Tomographic Reconstruction of Dynamic Features with Streaming Sliding Subsets

Tekin Bicer*†§, Viktor Nikitin†, Selin Aslan†, Doğa Gürsoy†, Rajkumar Kettimuthu*§, Ian T. Foster*‡

*Data Science & Learning Division and †X-Ray Science Division, Argonne National Laboratory
{tbicer, vnikitin, saslan, dgursoy, kettimuthu, foster}@anl.gov
‡Department of Computer Science and §Consortium for Advanced Science and Engineering, University of Chicago

***Abstract*—As the sophistication and speed of today's X-ray experiments grow, collecting the most informative data has become ever more relevant, necessitating the development of algorithms that can provide good quality reconstructions from tomographic data streams. However, almost all conventional reconstruction systems and algorithms work exclusively offline, requiring that complete datasets be collected and available before they can be processed.Further, these systems and algorithms provide limited consideration and support for challenging experiments, such as imaging samples with dynamic features, where both spatial and temporal properties of the features rapidly change.**

We describe here a high-performance runtime system for analyzing and reconstructing streaming tomography datasets using sliding subsets of projection images, and evaluate the reconstruction quality of dynamic features with respect to different runtime configuration parameters using phantom and real-world tomography datasets. Our system enables runtime system parameters to be adjusted dynamically over the course of experiment, providing opportunities for balancing the quality and computational demands of tasks, better observation of phenomena, and improving advanced experimental techniques such as autonomous experimental steering.

***Index Terms*—Tomography, stream processing, dynamic features, image reconstruction**

I. Introduction

Synchrotron light sources are crucial tools for addressing grand challenge problems in life sciences, energy, climate change, and information technology [1, 2]. High-quality, timely data analysis and feedback are not only crucial for advancing these sciences but also important for reliable experimentation, correct data acquisition, and efficient utilization of instruments. Some synchrotron radiation experiments can have extremely complex setups and target samples, and may require advanced imaging techniques, such as the imaging of samples with dynamic features [3] using high-speed time-resolved tomography [4]. Current data analysis pipelines for such experiments run only after data acquisition is completed, and on small, often local computers with preset configuration parameters. The resulting delay between data collection and (*offline*) analysis, together with predefined runtime parameters, prevents high-quality data acquisition and real-time experimental decision making, compromising timely insights on collected data and researcher productivity.

The need for analysis software that can accommodate large-scale experimental data will increase significantly in the near future. For example, tomography experiments at Advanced Photon Source (APS) imaging beamlines can produce data at more than 8 Gbps today, while generating hundreds of raw measurements (projections) per second [5]; the imminent APS upgrade will provide x-rays with 100–1000 times more brightness compared to today's light source [1, 2], which are expected to increase the data generation rate at micro-computed tomography (microCT) beamlines by at least 2–3 times compared to current rates.

Tomographic reconstruction, typically the main processing step for tomography experiments, may be performed with *analytical* or *iterative* techniques. The two classes of technique can be compared as follows: Analytical techniques, e.g., filtered-backprojection, are computationally more efficient, but are typically more sensitive to noise and environmental factors, such as dose, vibration, drifts, and motion. Further, they more data for good quality reconstructions. In contrast, iterative techniques can use statistical models that can work with limited data and are more resilient to noise [6], but are computationally more expensive due to their iterative nature [7, 8].

Existing tools have been developed for analysis of x-ray tomographic data [9, 10] provide sufficient capabilities for small- to medium-scale *offline* data analysis, but do not support the high-performance reconstruction required for large datasets. Furthermore, they provide limited or no support for analysis of streaming tomography datasets with dynamic features, such as a sample with foam features that expand (or shrink) and appear (or disappear) over a period of time [11, 12]. The (real-time) analysis of such datasets is challenging for several reasons. First, since the spatio-temporal properties of the sample change over time, the whole dataset does not represent a stable state of the features and therefore regular reconstruction approaches cannot be applied directly. Second, if the experimentation is long running, the collected dataset can be extremely large (hundreds of TBs or even PB-scale in some cases) and impossible to analyze as a whole.

In the work reported here, we explore a new approach

This material was partially supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Sciences, under Contract DE-AC02-06CH11357. We gratefully acknowledge the computing resources provided on Bebop (and/or Blues), a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory.

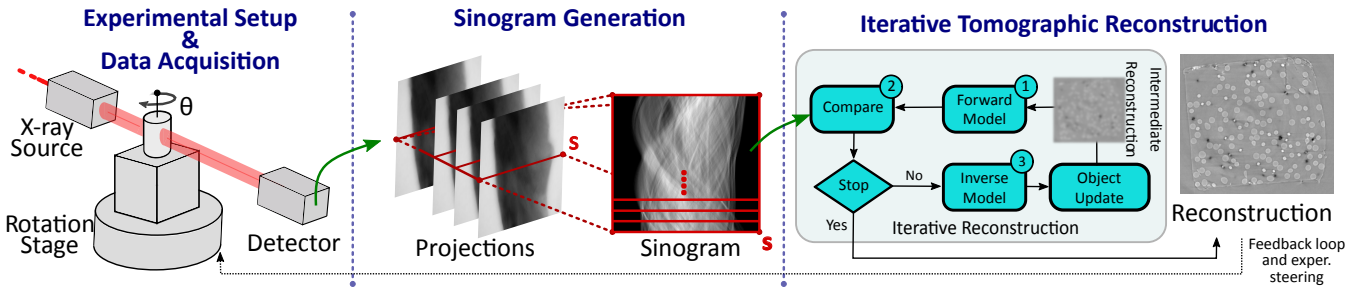

Fig. 1. Schematic of a typical tomographic data acquisition and reconstruction pipeline. The experiment can be controled with a feedback loop after analyzing reconstructed image.

to the reconstruction of samples with dynamic features that allows for streaming analysis with fine-grained control, via setting of configuration parameters, over both reconstruction quality and computational costs. We describe the configuration parameters and show their effect on both reconstruction quality and computational costs. The runtime system parameters can be be adjusted dynamically during the experimentation, providing opportunities for balancing the quality and computational requirements of tasks, better observation of phenomena, and improving advanced experimentation techniques such as autonomous experimental steering.

