

Live Iterative Ptychography

Dieter Weber^{1,*} , Simeon Ehrig^{2,3} , Andreas Schropp^{4,5} , Alexander Clausen¹ ,
Silvio Achilles⁴ , Nico Hoffmann², Michael Bussmann^{2,3} , Rafal E. Dunin-Borkowski¹ ,
and Christian G. Schroer^{4,6} 

¹Ernst Ruska-Centre for Microscopy and Spectroscopy with Electrons, Forschungszentrum Jülich, Jülich 52425, Germany

²Helmholtz-Zentrum Dresden-Rossendorf, Dresden D-01328, Germany

³Center for Advanced Systems Understanding, Untermarkt 20, Görlitz 02826, Germany

⁴Center for X-ray and Nano Science CXNS, Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, Hamburg 22607, Germany

⁵Helmholtz Imaging, Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, Hamburg 22607, Germany

⁶Department Physik, Universität Hamburg, Luruper Chaussee 149, 22761 Hamburg, Germany

*Corresponding author: Dieter Weber, E-mail: d.weber@fz-juelich.de

Abstract

We demonstrate live-updating ptychographic reconstruction with the extended ptychographical iterative engine, an iterative ptychography method, during ongoing data acquisition. The reconstruction starts with a small subset of the total data, and as the acquisition proceeds the data used for reconstruction are extended. This creates a live-updating view of object and illumination that allows monitoring the ongoing experiment and adjusting parameters with quick turn around. This is particularly advantageous for long-running acquisitions. We show that such a gradual reconstruction yields interpretable results already with a small subset of the data. We show simulated live processing with various scan patterns, parallelized reconstruction, and real-world live processing at the hard X-ray ptychographic nanoanalytical microscope PtyNAMi at the PETRA III beamline.

Key words: ptychography, X-ray microscopy, live processing

Introduction

Ptychography (Hoppe, 1969a, 1969b; Hoppe & Strube, 1969) can reconstruct a quantitative object transfer function for a specimen from a set of measurements where a spatially modulated illumination is shifted relative to the object and the intensity distribution of the transmitted and projected beam, typically far-field diffraction patterns, are measured for the different shift positions (Fig. 1). The alternating projection method, i.e. ptychographical iterative engine/extended ptychographical iterative engine (ePIE) reconstruction algorithm (Rodenburg & Faulkner, 2004; Maiden & Rodenburg, 2009) is one of the classical and well-known approaches to perform this reconstruction.

This method is currently in use at the hard X-ray ptychographic nanoanalytical microscope PtyNAMi at the PETRA III beamline P06 at DESY (Schropp et al., 2020). It allows to retrieve images of objects at a higher spatial resolution than the size of the X-ray focus and is physically only limited by the collection angle, wave length of the probe, and the available coherent photon flux (Schropp & Schroer, 2010).

Here, a ptychographic dataset is used for live processing that is recorded in the standard data format available the beamline. The raw data including the diffractions patterns and position values are stored in individual HDF5 files following in general the NeXus format.

Currently, the data are exported from the standard data format available at the beamline, and then processed offline using stand-alone software. The raw data, including the diffractions

patterns and position values, are stored in individual HDF5 files following, in general, the NeXus format (Könnecke et al., 2015). Weber et al. (2023) contains the dataset “scan_00063” with such a raw beamline data structure. Adjusting the parameters for ptychography requires a range of test measurements to adjust exposure, beam position on the detector, specimen region, focal distance and settings for the ptychography engine. Since many specimens exhibit mostly phase contrast, even simple previews of the data require a phase contrast method.

Live reconstruction shortens the time until feedback is available to the operators and can help to use the scarce beam time efficiently by readjusting or aborting measurements before they complete. Measurement times at PtyNAMi range between minutes for test measurements and many hours for large acquisitions.

Previously, Strauch et al. (2021) showed live single sideband (SSB) ptychography. However, this algorithm is not suitable for this particular application since the reconstruction resolution is limited to the scan grid, and it can only reconstruct successfully if the scan step size is smaller than the resolving power of the illuminating beam, determined by its convergence angle. The low acquisition speed and resolving power of X-ray microscopes compared to electron microscopes means reconstruction with higher resolution than the scan grid, which allows coarse sampling of the specimen, is highly favorable here. Live Wigner distribution deconvolution (WDD) (Bangun et al., 2022) has similar limitations.

Near live ptychographic reconstruction has also been shown by Pelz et al. (2021) by speeding up the reconstruction

Received: August 11, 2023. Revised: December 15, 2023. Accepted: January 9, 2024

© The Author(s) 2024. Published by Oxford University Press on behalf of the Microscopy Society of America.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

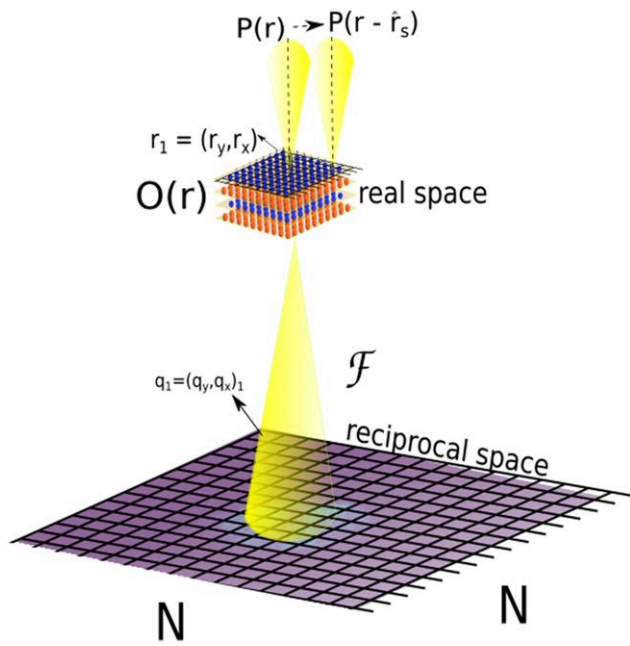


Fig. 1. Schematic setup for far field ptychography. The illumination is localized and scanned over the object $O(r)$. The detector captures the transmitted beam in the far field for each scan position. Illustration based on [Bangun et al. \(2022\)](#).

after fast scans so that it appears nearly instantaneous. However, this is not applicable for this use case since the acquisition speed and not the reconstruction is the limiting factor, making a partial reconstruction during an ongoing scan desirable.

Using the well-established ePIE implementation at DESY is favorable since it is in practical use for offline processing and the users' know-how is transferable. Towards this goal, three steps were undertaken:

1. Create an API for the established stand-alone ePIE implementation that allows to decouple data source, ptychographic processing and display of results.
2. Investigate the behavior of iterative phase retrieval on partial input data in a simulated setting to determine if live iterative ptychography is possible for typical datasets acquired at PtyNAMI.
3. Implement live ptychographic reconstruction at PtyNAMI.

Materials and Methods

A reconstruction with an iterative method encompasses the following aspects:

- Determine general parameters such as object and illumination size in pixel units, relation between physical coordinates and pixel coordinates.
- Preprocessing in order to match the different coordinate systems of object, illumination, scan and detector so that the forward and backward propagation model used in the optimization routine matches the actual measurements quantitatively.
- Assign initial values to object and illumination.
- Preprocess input data.
- Iterative optimization of illumination and object.

