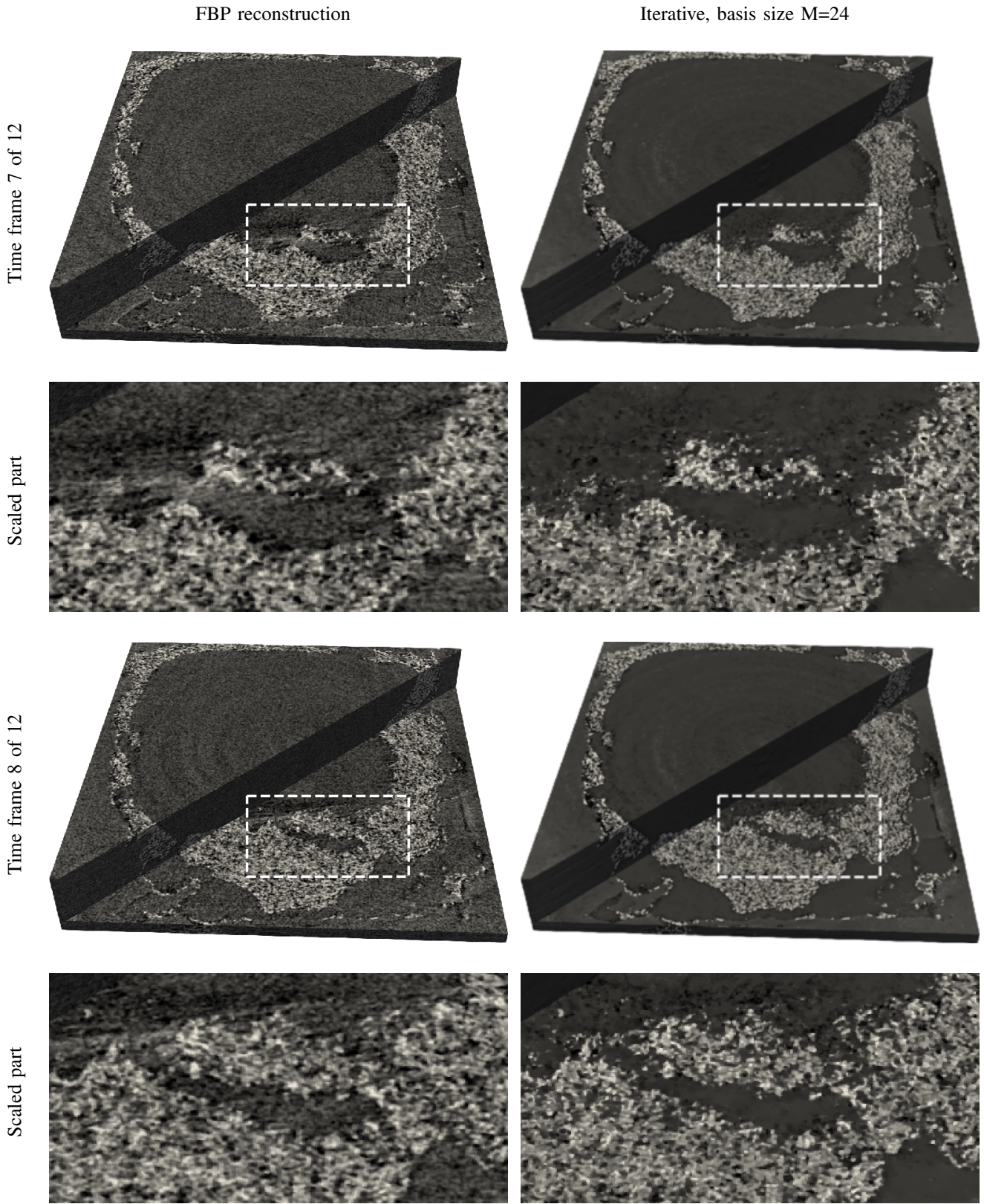
FBP reconstruction   Iterative, basis size M=24

Fig. 5. Two time frames of ceramic particles aggregation in alcohol, reconstructed by FBP method (left) and by the proposed iterative scheme with penalty on the gradient in spatial and temporal variables (right).

adjoint operator $\mathcal{R}_\alpha^*$ in Step 5. We have already described in Section IV how these operators are computed through the standard Radon transform and the back-projection for the angular interval of size π, see formulas (9) and (10). There exist several fast GPU implementations for computing these standard operators. For the proposed algorithm we decided to use the log-polar-based method [37] implemented on GPU since it demonstrates the best performance compared to known analogs, cf. [38]. We have also constructed fast GPU kernels for evaluation of gradient and divergence operators in Steps 4 and 5, as well as for evaluation of standard algebraic operations in Steps 3-6.