II. Background

We first introduce the computed tomography (CT) image analysis pipeline used at synchrotron light sources and then discuss the dynamic feature imaging and reconstruction.

A. Tomographic Data Acquisition and Reconstruction

In Fig. 1, we present the tomography experimental setup and its corresponding data acquisition process, sinogram generation, and reconstruction stages. During a tomography experiment, the target sample is placed on a rotation stage and illuminated by an X-ray source. As generated X-rays pass through the sample, they attenuate according to the thickness and density of the target sample. For high-density samples, X-ray attenuates more and therefore results in low readings compared to low-density samples. The photons are then measured by a photon detector.

The corresponding measurement is called a *projection*. During a tomography experiment, a set of projections is collected from different rotations, θ, with typically a fixed exposure time for each. An ideal experiment collects projections that fully cover the sample.

The attenuation of X-ray intensity is modeled according to Beer-Lambert law, $I_\theta(s) = I_0(s)\exp[-p_\theta(s)]$, where $I_0(s)$ is the incident x-ray illumination on the sample and $I_\theta(s)$ are the collected measurements at a number of θs, as a result of a tomographic scan. $p_\theta(s)$ represents a cross section of projections (shown in red in the central section of Fig. 1), known as a *sinogram*. For parallel beam geometry, measurements in a sinogram correspond to a cross section of the target sample.

The tomographic reconstruction process aims to recover 2D cross section images of a sample from their corresponding sinograms. Iterative reconstruction techniques aim to solve $\hat{x} = \underset{x \in C}{\operatorname{argmin}} \|y - Ax\|^2 + R(x)$, where $\hat{x}$ is the reconstructed tomogram, A is the forward model, y is the sinogram, $R(x)$ is a regularizer functional, x is the search variable, and C is a constraint on x.

Iterative reconstruction techniques consist of three steps, as depicted in the last phase of Fig. 1. First, a forward model is applied to an intermediate image estimate in order to find a measurement. Then, the estimated and real measurements are compared. Finally, the estimated image is updated according to the difference between the real and estimated measurements. These steps are repeated until a user-defined constraint is met, such as total number of iterations or error threshold.

Iterative techniques can provide better reconstructions compared to analytical techniques for dynamically evolving features [12, 13]. They are also more suitable for real-time reconstruction of samples with limited data, if computational resources are sufficient [14, 15].

III. Our Configurable Reconstruction System

We now introduce our high-performance system for reconstructing streaming tomography data with dynamic features in real-time and explain its configuration parameters.

A. System Design

Our data analysis pipeline, shown in Fig. 2 consists of two components: *load balancer* and *runtime engine*. The load balancer chunks the incoming (or generated) projections to sinograms and forwards them to their corresponding processes; the runtime engine reconstructs the incoming projections according to parameters set by the user. We further explain these components in the following sections.

Load Balancer: As shown in Fig. 2, the load balancer (LB) receives projections from experimental setup. Each incoming projection is split according to the total number of rows and processes in the system. For example, if the dimensions of an incoming projection is (2048, 2048) pixels and there are 4 running processes, then the projection is split into 4 chunks where each chunk's dimension is (512, 2048). Note that the chunking is based on the rows, so that each process receives the same sequence of rows (sinograms) throughout the processing. Load balancer uses synchronous communication (with ZMQ) to guarantee all the rows are orderly delivered to processes. Our

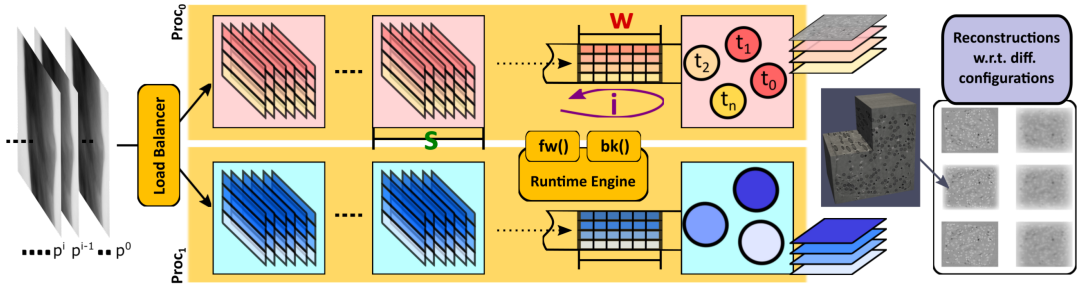

Fig. 2. The components of high-performance runtime system for streaming tomography dataset. Window size, step size and number of iterations are shown with w, s and i, respectively. *fw()* and *bk()* functions represent forward and backprojection operators.

load balancer can also perform light weight preprocessing steps, such as dark and white field normalization, though these steps can easily be pushed to reconstruction processes.