The following subsections detail how these aspects are adapted for live reconstruction.

ePIE Implementation

For the reconstruction of the experimental data, a GPU accelerated implementation of the ePIE ([Maiden & Rodenburg, 2009](#)) algorithm is used. The update strength for object and illumination, typically denoted by α and β , was 1 for all reconstructions shown here. The software is written in CUDA C++ and developed by the X-ray Nanoscience and X-ray Optics group at DESY for PtyNAMI. ePIE is an iterative algorithm, which reconstructs the object and the illumination at the same time. The implementation starts with a general initialization phase which is then followed by the main reconstruction routine. The initialization phase prepares the diffraction patterns for the reconstruction and sets up the object and illumination with an initial guess. In this implementation the diffraction patterns are cropped to a size of 2^n , with n being an integral value, in order to use a simple fast-Fourier transformation and increase the performance. Furthermore, the implementation can be configured to transform the diffraction patterns. Since the forward and back projection in this ePIE engine is implemented with a Discrete Fourier Transform (DFT), the sampling and coordinate system of the reconstructed object transfer function must match the sampling and coordinate system of the diffraction patterns used in the reconstruction to allow forward and backward projection with a DFT. In the ePIE implementation used at PtyNAMI, the spatial resolution and size of the object reconstruction in real space is determined by the size and scale of the diffraction patterns and the scan area. That means the spatial resolution and size of the object reconstruction can be adjusted by cropping the diffraction patterns before reconstruction. Furthermore, depending on the detector and its orientation, the recorded raw diffraction patterns might be rotated and/or flipped relative to the coordinate system of the position encoders. This can be compensated by rotating or flipping the diffraction patterns in such a way that their coordinate system and handedness match the position encoders. In the second phase, the object and illumination are reconstructed. The implementation iterates over the diffraction patterns in a randomly shuffled order and calculates an update for the object and illumination. The reconstruction ends after a specific number of iterations.

Extensions of the ePIE Implementation

The original ePIE application is designed to perform an offline reconstruction, which means all diffraction patterns are available at the beginning of the reconstruction and the reconstruction configuration remains constant during runtime. There is no user input to add new diffraction patterns, and changes of the reconstruction parameters during the reconstruction are not possible. However, a live reconstruction requires that diffraction patterns can be added to follow the progressing scan during the reconstruction. Furthermore, it is necessary to stop and continue the reconstruction in order to extract intermediate updates, include new data and match the iteration speed with the acquisition speed to not over-optimize prematurely on a small subset of the input data.

For the interactive live reconstruction, Python bindings have been added to the existing C++ reconstruction engine using Pybind11 in order to connect it with the Python-based data

handling and control logic. Compared to C++, Python provides a read-eval-print loop (REPL) interpreter, which is perfectly suited for interactive workflows. Furthermore, Jupyter Notebooks can be used, giving some extra benefits such as an easy programming interface in the web browser and the possibility to combine source code with different media types, like markdown text and images to better document the reconstruction.

Preprocessing

Real-world detector data may require preprocessing such as cropping, scaling and orientation changes. Since the bindings do not support exporting or extending the loaded data, but require creating a new instance of the engine whenever the input data changes, a LiberTEM user-defined function (UDF) was implemented to perform preprocessing equivalent to the ePIE implementation based on its configuration file. This allows reusing the preprocessed data and reduces the amount of data loaded into the ePIE implementation. Furthermore, LiberTEM can perform this efficiently on live data streams in a parallel fashion.

Determine Parameters

For offline processing of an entire dataset, the ePIE engine determines the reconstruction parameters from the size of the detector data in pixels after preprocessing, bounding box of physical scan positions, and optical parameters such as camera length, detector pixel size and wavelength from the configuration file. For live processing, the ePIE engine is initialized with mock detector data and approximate scan positions based on the scan settings that are known beforehand. The engine uses the nominal scan positions to calculate the nominal object size and pads it with half the illumination size so that the full area of the object that intersects with the illumination is covered. During the experiment, the scan positions are measured by interferometers or encoders.

Based on this initialization, a configuration for the engine is derived from the initial configuration that fixes the size and relation between physical coordinates and pixel coordinates. That way the input data can change dynamically during live processing without influencing the reconstruction parameters.

Simulating Live and Parallel Processing

Live processing starts with a small initial subset of data that is extended gradually as the scan proceeds (Fig. 2). As a preliminary test before implementing true live processing, this was simulated on offline data using the bindings described above: The initial values for object and illumination together with a small subset of the input data are loaded, a small number of iterations is performed, the resulting object and illumination are recorded, and the cycle is repeated with a larger subset of the input data using the previous state of object and illumination as initialization values until the complete dataset is consumed. Different scan schemes such as random sampling or spiral scans can be simulated by reordering the input data and positions.

Parallel processing can be simulated in a similar fashion by independent iteration on disjoint subsets of the input data with intermittent averaging between the object and illumination for synchronization, similar to Nashed et al. (2014). This creates a

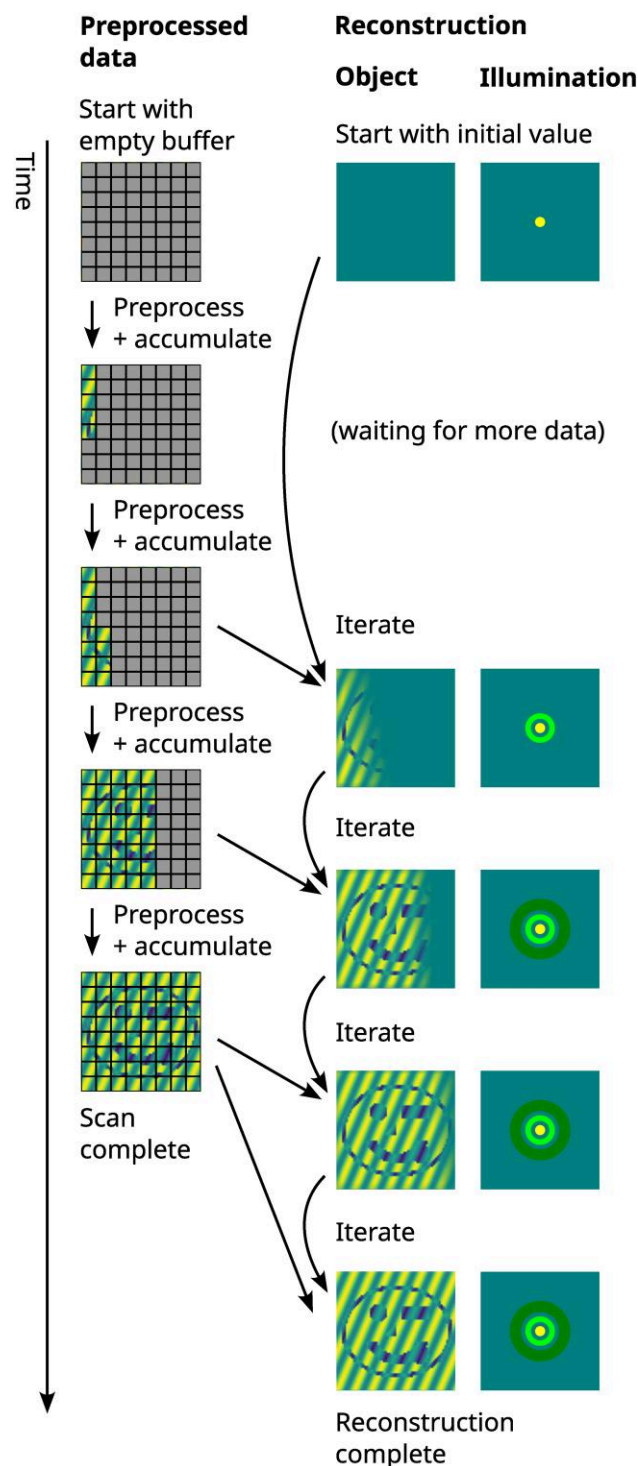


Fig. 2. Schematic progress of live iterative ptychography. Data are preprocessed and accumulated, and the reconstruction algorithm includes more and more data in the iteration as it becomes available. This creates an evolving view of the object and illumination reconstruction. In real-world live processing, the sequence is determined by the acquisition speed, iteration speed and possibly limits on the number of iterations to avoid over-optimization. For simulated live processing, the sequence of available data and iteration steps is predetermined by a script.

hybrid between classical ePIE where the object and illumination are updated dynamically with each diffraction pattern during a pass, and batch-oriented algorithms that accumulate updates and apply them only after a pass.

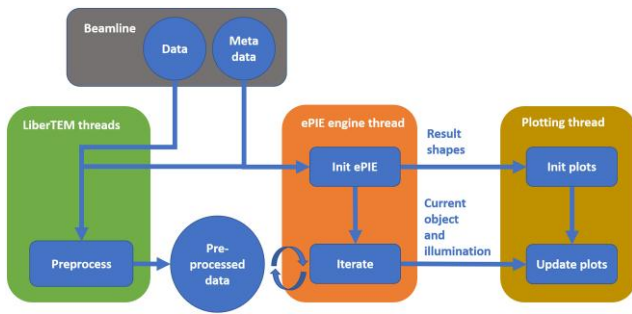


Fig. 3. Diagram of the data flow for live ePIE.