The time-resolved reconstruction algorithm must operate with four-dimensional big data sets, therefore, computations have to be split by parts so that the processing data fit into the operating memory, and to the memory on GPU. A straightforward approach generally used for reconstructing tomography data is to split computations by slices (in the z-variable) where all slices are recovered independently. Typically one GPU is used to recover a set of slices that fit into the GPU memory. This approach has to be slightly modified in our case since the algorithm also includes computations of derivatives in the z direction. To deal with that we transfer to GPU two extra slices located before and after the current set of slices processed by one GPU. There are no thread concurrency errors since the additional data from the two slices is kept read-only. Suppose that we intend to parallelize the algorithm by using `Nth` number of threads denoted by `ith=0 : Nth-1`. For simplicity, we assume that the total number of slices is a multiple of `Nth`. The procedure is done for Steps 3..6 located inside the iterative loop **repeat**..**until**, synchronization of parallel threads is done after Step 6. Parallel computation of Step 3 is done by separating the whole set of slices by parts associated with the thread number `ith as z = ith*Nz/Nth : (ith+1)*Nz/Nth-1`. Steps 4 and 5 include computation of operators ∇_{λ_2} and div_{λ_2} that assumes using extra slices in the z direction. Therefore, in Step 3 for the input read-only function $\tilde{f}^n$ we use slices with numbers `z = max(0,ith*Nz/Nth-1) : min((ith+1)*Nz/Nth,Nz-1)` and compute the vector field $\vec{h}_2$ for `z = max(0,ith*Nz/Nth-1) : (ith+1)*Nz/Nth-1`. In turn, in Step 4 the new function f^{n+1} appears for `z = ith*Nz/Nth : (ith+1)*Nz/Nth-1` by making use of the input vector field $\vec{h}_2$. Parallel computation of the Step 6 is straightforward with `z = ith*Nz/Nth-1 : (ith+1)*Nz/Nth-1`. Switching to the next iteration $n \to n+1$ is performed after thread synchronization.

The GPU implementation is also optimized by making use of the technique for overlapping data transfers with computations. Nvidia CUDA library provides the streams technology for simultaneous execution of different code parts. One optimal solution is to create three streams for 1) Host to Device data migration, 2) Computation on GPU, 3) Device to Host data migration. While one data part is computed, the next part can be loaded to GPU memory, as well as the result of the previous part can be unloaded from GPU memory. This strategy is especially good with introducing NVLink technology [48] that allows fast bidirectional copy between CPU and GPU.

We note that the parallelization strategy proposed above can be also used to accelerate computations by using OpenMP and MPI technologies. In this case, the whole dataset is split by parts in order to fit the operating memory. Overlap of computations and data transfers between operating memory and hard disks is done by different parallel OpenMP threads.

Performance tests are carried out by making use of the foam data from the previous section. Recall that, the foam data with sizes $(N_\theta, N, N_z) = (130 \cdot 300, 2016, 1800)$ produces the object of sizes $(N, N, N_z, N_t) = (2016, 2016, 1800, 130)$. There is typically no need to recover all time frames by using the proposed algorithm since only particular time frames contain motion artifacts. Thus, for the performance test, we decrease the total number of time frames to a smaller value, $N_t = 8$. The linear binning procedure [49] with reducing data sizes is used to demonstrate scalability and computational complexity of the algorithm. Bins of sizes 4,2, and 1 (no binning) in each spatial direction produce data sizes $(N, N_z) = (504, 450), (N, N_z) = (1008, 900), (N, N_z) = (2016, 1800)$, respectively. Number of basis functions for the function representation (8) in the temporal direction is chosen as $M = 16$ since it demonstrates appropriate quality for reconstruction, see Figure 4. All computations are carried out in single precision. Table I shows average time for one iteration of the algorithm in Figure 2 for different platforms. Graphical processors Tesla P100 and Tesla V100 demonstrate 7-11 performance gain compared to Intel Xeon E5-2650 processor with 12 cores and 24 threads with hyper-threading. Tesla P100 has a fast NVLink connection between CPU and GPU. This type of connection allows bidirectional data transfers with more than 3 times faster speed compared to PCI Express 3.0. Such a fast connection gives almost linear performance gain when utilizing a system with 4 Tesla P100 connected to one CPU, see the last column in Table I. Computational times for the system with 4 Tesla V100 connected to one CPU with PCI Express 3.0 are not presented in the table due to the lack of time scaling caused by high system bus load.