Runtime Engine: After chunking the incoming projections, they are forwarded to the runtime engine. This component performs the computationally demanding tasks for the reconstruction, in particular *forward* and *backprojection* operations (shown as **fw()** and **bk()** in Fig. 2). Runtime system provides several preimplemented iterative reconstruction algorithms, including simultaneuous iterative reconstruction technique (SIRT), Maximum-likelihood Expectation Maximization (MLEM) and Penalized Likelihood Method (PML), that are ported from the TomoPy Python library [9]. The runtime engine also exposes several APIs for users to implement their own algorithms. Specifically, it uses reduction-based processing structure for high-performance implementation of forward and backprojection operators [16]. This processing structure mimics that of MapReduce [17–20], where users can plug-and-play their algorithms, and that scales efficiently to tens of thousands of cores [21, 22].

The runtime system processes use two levels of parallelization: process and thread level. The process-level parallelization, based on MPI, is used for coarse-grained load balancing as mentioned in LB. Thread-level parallelization, on the other hand, performs fine-grained shared-memory parallelization. These two levels of parallelization enables the runtime system to perform efficient *local* thread-level and *global* process-level reduction and synchronization operations.

Besides the number of processes and threads in the system, several configuration parameters and data structures are exposed to users. These configuration parameters can be used to adjust both computational demands and the quality of the reconstructed images. We focus here on the following three parameters, depicted in Fig. 2.

- Window size, w, is the number of projections used in each reconstruction. This determines the size of a sliding (window) buffer in the runtime system to store incoming projections. For example, if $w = 32$, only the last 32 projections are stored and used for reconstruction.
- Step size, s, is how often reconstructions are computed. The reconstruction operations are triggered according to this parameter. For example, if $s = 1$ then the reconstruction is initiated after every projection. However, if $s = 2$ then reconstruction is triggered after every other projection. Note that the runtime system uses all projections in its buffer.
- Iterations, i, is the number of iterations in a reconstruction. This parameter determines how many times the forward and backprojection operators are applied to the object function.

Different (w, s, i) combinations are appropriate for dynamic data with different spatio-temporal properties. We evaluate these parameters and show their resulting reconstructions with different dynamic features in the following section.

IV. Experimental Results

We evaluate our system in the context of the parallelized simultaneous iterative reconstruction technique (SIRT) [9]. We use SIRT to reconstruct two different datasets for a range of different runtime configuration parameters, and report on how different configuration parameter values effect the quality of the resulting reconstructed image.

The two datasets that we use to evaluate our system are (1) a phantom dataset, and (2) a foam dataset collected at the MAX IV Laboratory of Lund University, Sweden using fast X-ray tomographic microscopy [11, 23, 24]. Each dataset contains features of different sizes and shapes that change over time in different ways (e.g., grow, shrink, move) and at different rates. These different properties can introduce varying artifacts in reconstructions.

The reconstructed image quality depends on the rate of change in features. Since the reconstruction algorithms require sufficient data to recover features, reconstructions with dynamic features require more measurements compared to samples with static features.

We reconstructed images by using the Trace high-performance computing middleware [16, 21] for different sets of configuration parameter (w, s, i), where both w and s ranged from 1 to 32, and i ranged from 1 to 10. Computational costs scale roughly linearly with each of w, s^{-1}, and i; thus configuration (32, 1, 32) involves roughly 32^3 more computation than configuration (1, 32, 1).

A. Phantom Dataset

The phantom dataset comprises 6480 projections—36 full rotations, each with 180 projections, spaced equally in time—

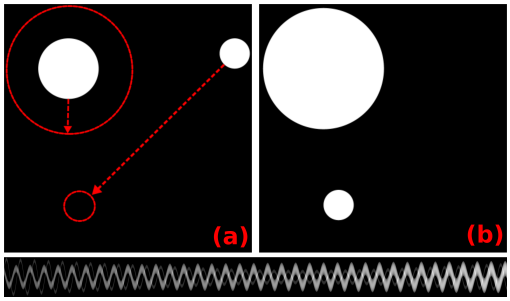

Fig. 3. Two 2-D slices from the phantom dataset ground truth. (a) shows the initial position of the circles (projection 0), and their motion; (b) shows their final location (projection 6479). The bottom figure shows the 6480-sinogram dataset corresponding for these slices.

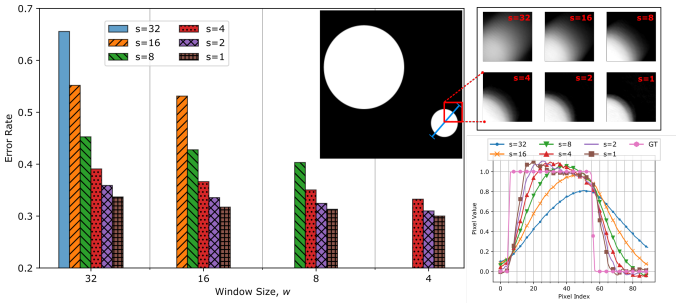

Fig. 4. Impact of configuration parameters on reconstruction errors for the *moving circle*, for $i = 1$ and different w and s. Upper middle shows ground truth for projection 3240; blue line indicates the pixel values that are used to compute error rates: see text. Right top shows reconstructions with $w = 32$. Right bottom shows how pixel values change for the same configuration and also with ground truth.

generated synthetically from a simulated environment containing two circular features, one smaller and one larger, which expand and move over time, respectively, as shown in Fig. 3. The number of pixels in each projection row is 512, resulting in a reconstructed image slice of size 512×512. The rate of change in both features is constant; that is, the speeds of expansion and movement are fixed during the simulation. The radius of the expanding circle changes from ∼51 to ∼102 pixels during the 6480-projection data acquisition process: a rate of 51/6480 = 7.87×10^{-3} pixel changes per projection acquisition. The second circle moves ∼375.46 pixels: a rate of 375.46/6480 = 5.79×10^{-2} pixels per projection.