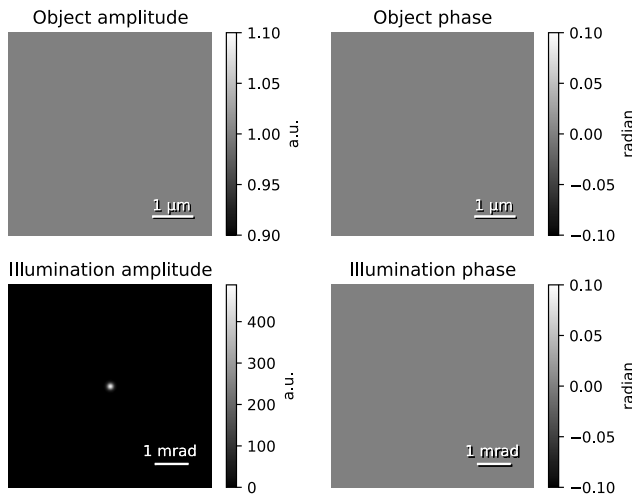


Fig. 4. Starting value for all simulated live reconstructions. The object is initialized with the identity function. The illumination amplitude is initialized with a small Gaussian peak and flat phase of 0.

A ptychographic dataset for offline reconstruction had been recorded at PtyNAMi (Schropp et al., 2020) on an NTT-AT resolution test chart with 50 nm thinnest lines and spaces (model ATN/XRESO-50HC) at an X-ray energy of $E = 13.5$ keV. The X-ray beam was focused by a pair of nanofocusing refractive lenses (NFLs) (Schroer et al., 2005; Schropp et al., 2010) to a spot size of 116 nm (horizontal) \times 118 nm (vertical) full width at half maximum. The sample was positioned slightly in defocus at a distance of approximately 200 μm . The diffraction patterns were collected with an Eiger X 4M detector (DECTRIS AG) positioned at a distance of 3.2 m behind the sample. In the reconstructions these patterns were cropped to a size of (256, 256) pixels, which yielded in this case an effective real space pixel size of 15.3 nm. The dataset was acquired in step-scan mode with a nominal scan window of $1\ \mu\text{m} \times 1\ \mu\text{m}$ with a step size of 50 nm. This gives 21 scan positions in vertical and horizontal scan direction, so that the resulting dataset consisted of 441 diffraction patterns in total, each with an exposure time of 0.5 s. The approximate scan positions were obtained from the encoders of the piezo scanner and were used as a direct input for ptychographic phase retrieval. The position errors of the setup were not considered here. In practice, the beam, specimen, and encoders can shift relative to each other.

The following processing schemes were implemented using this test dataset:

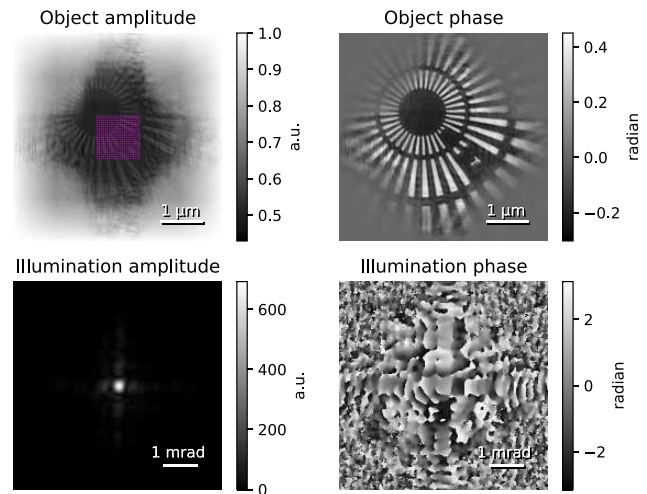


Fig. 5. Reference result with 1,000 iterations on the full dataset. The dots in the upper left plot indicate the scan positions.

- 1,000 iterations on the full dataset for reference (Fig. 5).
- Following the original line scan pattern in y direction, add 10 patterns and perform 10 iterations per step. After adding the complete data, iterate on the full dataset until 1,000 iterations in total are reached (Fig. 6).
- Add a random pick of 10 patterns and perform 10 iterations per step. Iterate on the full dataset until 1,000 iterations in total are reached (Fig. 7).
- Add 10 patterns following an inwards square spirangle pattern and perform 10 iterations per step. Iterate on the full dataset until 1,000 iterations in total are reached (Fig. 8).
- Split the dataset into four random disjoint subsets. Perform 10 iterations on each subset independently, synchronize and repeat until 1,000 iterations are reached. This simulates parallel processing on offline data (Fig. 9).
- Add 10 patterns following the original line scan in y direction to four disjoint subsets in a round-robin (Kleinrock, 1964) fashion, perform 10 iterations on each subset after adding patterns to a subset, and synchronize. Iterate until 1,000 total iterations are reached. This simulates parallel processing on live data (Fig. 10).
- Following the original line scan pattern in y direction, add 10 patterns and perform 1,000 iterations per step to investigate impact of early over-optimization (Fig. 13).

See “Data and code availability” section for the test data, ePIE implementation, and test code.

Live ePIE

In the simulated live processing described above, the processing follows a predefined sequence of data loading, iteration and visualization that is executed sequentially in a deterministic fashion. All necessary parameters are read from configuration files.

After feasibility of live processing was demonstrated, these processes were implemented for real-world live processing at the DESY beamline in a Jupyter notebook as concurrent threads that follow the speed of the data source and iterate as fast as possible (Fig. 3). It was executed on a node of the Maxwell cluster at DESY which has access to a cluster file system where results from the beamline are written in chunks