The last observation from Table I is that computational complexity of the algorithm corresponds to the complexity $\mathcal{O}(N_z N^2 \log N)$ of computing the Radon transform with the log-polar-based method. For GPU systems one can observe that 2x increase of the binning size gives approximately 8x decrease in computational times (7-9x in reality due to hardware specific reasons).

VIII. CONCLUSIONS AND OUTLOOK

We derived, validated and applied a new method for reconstruction of four-dimensional tomographic data sets by time-domain decomposition and this way for the first time couple directly space and time domains. Our approach works on continuous acquisition where multiple time frames are recorded through multiple tomographic rotations of π while sample motion is not restricted within the individual intervals. Motion artifacts have been fully suppressed in by selecting the corresponding number of basis functions. The implementation on modern GPU systems demonstrates 7-11 performance gain

TABLE I
AVERAGE TIME (IN SECONDS) FOR 1 ITERATION OF THE PROPOSED ALGORITHM (FIGURE 2) FOR RECONSTRUCTION N_t TIME FRAMES OF SIZES (N, N, N_z) FROM THE MEASUREMENTS OF SIZES (N, N_θ, N_z) BY USING M BASIS FUNCTIONS. CASE: $N_\theta = 2400$, $M = 16$, $N_t = 8$, (N, N_z) ARE CHOSEN WITH RESPECT TO 4,2,1 BINNING.

N, N_z (binning)	Intel Xeon E5-2650 (12 cores/24 threads)	Tesla V100 (PCI Express 3.0)	Tesla P100 (NVLink)	4x Tesla P100 (NVLink)
504, 450 (4)	1.5e+02	1.2e+01	2.1e+01	6.2e+00
1008, 900 (2)	1.1e+03	9.6e+01	1.5e+02	4.8e+01
2016, 1800 (1)	7.7e+03	8.7e+02	1.2e+03	4.2e+02

compared to one modern CPU with 12 cores and 24 threads. Computational times results are acceptable for their use in practice.

The source code is publicly available (https://github.com/math-vrn/rectv_gpu). The foam data with an example script for reconstruction by the proposed method can be downloaded via Tomobank [50], see section Datasets - Dynamic - Foam data.

We assume that the proposed method can be improved in terms of performance. First, the adapted Chambolle-Pock algorithm has $\mathcal{O}(1/N)$ rate of convergence, where N is the iteration number. In our opinion, the convergence rate may probably be increased to $\mathcal{O}(1/N^2)$ with some assumptions on the data structure, that will sufficiently accelerate the whole reconstruction process. Second, different types of representation basis, such as Haar wavelets or Heaviside step functions, may be considered in order to decrease the number of decomposition coefficients, and thus increase the performance. Finally, the parallel implementation on several GPUs may potentially be faster with utilizing new technologies provided by latest versions of NVidia CUDA library. We are going to address all these problems in our further research.

IX. ACKNOWLEDGMENTS

REFERENCES

[1] R. Mokso, C. M. Schlepuetz, G. Theidel, H. Billich, E. Schmid, T. Celcer, G. Mikuljan, L. Sala, F. Marone, N. Schlumpf, and M. Stampanoni, "Gigafrost: the gigabit fast readout system for tomography," *Journal of Synchrotron Radiation*, vol. 24, no. 6, pp. 1250–1259, 2017.

[2] D. S. Lalush and B. M. Tsui, "Block-iterative techniques for fast 4D reconstruction using a priori motion models in gated cardiac SPECT," *Physics in Medicine & Biology*, vol. 43, no. 4, p. 875, 1998.

[3] T.-S. Lee, W. P. Segars, and B. M. Tsui, "Study of parameters characterizing space-time Gibbs priors for 4D MAP-RBI-EM in gated myocardial perfusion SPECT," in *Nuclear Science Symposium Conference Record, 2005 IEEE*, vol. 4. IEEE, 2005, pp. 2124–2128.

[4] E. Gravier, Y. Yang, M. A. King, and M. Jin, "Fully 4D motion-compensated reconstruction of cardiac SPECT images," *Physics in Medicine & Biology*, vol. 51, no. 18, p. 4603, 2006.

[5] V. Van Nieuwenhove, J. De Beenhouwer, J. Vlassenbroeck, M. Moesen, M. Brennan, and J. Sijbers, "Registration based SIRT: A reconstruction algorithm for 4D CT," 2017.

[6] S. Kabus, T. Klinder, K. Murphy, B. van Ginneken, C. Lorenz, and J. P. Pluim, "Evaluation of 4D-CT lung registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2009, pp. 747–754.