We first study the impact of configuration parameters on reconstruction error for the small (moving) circle. Fig. 4 (left) shows errors for $w \in \{4, 8, 16, 32\}$ and $s \in \{1, 2, 4, 8, 16, 32\}$, with $i = 1$ in all cases. Since w determines the number of projections that are used for reconstruction and the runtime system works with only limited data at any given time, the reconstructions (moving features and background) are exposed to noise. Consequently, conventional quality metrics such as SSIM, PSNR and SNR cannot provide reliable results. Instead, we compute error as $\frac{\|x-\hat{x}\|_2}{\|\hat{x}\|_2}$, where x and $\hat{x}$ are from the reconstructed and ground truth pixel values on a line with direction aligned with the feature's trajectory and length (shown in blue in the figure). The lengths of the lines are selected according to the features' artifacts and set to ∼90 and

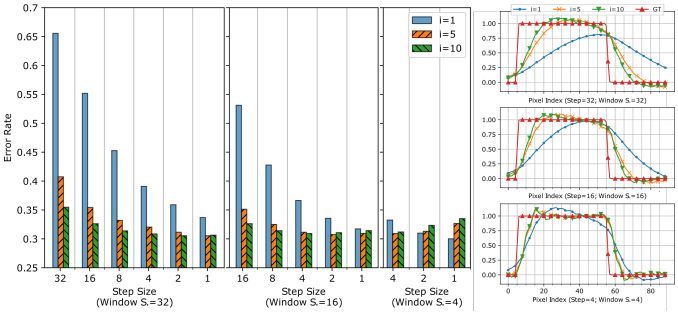

Fig. 5. Error rates of small *moving* circle with respect to number of iterations. The error rates are based on line e in Fig. 4. All configuration parameters are varied. Right figures show the line profiles of e for given configuration parameters.

∼70 pixels for small (moving) and large (expanding) circles, respectively.

We see that error rates improve with decreasing w for all configurations, indicating that smaller window sizes provide better spatio-temporal information and thus higher image quality. Note that as computational costs decrease linearly with w, smaller w also reduces computational requirements. For each value of w, smaller s significantly improves error rate. For instance, in the (32, $s \leq 32$, 1) configurations, error rates range over 33.6–65.6%, with (32, 1, 1) having the lowest error. In the right top of Fig. 4, we show the reconstructed images with (32, $s \leq 32$, 1) for projection 3240; in the bottom right, we show the values along the blue line for different configurations, plus ground truth (GT) values in pink. As expected, GT changes rapidly between empty space and circle, reflecting the change from pixel value 0 to 1. The $s = 1$ configuration matches GT most closely and $s = 32$ the least.

So far we have considered only configurations with $i = 1$. Fig. 5 shows the effect of increased iterations on small circle reconstruction quality. We consider $i \in \{1, 5, 10\}$ and a range of w and s values. The relative errors in range over 30–65.5%, with the lowest and highest being for the (4, 1, 1) and (32, 32, 1) configurations, respectively. This indicates that for this moving circle feature, the best reconstruction quality is delivered with frequent updates on the reconstructed object using a small number of projections and iterations. We also observe that increased iterations benefit the larger window sizes the most. For instance, the (32, 1, 10) configuration has a 31% relative error compared to GT, i.e., less than 1% more than the best configuration (4, 1, 1). The line profiles in the figure show the effect of iterations on large window size and step sizes, in which increased iterations result in the closest match to GT (red). Note that, the same observation is not true for the small window sizes, i.e. (4, 1, $i \leq 10$), in which the reconstruction quality drops with increasing number of iterations. This shows that more computation does not always translate to better reconstructions.

Next we study the large (expanding) circle. Figs. 6 and 7 show results for $i = 1$ and varying w and s values. As this circle expands throughout the simulation, the reconstruction parameters provide slightly different quality results compared to the *moving* small circle.

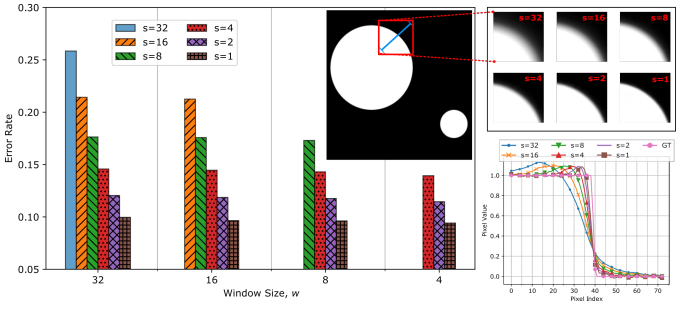

Fig. 6. Error rates and line profiles of big *expanding* circle with respect to window size (w) and step size (s) with one iteration ($i = 1$).