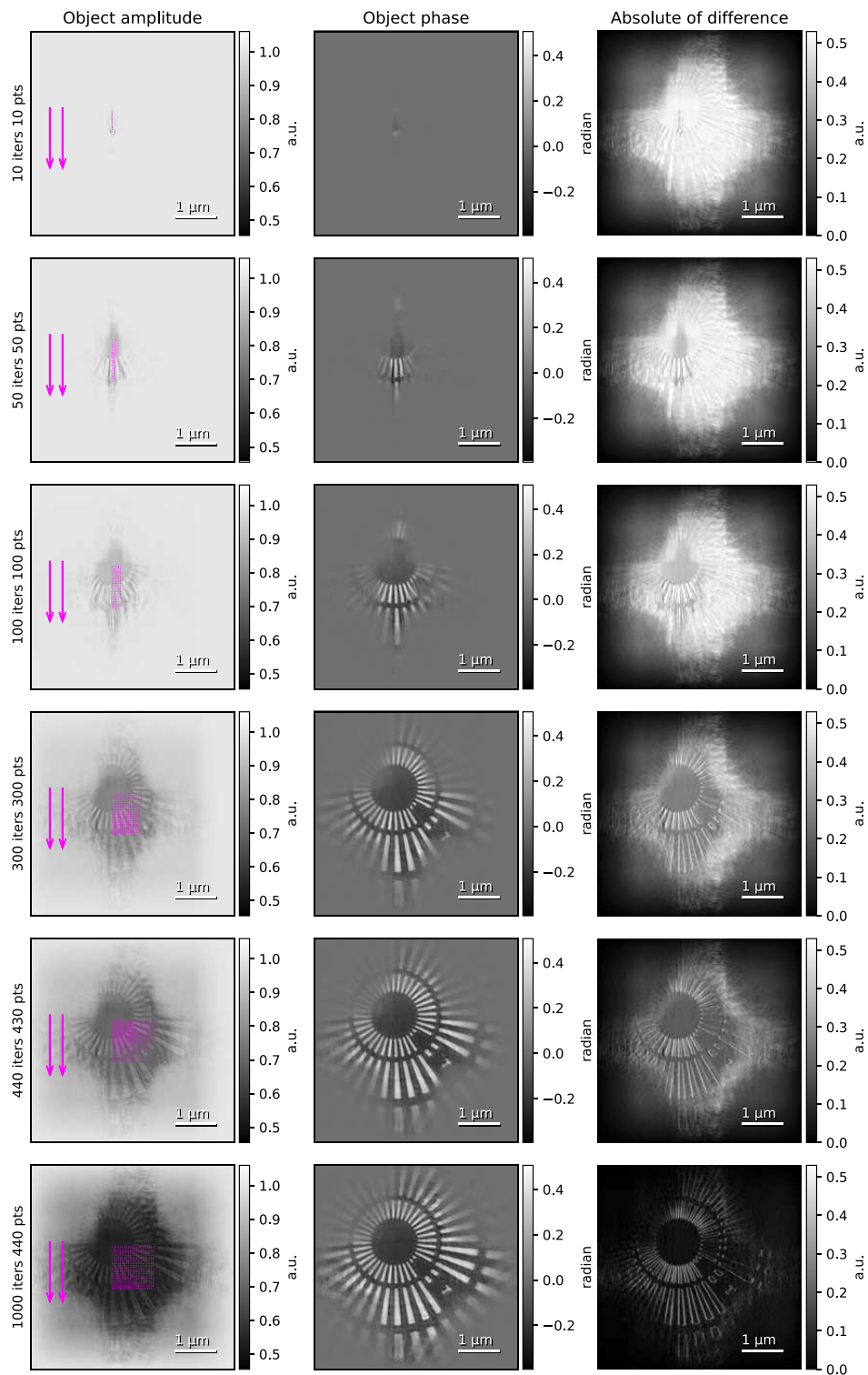


Fig. 6. Simulated live reconstruction following the original line scan pattern in y direction. The dots in the left column indicate the scan positions. The right column shows the absolute value of the difference to the reference result.

with a short time delay, as well as the beamline's event interface. In principle, the data could also be streamed over network instead of using chunks in a file system, but such an interface is currently not implemented at the beamline. The pipeline is armed prior to data acquisition and starts processing as soon as a scan_started event is received. Alternatively, the processing can attach to a running scan or load data

from previous scans. The expected scan area is extracted from the scan_started event, while other reconstruction parameters are determined from a configuration file for the ePIE engine.

A custom LiberTEM acquisition object was developed that reads the detector data from the folder structure that the beamline creates. This acquisition object is used as a data

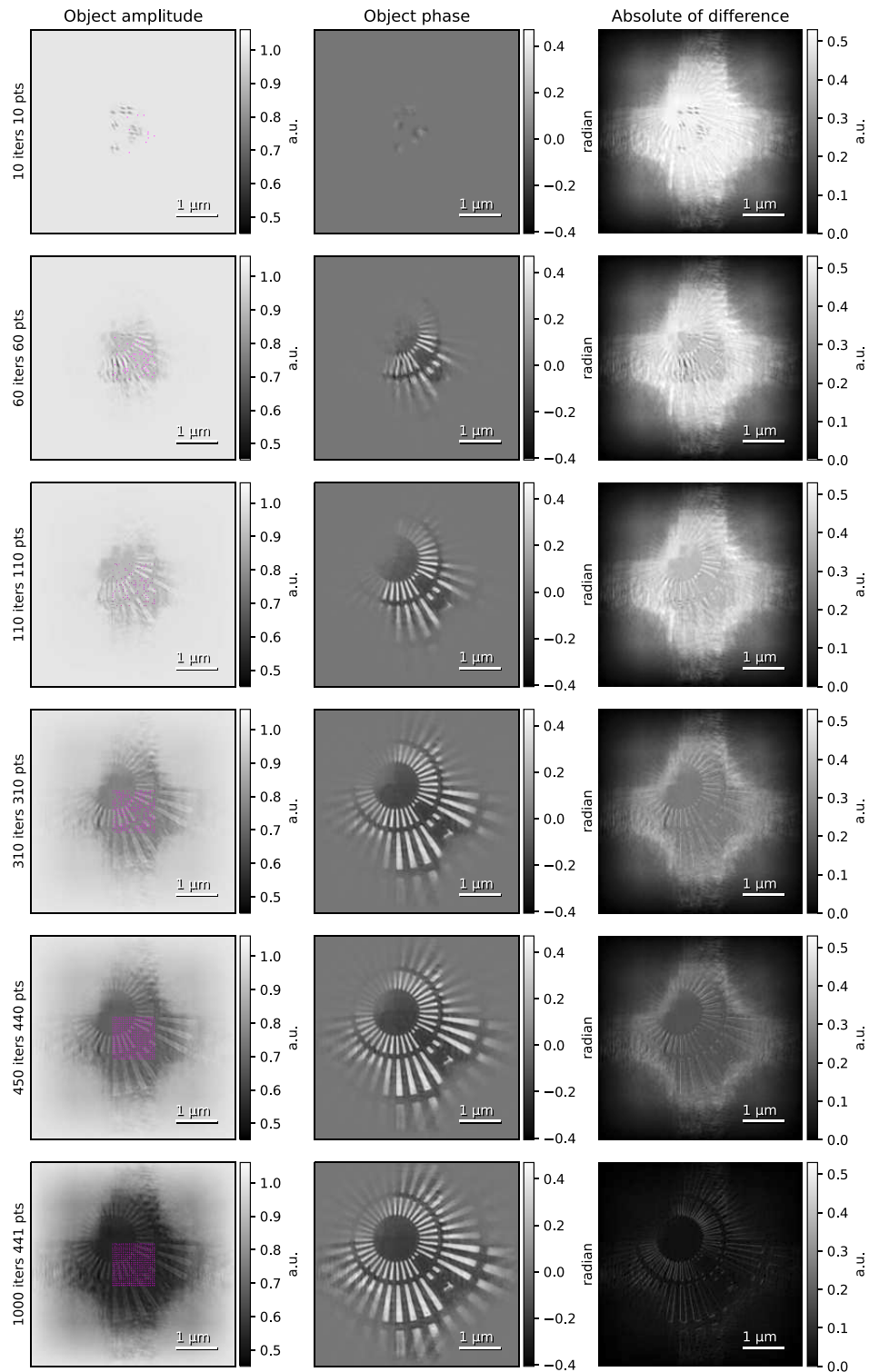


Fig. 7. Simulated live reconstruction with a random scan pattern. The dots in the left column indicate the scan positions. The right column shows the absolute value of the difference to the reference result.

source for the preprocessing UDF. In addition, another UDF reads the matching stage encoder values from the beamline data to determine the position of each scan point. LiberTEM runs these UDFs in a pool of worker processes in the background. The raw beamline data are available in (Weber et al., 2023).

The `run_udf_iter` interface of LiberTEM provides a Python generator with updated partial preprocessing result from the

UDF for each received input data partition, together with a map that describes which parts of the expected data have been received so far. A feeder thread takes these updates from the generator and feeds them into the processing thread.

The processing thread consumes these updates and waits until a sufficient subset of data is received. Then, it loads the data into the ePIE implementation and starts iterating. Between iterations, it checks if new data are available and