[7] M. Chen, W. Lu, Q. Chen, K. J. Ruchala, and G. H. Olivera, "A simple fixed-point approach to invert a deformation field," *Medical physics*, vol. 35, no. 1, pp. 81–88, 2008.

[8] A. Ruhlandt, M. Töpperwien, M. Krenkel, R. Mokso, and T. Salditt, "Four dimensional material movies: High speed phase-contrast tomography by backprojection along dynamically curved paths," *Scientific reports*, vol. 7, no. 1, p. 6487, 2017.

[9] Z. Yang and M. Jacob, "Robust non-local regularization framework for motion compensated dynamic imaging without explicit motion estimation," in *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*. IEEE, 2012, pp. 1056–1059.

[10] X. Jia, Y. Lou, B. Dong, Z. Tian, and S. Jiang, "4D computed tomography reconstruction from few-projection data via temporal non-local regularization," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2010, pp. 143–150.

[11] D. Kazantsev, W. M. Thompson, W. R. Lionheart, G. Van Eyndhoven, A. P. Kaestner, K. J. Dobson, P. J. Withers, and P. D. Lee, "4D-CT reconstruction with unified spatial-temporal patch-based regularization." *Inverse problems and imaging.*, vol. 9, no. 2, pp. 447–467, 2015.

[12] D. Kazantsev, W. R. Lionheart, P. J. Withers, and P. D. Lee, "GPU accelerated 4D-CT reconstruction using higher order pde regularization in spatial and temporal domains," *Proc. CMSSE*, vol. 3, pp. 843–852, 2013.

[13] D. Kazantsev, E. Guo, A. Kaestner, W. R. Lionheart, J. Bent, P. J. Withers, and P. D. Lee, "Temporal sparsity exploiting nonlocal regularization for 4D computed tomography reconstruction," *Journal of X-ray science and technology*, vol. 24, no. 2, pp. 207–219, 2016.

[14] H. Wu, A. Maier, R. Fahrig, and J. Hornegger, "Spatial-temporal total variation regularization (sttvr) for 4D-CT reconstruction," in *Medical Imaging 2012: Physics of Medical Imaging*, vol. 8313. International Society for Optics and Photonics, 2012, p. 83133J.

[15] L. Ritschl, S. Sawall, M. Knaup, A. Hess, and M. Kachelrieß, "Iterative 4D cardiac micro-CT image reconstruction using an adaptive spatio-temporal sparsity prior," *Physics in Medicine & Biology*, vol. 57, no. 6, p. 1517, 2012.

[16] E. Y. Sidky, R. Chartrand, J. M. Boone, and X. Pan, "Constrained TpV Minimization for enhanced exploitation of gradient sparsity: application to CT image reconstruction," *IEEE journal of translational engineering in health and medicine*, vol. 2, pp. 1–18, 2014.

[17] R. Chartrand, E. Y. Sidky, and X. Pan, "Nonconvex compressive sensing for X-ray CT: an algorithm comparison," in *Signals, Systems and Computers, 2013 Asilomar Conference on*. IEEE, 2013, pp. 665–669.

[18] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of mathematical imaging and vision*, vol. 40, no. 1, pp. 120–145, 2011.

[19] K. J. Arrow, L. Hurwicz, H. Uzawa, and H. B. Chenery, "Studies in linear and non-linear programming," 1958.

[20] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, 1992.

[21] B. He and X. Yuan, "On the O(1/n) convergence rate of the Douglas-

Rachford Alternating Direction Method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

[22] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1015–1046, 2010.

[23] M. Benning, F. Knoll, C.-B. Schönlieb, and T. Valkonen, "Preconditioned ADMM with nonlinear operator constraint," in *IFIP Conference on System Modeling and Optimization*. Springer, 2015, pp. 117–126.

[24] N.-Y. Lee and B. J. Lucier, "Wavelet methods for inverting the radon transform with noisy data," *IEEE transactions on image processing*, vol. 10, no. 1, pp. 79–94, 2001.

[25] E. J. Candes and D. L. Donoho, "Curvelets and reconstruction of images from noisy radon data," in *Wavelet applications in signal and image processing VIII*, vol. 4119. International Society for Optics and Photonics, 2000, pp. 108–118.

[26] P. Cerejeiras, M. Ferreira, U. Kähler, and G. Teschke, "Inversion of the noisy radon transform on So (3) by Gabor frames and sparse recovery principles," *Applied and Computational Harmonic Analysis*, vol. 31, no. 3, pp. 325–345, 2011.

[27] F. Colonna, G. Easley, K. Guo, and D. Labate, "Radon transform inversion using the shearlet representation," *Applied and Computational Harmonic Analysis*, vol. 29, no. 2, pp. 232–250, 2010.