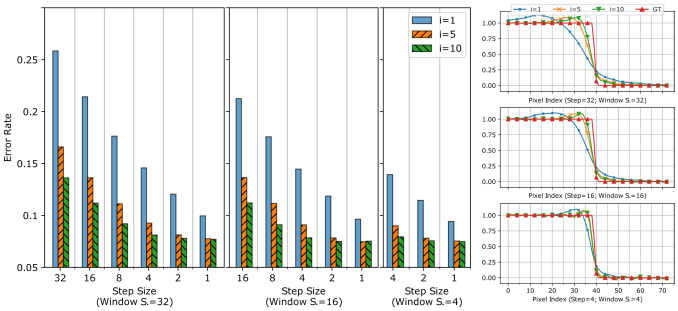

Fig. 7. Error rates and line profiles of big *expanding* circle with respect to different i when w=(32, 16, 4) and ($s \leq 32$).

Fig. 6 shows how error rates vary with w and s when $i = 1$. The error rates range from 25.8% to 7.5%, with configurations (32, 32, 1) and (4, 1, 1) providing the worst and best reconstruction qualities, respectively. As in the small circle case, smaller w and s values enhance quality. For example, (32, 1, 1) reduces reconstruction error from 25.8% to 9.96% relative to (32, 32, 1). The line profiles in Fig. 6 illustrate the effect of s on pixel values. As in the small circle case, configuration (32, 1, 1) shows the closest correlation with GT.

When we also vary i, we observe a different behavior than in the small *moving* circle case. The large *expanding* circle, in contrast to the small moving circle, benefits from additional iterations even for small window sizes. For instance, the best configuration (4, 1, 10), has an error 20.2% better than that of (4, 1, 1): 7.5% rather than 9.4%. Similarly, the error for (16, 16, 10) is only 11.2%, while that for (16, 16, 1) is 21.3%. Increased iterations also decrease the error rate from 10% for (32, 1, 1) to 7.7% and (32, 1, 10): on par with the quality configuration. In summary, large *expanding* circle benefits from more iterations in *all* configurations, unlike the *moving* small circle (4, 1, $*$) configurations.

Overall, two key observations from these experiments are (i) more computation does not always improve reconstruction quality, e.g., (4, 1, 1) vs. (4, 1, 10) in *moving* small circle; and (ii) the effect of configuration parameters depends on the features' spatio-temporal properties, e.g., the best qualities for the *moving* small circle and *expanding* large circle are observed with (4, 1, 1) and (4, 1, 10) configurations, respectively.

B. Foam Dataset

The second dataset is a real-world experimental dataset, in which complex liquid foam features both expand (or shrink) and appear (or disappear), depending on the environmental conditions. During the experiment, the rate of changes in features, and thus reconstruction difficulty, increases as the experiment progresses.Projections are collected at 840°/sec with 0.7 ms exposure time, resulting in 130 rotations—each representing a time frame. The resulting dataset has dimensions (2016, 300×130, 1800), where 300 is the number of projections over $[0, \pi)$. The resulting dataset has size 590 GB.

As with the first dataset, we reconstruct the foam dataset with different parameter configurations. Lacking ground truth, we present the reconstructed images, along with their line profiles. Fig. 8 shows our reconstructions. The left box shows the reconstructions with respect to time. Specifically, the left image shows the reconstruction of a slice from the whole foam dataset at the beginning of the experiment with (32, 1, 1). Since the motion of the features in the sample is limited, we do not observe much artifact. The next image shows the reconstructions of a specific region (red rectangle) from the same sample with different configuration parameters as the experiment progress. Specifically, the y-axis shows the system configuration, similar to previous set of experiments, and the x-axis shows time, e.g., *2t/4* and *t* represent the reconstructions in the middle and at the end of the experiment, respectively. For all configurations, as the experiment progresses the motion artifacts increase. The (32, 32, 1) configuration is affected more than the others. The main reason of this is this configuration performs a single pass over the incoming projection stream and therefore carries (or copies) previously reconstructed features more compared to other configurations. Focusing on the (32, 1, 1) configuration, we see that the features are easier to distinguish; however, there are still some artifacts due to the fast moving features.

We further analyze the effect of motion artifacts with different configurations in the right box in Fig. 8. Complementary to the experiments in left box, we focus more on the number of iterations. We only show the reconstructions at the end of the experiment (*t* time), since this is when the maximum amount of motion artifacts are encountered. We see that each configuration presents different properties. For example, if we compare (32, 32, $*$) configurations, first column, we see that the number of iterations significantly help reconstruction quality. On the other hand, the more number of iterations starts introducing salt-and-pepper like artifacts for (32, 1, $*$) configurations. The main reason for this is that the total number of updates is 32x more when we compare (32, 1, $*$) to (32, 32, $*$) (because of the step size parameter s). Iterations further increase the number of updates and contribute to the salt-and-pepper artifact, e.g., configuration (32, 1, 10) performs $32 \times 10 = 320$ more updates than the configuration (32, 32, 1).