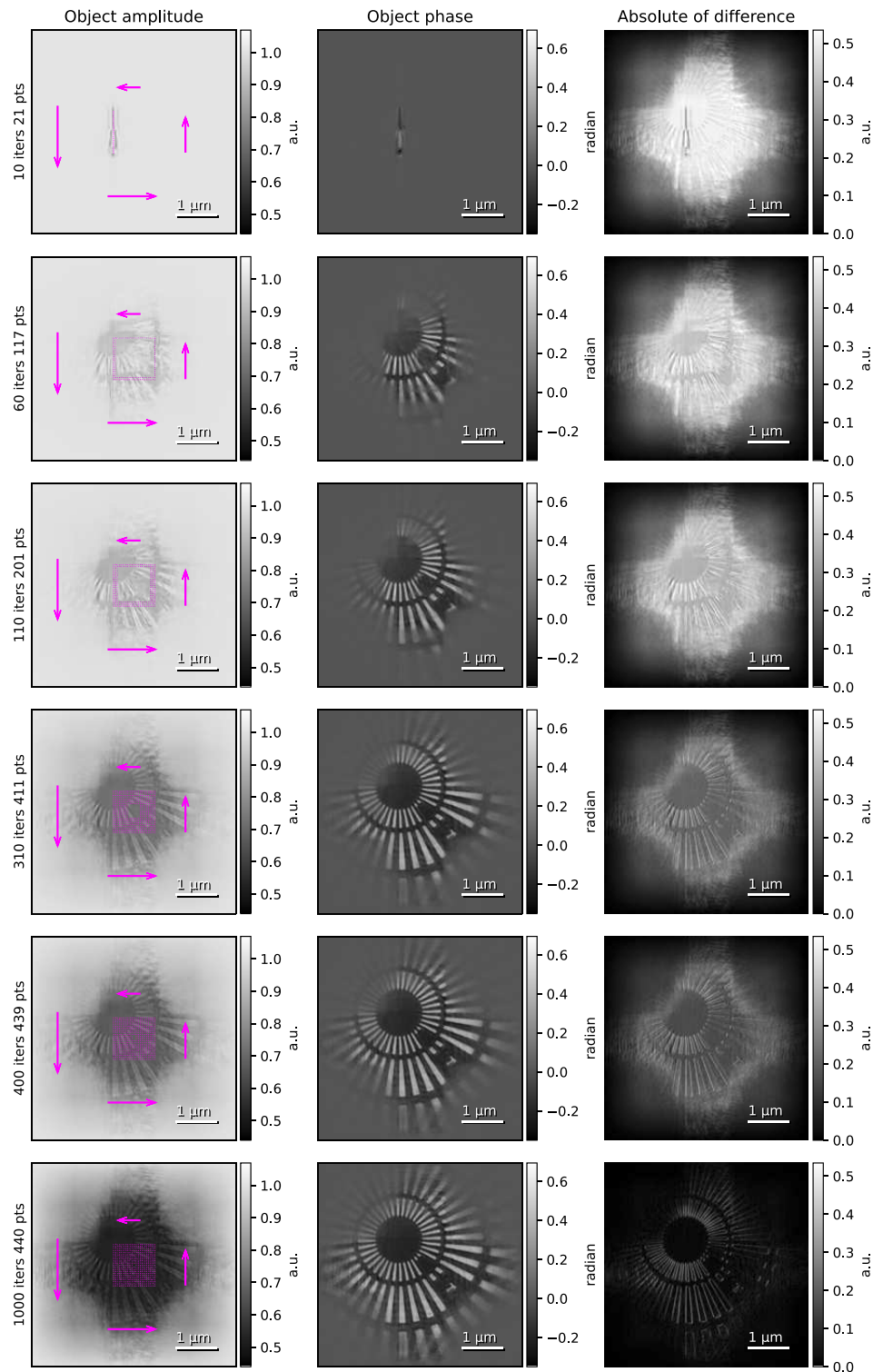


Fig. 8. Simulated live reconstruction following a spirangle scan pattern. The dots in the left column indicate the scan positions. The right column shows the absolute value of the difference to the reference result.

loads it as required, and regularly feeds an intermediate state of the reconstruction into a queue for display.

The visualization thread reads from this display queue and updates LiberTEM live plots with each new result.

Live processing was performed while recording data on an NTT-AT resolution test chart with 50 nm thinnest lines and spaces (model ATN/XRESO-50HC) at an X-ray energy of

$E = 9$ keV. The X-ray beam was focused by a Fresnel zone plate (FZP) with outer diameter of $125 \mu\text{m}$ and thinnest outermost lines of 70 nm. At the employed X-ray energy, these optics create a focus with a size of about 85 nm at a focal distance of 64 mm. The diffraction patterns were collected with an Eiger X 4M detector (Dectris AG) positioned at a distance of 3.265 m behind the sample. In the reconstructions, these

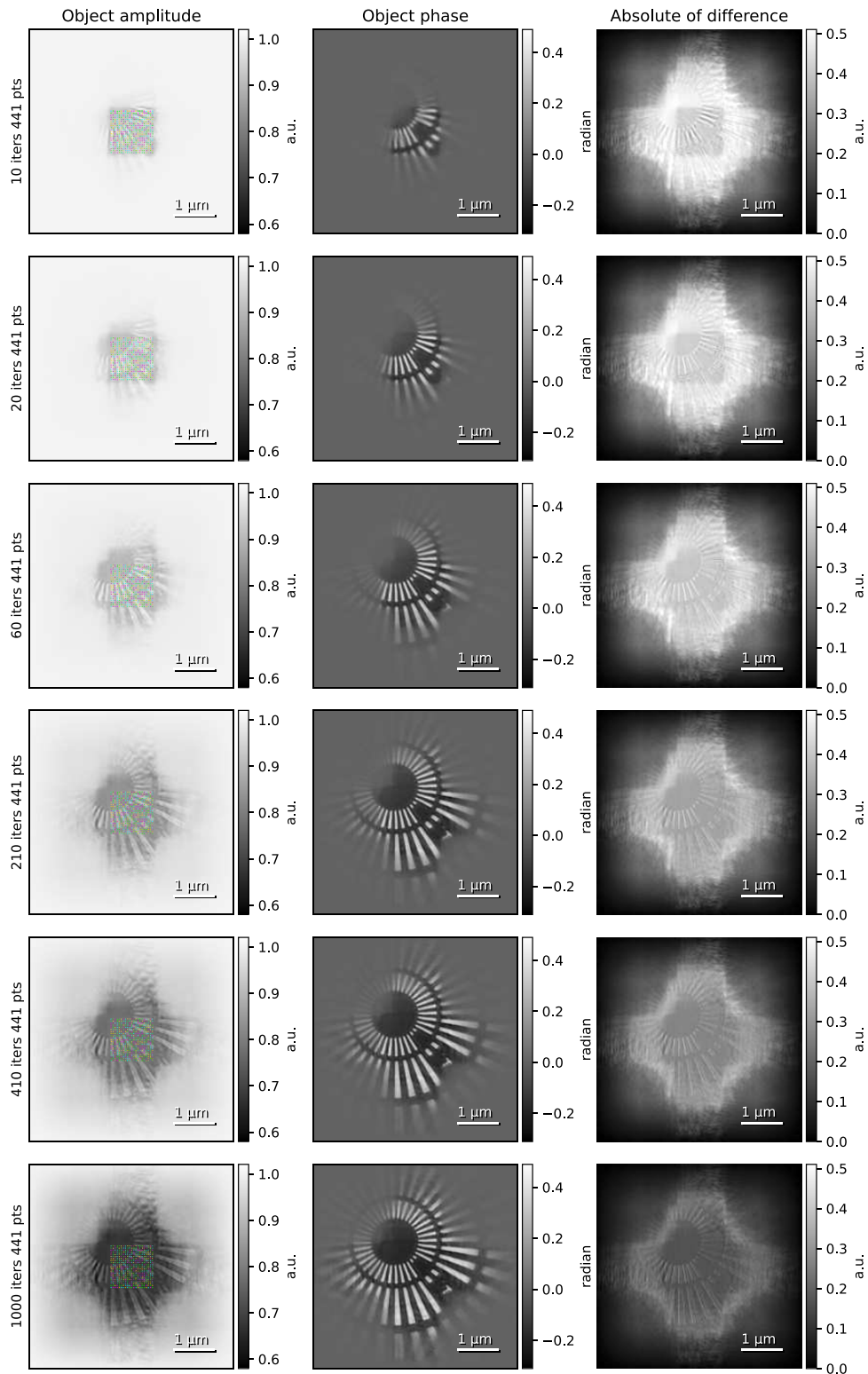


Fig. 9. Simulated parallelized reconstruction on four random disjoint subsets. The dots in the left column indicate the scan positions, with the dot style indicating the subset. The right column shows the absolute value of the difference to the reference result.

patterns were cropped to a size of (256, 256) pixels. The dataset was acquired using a continuous meander scan with a nominal scan window of $10\ \mu\text{m} \times 11\ \mu\text{m}$ and a step size of 100 nm. The resulting dataset consisted of 11,245 diffraction patterns in total, each with an exposure time of 20 ms.

The test object was mounted in a different orientation between the measurement campaign for the simulated live

processing and the actual live processing. Both the recording for simulated live processing and for real live processing were not adapted specifically for this purpose, but performed alongside regular beamtimes at PtyNAMI during the preparation phase for real measurements. The parameters are different and in the typical range for PtyNAMI for that reason.