[28] A. H. Andersen and A. C. Kak, "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm," *Ultrasonic imaging*, vol. 6, no. 1, pp. 81–94, 1984.

[29] M. Jiang and G. Wang, "Convergence of the simultaneous algebraic reconstruction technique (SART)," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 957–961, 2003.

[30] A. H. Andersen, "Algebraic reconstruction in CT from limited views," *IEEE transactions on medical imaging*, vol. 8, no. 1, pp. 50–55, 1989.

[31] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[32] J. Wu, "On the convergence properties of the EM algorithm," *The Annals of statistics*, pp. 95–103, 1983.

[33] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.

[34] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[35] G. Beylkin, "On the fast fourier transform of functions with singularities," *Applied and Computational Harmonic Analysis*, vol. 2, no. 4, pp. 363–381, 1995.

[36] J. A. Fessler and B. P. Sutton, "Nonuniform fast fourier transforms using min-max interpolation," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 560–574, 2003.

[37] F. Andersson, "Fast inversion of the radon transform using log-polar coordinates and partial back-projections," *SIAM Journal on Applied Mathematics*, vol. 65, no. 3, pp. 818–837, 2005.

[38] F. Andersson, M. Carlsson, and V. V. Nikitin, "Fast algorithms and efficient GPU implementations for the Radon transform and the back-projection operator represented as convolution operators," *SIAM Journal on Imaging Sciences*, vol. 9, no. 2, pp. 637–664, 2016.

[39] S. Basu and Y. Bresler, "$\mathcal{O}(N^2 \log_2 N)$ filtered backprojection reconstruction algorithm for tomography," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1760–1773, 2000.

[40] A. George and Y. Bresler, "Fast tomographic reconstruction via rotation-based hierarchical backprojection," *SIAM Journal on Applied Mathematics*, vol. 68, no. 2, pp. 574–597, 2007.

[41] C. F. Van Loan, "Matrix computations (Johns Hopkins studies in mathematical sciences)," 1996.

[42] C. Raufaste, B. Dollet, K. Mader, S. Santucci, and R. Mokso, "Three-dimensional foam flow resolved by fast X-ray tomographic microscopy," *EPL*, vol. 111, p. 38004, 2015.

[43] "Simultaneous phase and amplitude extraction from a single defocused image of a homogeneous object." *Journal of Microscopy*, vol. 206, no. Pt 1, pp. 33–40, Apr. 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12000561

[44] R. Mokso, F. Marone, S. Irvine, M. Nyvlt, D. Schwyn, K. Mader, G. K. Taylor, H. G. Krapp, M. Skeren, and M. Stampanoni, "Advantages of phase retrieval for fast x-ray tomographic microscopy," *Journal of Physics D: Applied Physics*, vol. 46, no. 49, p. 494004, Dec. 2013. [Online]. Available: http://stacks.iop.org/0022-3727/46/i=49/a=494004?key=crossref.b6573dce28ec3bc1160172bbc2906eec

[45] H. Scherl, B. Keck, M. Kowarschik, and J. Hornegger, "Fast GPU-based CT reconstruction using the common unified device architecture (CUDA)," in *Nuclear Science Symposium Conference Record, 2007. NSS'07. IEEE*, vol. 6. IEEE, 2007, pp. 4464–4466.

[46] N. Gac, S. Mancini, M. Desvignes, and D. Houzet, "High speed 3D tomography on CPU, GPU, and FPGA," *EURASIP Journal on Embedded systems*, vol. 2008, p. 5, 2008.

[47] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, J. Batenburg, and J. Sijbers, "Fast and flexible X-ray tomography using the ASTRA toolbox," *Optics express*, vol. 24, no. 22, pp. 25 129–25 147, 2016.

[48] D. Foley and J. Danskin, "Ultra-performance Pascal GPU and NVLink interconnect," *IEEE Micro*, vol. 37, no. 2, pp. 7–17, 2017.

[49] S. Handrick, B. Naimipour, D. Raicu, and J. Furst, "Evaluation of binning strategies for tissue classification in computed tomography images," in *Medical Imaging 2006: Image Processing*, vol. 6144. International Society for Optics and Photonics, 2006, p. 61444L.

[50] F. De Carlo, D. Gürsoy, D. J. Ching, K. J. Batenburg, W. Ludwig, L. Mancini, F. Marone, R. Mokso, D. M. Pelt, J. Sijbers *et al.*, "Tomobank: a tomographic data repository for computational x-ray science," *Measurement Science and Technology*, vol. 29, no. 3, p. 034004, 2018.