The line profiles of each iteration configuration is given in the right most figure (organized with respect to rows). The profiles show the pixel values of the lines on images. The

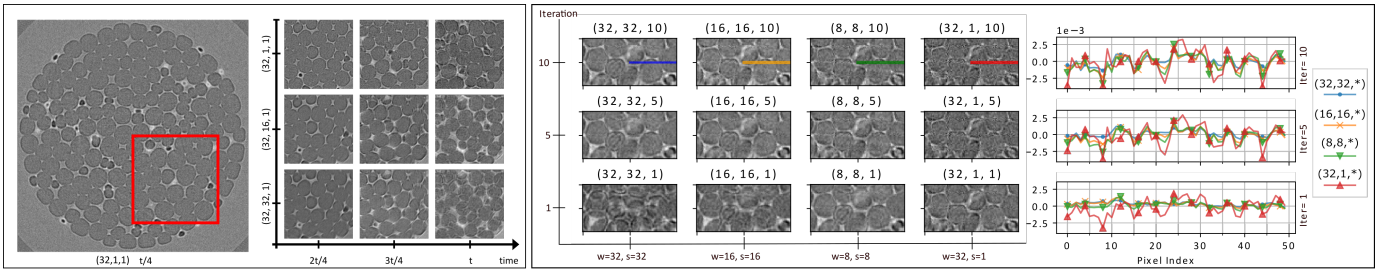

Fig. 8. Reconstruction results for foam dataset. Left: Reconstructions of a single slice with respect to time and different configuration parameters (for the region specified with red rectangle). Right: Effect of iterations with line profiles.

lines are color coded to represent each (w, s) on x-axis. As the number of iterations increases the contrast between the features and the background increases, these can be observed with the maximum and minimum pixel values. We also see artifacts with high i and low s values similar to previous set of experiments.

C. Understanding the Computational Requirements

The computational complexities of the reconstruction operations vary depending on the selected configuration parameters. In general, w and i have a linear effect on computation (c), whereas s is inversely proportional: $c \propto (w \times i)/s$. Although higher computational operations typically translate to better reconstructions, this is not guaranteed. For example, (4, 1, 1) provides the best quality for the large *expanding* circle in the phantom dataset, which, in turn, requires almost 80× less computation compared to (32, 1, 10).

Our stream reconstruction system enables users to balance between computational cost (system responsiveness) and reconstruction quality so that users can adjust the parameters depending on the goal of the application.

V. Related Work

Reconstruction of dynamic features from tomography datasets has been an active research area [25–27]. Nikitin et al. developed a tomographic reconstruction method that decomposes dynamic tomography datasets in the temporal domain by projecting to a lower dimensional subspace of basis functions and deploying an additional L1 regularization technique [12]. Mohan et al. introduced a model-based iterative reconstruction method that can recover time-resolved volumes from 4D tomography datasets with interlaced view sampling [13, 28]. Complementary to these efforts, we evaluate the spatio-temporal properties of dynamic features and the reconstruction performance with our high-performance runtime system and its configuration parameters.

Large-scale experimental data generation at synchrotron light sources and their unique and challenging experimental conditions make it extremely difficult to develop efficient and generic software tools; therefore, many projects have been initiated to address the data management problems in the context of light sources. The Center for Advanced Mathematics for Energy Research Applications (CAMERA), which is lead by LBNL, has initiated several projects to address these issues [29, 30]. Similarly, other synchrotron light sources have adapted or started developing their own visualization, workflow management and analysis tools, including [31] from ANL and [32] from Diamond, ESRF and EMBL. Among these efforts, real-time high-performance tomographic reconstruction frameworks and enhancement techniques have received significant attention [33–38]. Our work utilizes Trace framework [16, 21] that performs high-performance reconstruction techniques using fine-grained parallelization and reduction-based processing structure.

Iterative reconstruction approaches have been successfully used on many tomography data [39–43]. Although, the computational requirements of iterative approaches are much more demanding than the analytical counterparts, advanced parallelization techniques, coupled with large-scale clusters and supercomputers, enabled their usage on medium to large experimental datasets [44–48].

VI. Conclusions

We have evaluated the effect of runtime system parameters on reconstruction quality of streaming tomographic data from experiments with dynamic features. Working with a synthetic phantom and an experimental foam sample, we varied three runtime system configuration parameters, *window size*, *step size*, and number of *iterations*, to generate different reconstructions from the same data. We find that these parameters have significant effect on reconstruction quality, and conclude that it is important to select the correct configuration parameters for a target feature's spatio-temporal properties. Specifically, we see that smaller step sizes can improve the reconstruction quality when the window size and the number of iterations are small. This information can be used during reconstruction when it is not feasible to collect sufficient data from the sample, e.g. dose sensitive biological samples. We also see that increasing the number of iterations can compensate the small step sizes and improve the image quality when the window size is medium to large. We observe that depending on the properties of a feature (and its motion), a configuration with less computational demand can produce better reconstruction quality.

Our system enables configuration parameters to be adjusted dynamically over the course of experiment, providing opportunities for balancing reconstruction quality and computational costs to achieve better observations of phenomena. It can be used for improving advanced experimentation techniques such as autonomous experimental steering.

REFERENCES

[1] Argonne National Laboratory, "Advanced Photon Source, An Office of Science National User Facility." https://www.aps.anl.gov. [Accessed: June 2019].

[2] "APS Science 2018," Tech. Rep. ANL-18/40, ISSN 1931-5007, Advanced Photon Source, Argonne National Laboratory, January 2019.

[3] T. dos Santos Rolo, A. Ershov, T. van de Kampa, and T. Baumbacha, "In vivo x-ray cine-tomography for tracking morphological dynamics," *PNAS*, vol. 111, no. 11, pp. 3921–3926, 2013.

[4] E. Maire and P. J. Withers, "Quantitative x-ray tomography," *International Materials Reviews*, vol. 59, no. 1, pp. 1–43, 2014.

[5] FLIR Systems, "Oryx 10GigE Detector." https://www.flir.com/products/oryx-10gige. [Accessed: May 2019].

[6] J. Nuyts, B. De Man, J. A. Fessler, W. Zbijewski, and F. J. Beekman, "Modelling the physics in the iterative reconstruction for transmission computed tomography," *Physics in Medicine & Biology*, vol. 58, no. 12, p. R63, 2013.