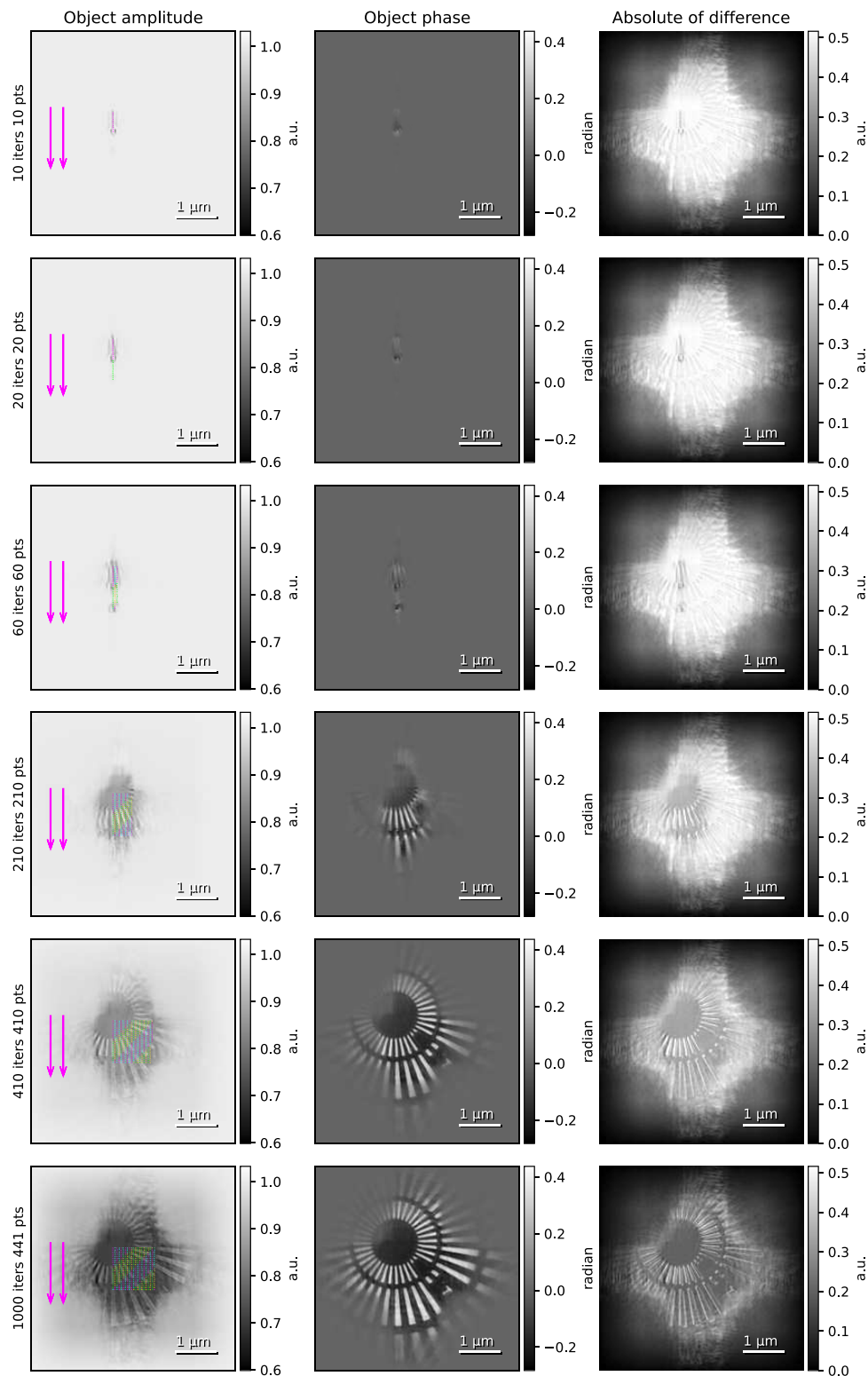


Fig. 10. Simulated parallelized live reconstruction on four gradually growing disjoint subsets. The dots in the left column indicate the scan positions, with the dot style indicating the subset. The right column shows the absolute value of the difference to the reference result.

Results

Simulated Live and Parallel Processing

Figure 4 shows the starting value of illumination and object used in all simulated live processing.

Figure 5 shows amplitude and phase of the reconstruction of both illumination and object using 1,000 iterations on the

complete dataset, corresponding to conventional offline processing.

Figures 6 to 10 show how the simulated live and parallel processing results evolve with different processing and subdivision schemes. Note how the reconstruction extends far beyond the scan range due to the extended size of the illumination.

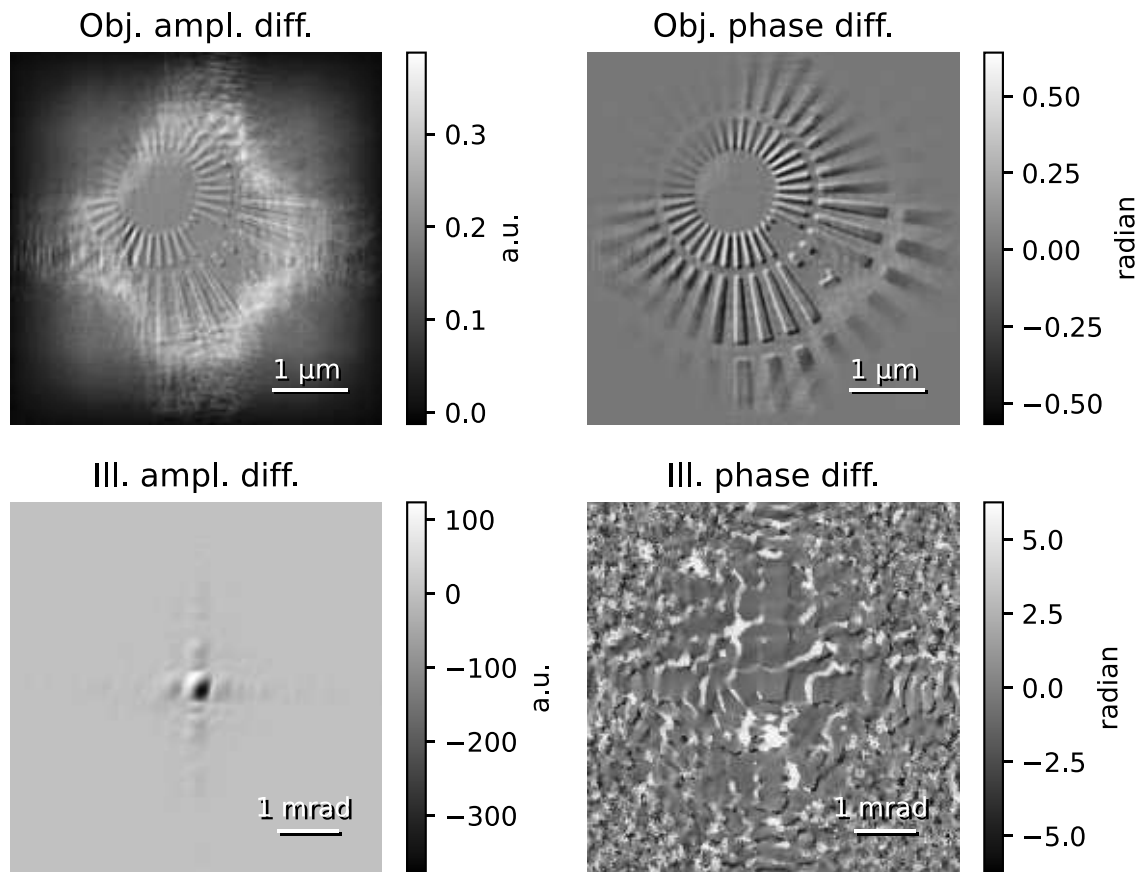


Fig. 11. Amplitude and phase difference between the final result of Figure 10 and the reference result (Fig. 5). The amplitude of the object is higher, while the amplitude of the illumination is lower. The contour line around the sharp features of the object phase indicate a small shift relative to the reference result.

In all cases, a reconstruction with strong resemblance to the final result emerges already after a few iterations on a small subset of the dataset. The random (Fig. 7) and spirangle (Fig. 8) scan patterns cover the entire scan area early on, demonstrating how such acquisition schemes can be advantageous for live processing compared to a sequential line scan to get an overview of the object early on.