[7] M. Hidayetoğlu, T. Biçer, S. G. De Gonzalo, B. Ren, D. Gürsoy, R. Kettimuthu, I. T. Foster, and W.-m. W. Hwu, "Memxct: Memory-centric x-ray ct reconstruction with massive parallelization," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–56, 2019.

[8] X. Wang, A. Sabne, P. Sakdhnagool, S. J. Kisner, C. A. Bouman, and S. P. Midkiff, "Massively parallel 3d image reconstruction," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–12, 2017.

[9] D. Gürsoy, F. De Carlo, X. Xiao, and C. Jacobsen, "TomoPy: a framework for the analysis of synchrotron tomographic data," *Journal of synchrotron radiation*, vol. 21, no. 5, pp. 1188–1193, 2014.

[10] W. van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers, "The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography," *Ultramicroscopy*, vol. 157, pp. 35–47, 2015.

[11] C. Raufaste, B. Dollet, K. Mader, S. Santucci, and R. Mokso, "Three-dimensional foam flow resolved by fast x-ray tomographic microscopy," *EPL (Europhysics Letters)*, vol. 111, p. 38004, aug 2015.

[12] V. V. Nikitin, M. Carlsson, F. Andersson, and R. Mokso, "Four-dimensional tomographic reconstruction by time domain decomposition," *IEEE Transactions on Computational Imaging*, vol. 5, no. 3, pp. 409–419, 2019.

[13] 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 Transactions on Computational Imaging*, vol. 1, no. 2, pp. 96–111, 2015.

[14] T. Bicer, D. Gürsoy, R. Kettimuthu, F. De Carlo, and I. T. Foster, "Optimization of tomographic reconstruction workflows on geographically distributed resources," *Journal of synchrotron radiation*, vol. 23, no. 4, pp. 997–1005, 2016.

[15] X. Wang, A. Sabne, S. Kisner, A. Raghunathan, C. Bouman, and S. Midkiff, "High performance model based image reconstruction," *ACM SIGPLAN Notices*, vol. 51, no. 8, pp. 1–12, 2016.

[16] T. Bicer, D. Gürsoy, V. D. Andrade, R. Kettimuthu, W. Scullin, F. D. Carlo, and I. T. Foster, "Trace: A high-throughput tomographic reconstruction engine for large-scale datasets," *Advanced Structural and Chemical Imaging*, vol. 3, p. 6, Jan 2017.

[17] Y. Wang, G. Agrawal, T. Bicer, and W. Jiang, "Smart: A mapreduce-like framework for in-situ scientific analytics," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '15, (New York, NY, USA), Association for Computing Machinery, 2015.

[18] E. Dede, M. Govindaraju, D. Gunter, R. S. Canon, and L. Ramakrishnan, "Performance evaluation of a mongodb and hadoop platform for scientific data analysis," in *Proceedings of the 4th ACM workshop on Scientific cloud computing*, pp. 13–20, 2013.

[19] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: a runtime for iterative mapreduce," in *Proceedings of the 19th ACM international symposium on high performance distributed computing*, pp. 810–818, 2010.

[20] Z. Fadika, E. Dede, M. Govindaraju, and L. Ramakrishnan, "Mariane: Using mapreduce in hpc environments," *Future Generation Computer Systems*, vol. 36, pp. 379–388, 2014.

[21] T. Bicer, D. Gursoy, R. Kettimuthu, F. De Carlo, G. Agrawal, and I. T. Foster, "Rapid tomographic image reconstruction via large-scale parallelization," in *Euro-Par 2015: Parallel Processing*, pp. 289–302, Springer, 2015.

[22] T. Bicer, *Supporting Data-Intensive Scientic Computing on Bandwidth and Space Constrained Environments*. PhD thesis, The Ohio State University, 2014.

[23] 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.

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

[25] F. Lucka, N. Huynh, M. Betcke, E. Zhang, P. Beard, B. Cox, and S. Arridge, "Enhancing compressed sensing 4d photoacoustic tomography by simultaneous motion estimation," *SIAM Journal on Imaging Sciences*, vol. 11, no. 4, pp. 2224–2253, 2018.

[26] M. Burger, H. Dirks, and C.-B. Schonlieb, "A variational model for joint motion estimation and image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 11, no. 1, pp. 94–128, 2018.

[27] G. Van Eyndhoven, K. J. Batenburg, D. Kazantsev, V. Van Nieuwenhove, P. D. Lee, K. J. Dobson, and J. Sijbers, "An iterative ct reconstruction algorithm for fast fluid flow imaging," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4446–4458, 2015.

[28] K. A. Mohan, S. Venkatakrishnan, J. W. Gibbs, E. B. Gulsoy, X. Xiao, M. De Graef, P. W. Voorhees, and C. A. Bouman, "4d model-based iterative reconstruction from interlaced views," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 783–787, IEEE, 2015.

[29] R. J. Pandolfi, D. B. Allan, E. Arenholz, L. Barroso-Luque, S. I. Campbell, T. A. Caswell, A. Blair, F. De Carlo, S. Fackler, A. P. Fournier, *et al.*, "Xi-cam: a versatile interface for data visualization and analysis," *Journal of synchrotron radiation*, vol. 25, no. 4, pp. 1261–1270, 2018.