After completing the same number of iterations, the reconstructions that are not parallelized differ from the reference result mainly by a small shift that manifests itself as a contour line around sharp features of the object in the difference plot. The parallelized results exhibit a stronger difference that did not vanish up to 1,000 iterations. The plot of amplitude and phase difference in Figure 11 reveals that the amplitude of the object is larger and the amplitude of the illumination is lower than the reference. However, they still show a qualitative visual resemblance to the reference result.

Animated videos of the simulated live reconstruction are available in Weber et al. (2023).

Live Processing

Figure 12 shows how the live view of object and illumination reconstruction develops during live processing. This allows monitoring the scan. Since the reconstruction can be restarted and reattached to a running scan, users can already start optimizing reconstruction parameters while a scan is running. See “Data and code availability” section regarding availability of source code, raw data and animated results.

A screen capture video of the live reconstruction is available in Weber et al. (2023) supplementary material.

Discussion

The results on simulated and real-world live processing show that iterative ptychography algorithms can be used for live reconstruction by starting the iteration on a small subset of the data that grows as the data acquisition proceeds. The state of object and illumination are carried forward. For this application, a ptychography engine should offer the following features:

- An API that allows integration in a complex workflow, as opposed to stand-alone programs.
- Flexible and fast import and export of object, illumination and data via the API.
- Stepping the iteration to extract intermediate values or stop as required.
- Options to encourage stable convergence, such as good starting values for illumination and object, restrictions on value ranges and illumination shapes, countering impact of noise and/or detector gain characteristics, masking or cropping invalid pixels, etc.
- Configuration of a stable, predetermined size and shape of the object and illumination as well as a relation between physical scan position and pixel position in the reconstructed object based on the expected acquisition parameters.
- Separation of preprocessing from the iteration engine.

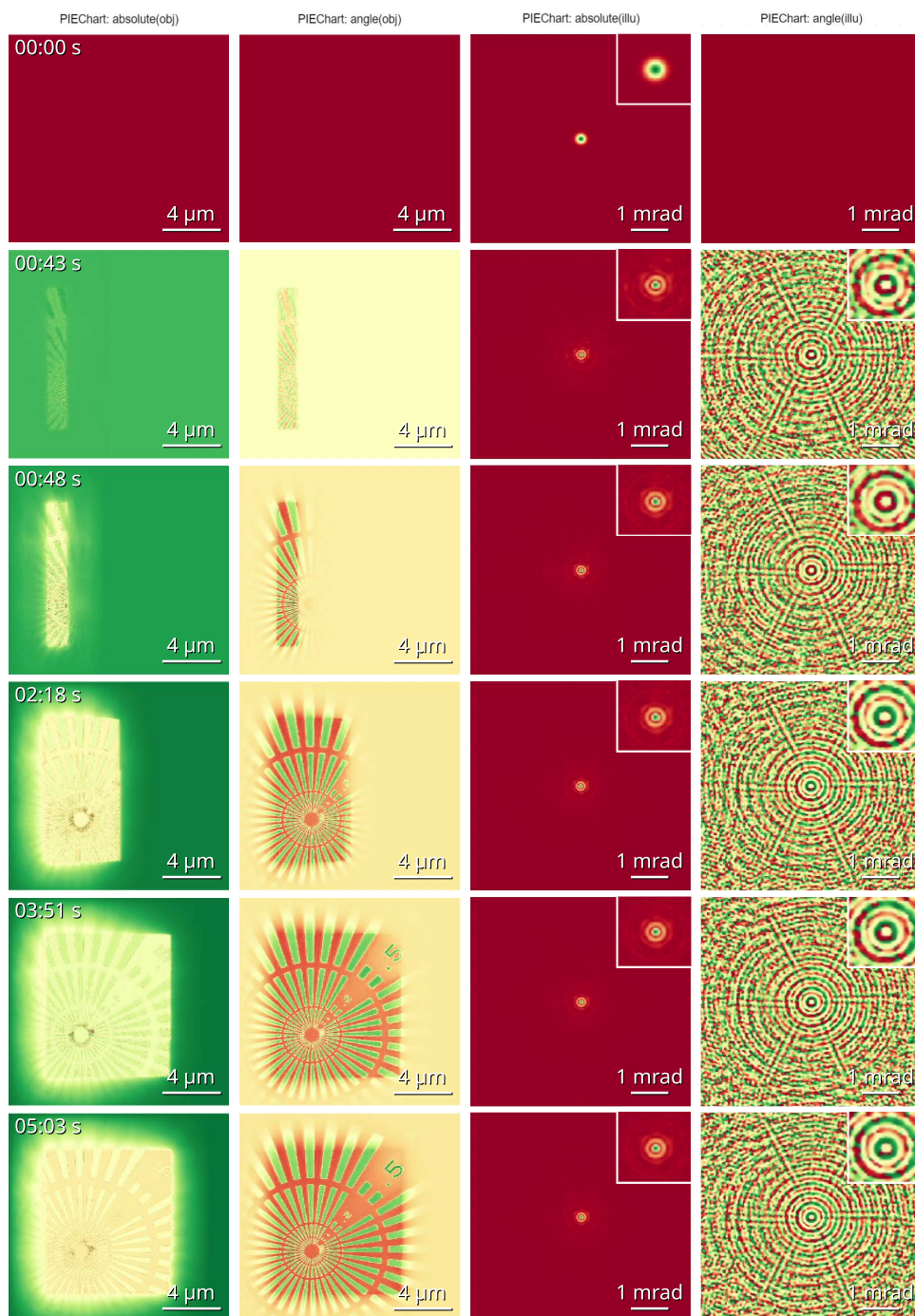


Fig. 12. Evolution of amplitude and phase for object and illumination that was recorded during live processing. The screen with the Jupyter notebook running the reconstruction was recorded as a video. This figure shows rearranged and annotated snapshots of the plots taken at the indicated times relative to the start of the scan. The full video is available in [Weber et al 2023](#) in [Supplementary Material](#). The insets show the center of the illumination scaled by a factor of 2. The first step shows the initialization before iteration starts. The second step was recorded just after the iterative reconstruction started. Steps 3 and 4 show arbitrary intermediate states. The second to last step shows the state at the moment the scan completes. The last snapshot was taken after a number of additional iterations. In the beginning, 50 iterations were executed for each batch of 500 images received from the beamline. Towards the end, about six iterations were executed per batch due to the much larger number of scan points included in each iteration. The live plots were updated every two iterations for quick visual feedback.

In particular when the beamline at DESY acquires data at a low rate, around 10 frames per second, it is very advantageous to obtain a view of the object early on during an acquisition to abort and adjust parameters in case issues are detected, such as the choice of acquisition area, exposure, or focal plane. During real-world live processing, the implementation could easily keep up with this data rate. During the experiment shown in [Figure 12](#),

the region on the test specimen was already clearly recognizable after completion of about a quarter of the scan. An operator has a chance to detect issues with the field of view or settings at this point. Since large scans at PtyNAMI can run for many hours overnight, detecting issues at 25% completion or earlier can save significant amounts of beam time and allows early optimization of reconstruction parameters.