[30] B. J. Daurer, H. Krishnan, T. Perciano, F. R. Maia, D. A. Shapiro, J. A. Sethian, and S. Marchesini, "Nanosurveyor: a framework for real-time data processing," *Advanced structural and chemical imaging*, vol. 3, no. 1, pp. 1–10, 2017.

[31] J. M. Wozniak, T. G. Armstrong, M. Wilde, D. S. Katz, E. Lusk, and I. T. Foster, "Swift/t: Large-scale application composition via distributed-memory dataflow processing," in *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, pp. 95–102, IEEE, 2013.

[32] M. Basham, J. Filik, M. T. Wharmby, P. C. Chang, B. El Kassaby, M. Gerring, J. Aishima, K. Levik, B. C. Pulford, I. Sikharulidze, *et al.*, "Data analysis workbench (dawn)," *Journal of synchrotron radiation*, vol. 22, no. 3, pp. 853–858, 2015.

[33] D. Y. Parkinson, K. Beattie, X. Chen, J. Correa, E. Dart, B. J. Daurer, J. R. Deslippe, A. Hexemer, H. Krishnan, A. A. MacDowell, *et al.*, "Real-time data-intensive computing," in *AIP Conference Proceedings*, vol. 1741, p. 050001, AIP Publishing LLC, 2016.

[34] J.-W. Buurlage, F. Marone, D. M. Pelt, W. J. Palenstijn, M. Stampanoni, K. J. Batenburg, and C. M. Schlepütz, "Real-time reconstruction and visualisation towards dynamic feedback control during time-resolved tomography experiments at tomcat," *Scientific Reports*, vol. 9, no. 1, pp. 1–11, 2019.

[35] T. Bicer, D. Gursoy, R. Kettimuthu, I. T. Foster, B. Ren, V. De Andrede, and F. De Carlo, "Real-time data analysis and autonomous steering of synchrotron light source experiments," in *2017 IEEE 13th International Conference on e-Science (e-Science)*, pp. 59–68, IEEE, 2017.

[36] R. Kettimuthu, Z. Liu, T. Bicer, and I. Foster, *Cyberinfrastructure and System Software for Online Analysis of Large-Scale Data: Challenges and Design Choices*, ch. Chapter 16, pp. 333–360.

[37] Z. Liu, T. Bicer, R. Kettimuthu, and I. Foster, "Deep learning accelerated light source experiments," in *2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*, pp. 20–28, IEEE, 2019.

[38] Z. Liu, T. Bicer, R. Kettimuthu, D. Gursoy, F. De Carlo, and I. Foster, "Tomogan: Low-dose synchrotron x-ray tomography with generative adversarial networks," *arXiv preprint arXiv:1902.07582*, 2019.

[39] D. Y. Parkinson, D. Gursoy, D. M. Pelt, S. Venkatakrishnan, R. Archibald, K. A. Mohan, T. Bicer, M. Vogelgesang, J. Sethian,

N. Wadeson, M. Basham, and T. Farago, *Tomographic Reconstruction for Synchrotron Tomography*, ch. Chapter 4, pp. 65–82.

[40] D. J. Duke, A. B. Swantek, N. M. Sovis, F. Z. Tilocco, C. F. Powell, A. L. Kastengren, D. Gürsoy, and T. Biçer, "Time-resolved x-ray tomography of gasoline direct injection sprays," *SAE International Journal of Engines*, vol. 9, no. 1, pp. 143–153, 2016.

[41] D. Gürsoy, T. Biçer, A. Lanzirotti, M. G. Newville, and F. D. Carlo, "Hyperspectral image reconstruction for x-ray fluorescence tomography," *Optics Express*, vol. 23, pp. 9014–9023, Apr 2015.

[42] D. Gürsoy, T. Biçer, J. D. Almer, R. Kettimuthu, S. R. Stock, and F. De Carlo, "Maximum a posteriori estimation of crystallographic phases in x-ray diffraction tomography," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2043, p. 20140392, 2015.

[43] S. Aslan, V. Nikitin, D. J. Ching, T. Bicer, S. Leyffer, and D. Gürsoy, "Joint ptycho-tomography reconstruction through alternating direction method of multipliers," *Optics express*, vol. 27, no. 6, pp. 9128–9143, 2019.

[44] M. Hidayetoğlu, T. Bicer, S. G. De Gonzalo, B. Ren, V. De Andrade, D. Gürsoy, R. Kettimuthu, I. T. Foster, and W.-m. W. Hwu, "Petascale xct: 3d image reconstruction with hierarchical communications on multi-gpu nodes," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ACM, in press.

[45] P. Chen, M. Wahib, S. Takizawa, R. Takano, and S. Matsuoka, "ifdk: a scalable framework for instant high-resolution image reconstruction," pp. 1–24, 11 2019.

[46] X. Wang, K. A. Mohan, S. J. Kisner, C. Bouman, and S. Midkiff, "Fast voxel line update for time-space image reconstruction," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1209–1213, IEEE, 2016.

[47] J. Blair, R. S. Canon, J. Deslippe, A. Essiari, A. Hexemer, A. A. MacDowell, D. Y. Parkinson, S. J. Patton, L. Ramakrishnan, N. Tamura, *et al.*, "High performance data management and analysis for tomography," in *Developments in X-Ray Tomography IX*, vol. 9212, p. 92121G, International Society for Optics and Photonics, 2014.

[48] G. Chantzialexiou, A. Luckow, and S. Jha, "Pilot-streaming: A stream processing framework for high-performance computing," in *2018 IEEE 14th International Conference on e-Science (e-Science)*, pp. 177–188, 2018.