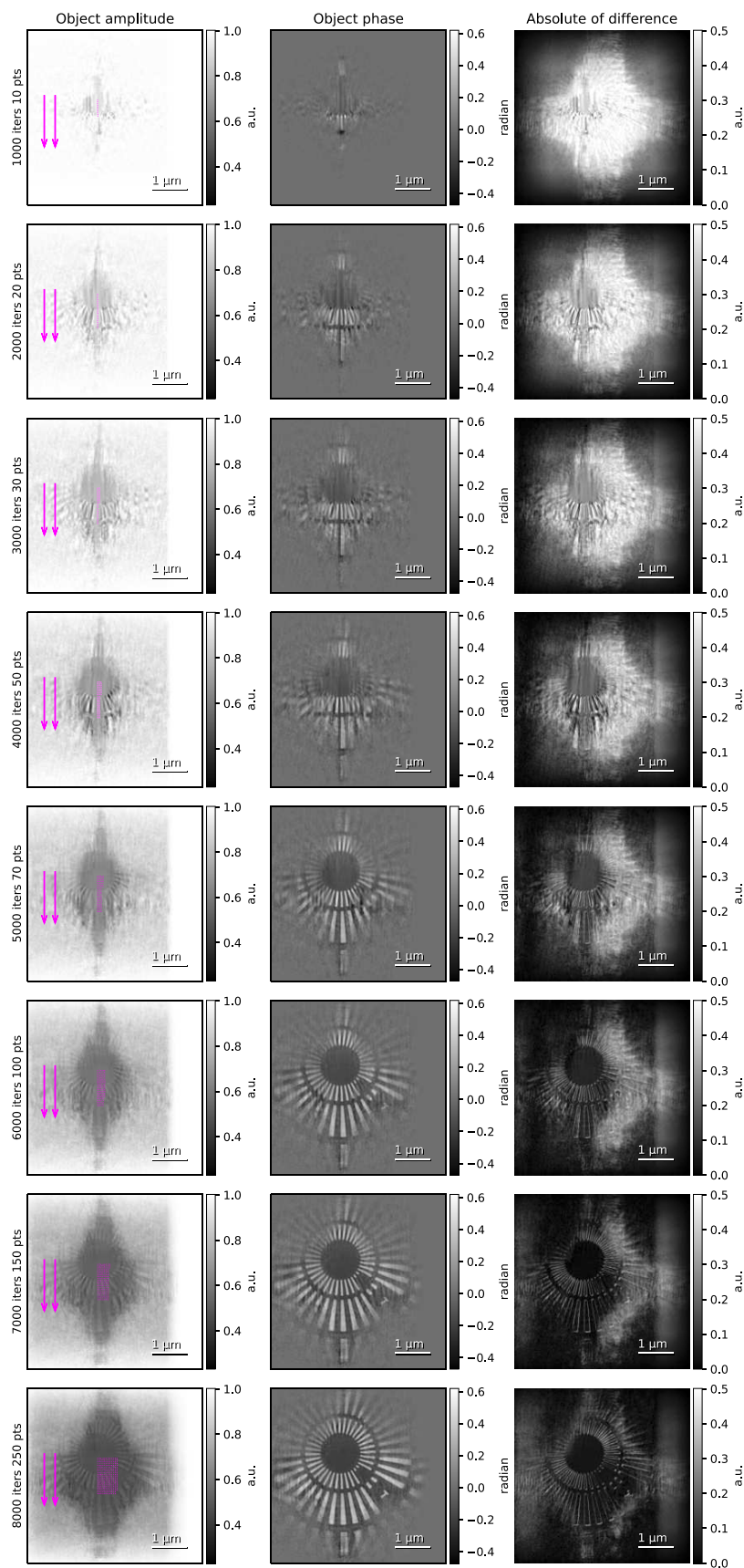


Fig. 13. Simulated live reconstruction following the original line scan pattern in y direction with over-optimization on a small subset of input data. The dots in the left column indicate the scan positions. The right column shows the absolute value of the difference to the reference result.

Figure 13 shows that the reconstruction does not diverge even with a high number of iterations on a small initial subset for this particular dataset, engine and choice of parameters. As soon as a region of the test specimen is covered by a sufficient number of scan points, the iteration stabilizes on a value similar to the reference result in this region. Similar to the other scans, this reconstruction exhibits a small shift that results in a bright line around sharp features in the difference plot. A result that clearly shows features of the test object was obtained with three and a half scan rows in this test. This indicates that a successful reconstruction requires some degree of lateral coverage with overlapping illuminations, but starting the iteration early with a smaller number of diffraction patterns is not harmful in this case. The lower limit for successful partial reconstruction will likely be different for different illumination shapes and sizes, scan patterns, objects, and reconstruction engine settings. Typically, engines for iterative ptychographic reconstruction will have many parameters and constraints for regularization in order to ensure a stable convergence with real-world data. Here, we merely show that the results for live processing for all tests that we performed eventually converged towards the offline result as more and more data were added.

As long as the convergence is stable, meaning it converges quickly towards the same result independent of the starting values for illumination and object, and stabilizes after a sufficient number of iterations, live reconstruction will eventually converge on the offline result under all circumstances: The previous reconstruction steps with a subset of the input data merely generate a starting value for illumination and object for the final iteration phase on the complete input data. The final iteration phase will then converge and stabilize on the offline result.

Furthermore, any real-world dataset for ptychographic reconstruction can be understood as a partial dataset: The actual scan size and the size of the object are only limited by practical concerns and can be, in principle, arbitrarily large. That means any real-world dataset can be extended by just continuing the scan beyond the initial borders, making the initial “full” dataset a partial dataset. The reconstruction on a subset of a large dataset is not distinguishable from the reconstruction of a “full” dataset with the same scan coordinates as the subset. Regions of the object transfer function that are not touched by the illumination function for any scan position just stay at their initial values.

The results shown here conform to this mathematical expectation: Eventually, a reconstruction on a growing subset of a sufficiently large dataset will give a stable reconstruction for all regions of the object that are covered by the illumination function in that subset, provided the reconstruction of the complete dataset is stable. For a general discussion and further references on the convergence of ptychographic reconstruction see, for example, Melnyk (2023).

Simulated live reconstruction and actual live reconstruction mainly differ in their input and output method: In simulated live processing, a “recipe” with well-defined steps is executed, and all intermediate results are stored and then plotted. In actual live processing, the sequence of data and iteration is determined by the interplay of data source, preprocessing and iteration thread. The intermediate results are not recorded here, but just plotted for quick visual feedback. Notably, here live reconstruction is not meant to replace careful offline reconstruction with optimized parameters. It is rather a quick preview tool to observe what data are being recorded and to

check if the chosen parameters are likely to give a successful offline reconstruction later.

In this article, we show practical evidence that such live reconstruction can be possible and beneficial under the typical conditions at PtyNAMi. In particular, live reconstruction worked reliably for different illumination shapes and scan parameters that were chosen somewhat arbitrarily from typical values. The live reconstruction seems rather robust against variations in the set of diffraction patterns and number of iterations: By adding more and more diffraction patterns, the result reliably converged to closely resemble the offline reconstruction result in the region covered by the scan, meaning it seems to meet the conditions for successful partial reconstruction described in the previous paragraphs. It seems intuitive that a scan that yields a satisfactory reconstruction with only a subset of scan points is likely to also yield a satisfactory result for the full scan. That means it is valuable as a preview tool to give an early indication to operators if the scan and reconstruction parameters will give a good final result. Beamline operator feedback was positive.

The parallelized reconstruction schemes can potentially be used to generate faster previews by using multiple GPUs in parallel. An engine and processing scheme that produces the same result independent of the number of parallel GPUs is required, which was not yet achieved here. Generally, algorithms that act in a parallel manner, such as relaxed averaged alternating reflections (RAAR) or difference map, are advantageous for this (Enders & Thibault, 2016). The PtyPy software package (Enders & Thibault, 2016) already offers parallelized reconstruction and can potentially be used as a reconstruction engine for live processing in a similar fashion.

Outlook

Faster processing seems feasible, in particular targeting electron ptychography with fast direct detection detectors and fourth-generation synchrotron radiation sources such as PETRA IV, since the preprocessing with LiberTEM is efficient and highly scalable, and parallelized live processing on several GPUs seems feasible with a suitable reconstruction engine. Further optimization of the ePIE engine is thinkable as well, for example to run on different platforms such as field-programmable gate arrays (FPGAs) using Alpaka (Matthes et al., 2017).

In the future, live reconstruction can be factored in when choosing scan patterns. A spiral or spirangle seems promising since it combines short movements of the stage between subsequent acquisitions with early coverage of the whole field of view, and such patterns are already available at PtyNAMi. A conventional line scan pattern can be advantageous if the position encoders exhibit hysteresis. Further experiments to test the behavior of live reconstruction for different scan patterns, specimens, illuminations, and scan step sizes seem promising. Here, we demonstrate that live reconstruction can yield useful results for different scan patterns.

The approach for live iterative ptychography described here is not limited to ePIE but should work with any reconstruction engine that offers a similar interface. Implementations of different reconstruction approaches such as multislice ptychography, mixed state ptychography, near field ptychography (You et al., 2023), and/or alternative projection schemes such as RAAR, alternating direction method of multipliers (ADMM) and similar can meet the requirements for live

processing described in the previous section: They optimize gradually, and the set of scan positions is flexible. Furthermore, this approach should be transferable to other experimental modalities such as extreme ultraviolet light, visible light, and electrons. A trial with electron microscopy data is included in the [Supplementary Material](#).

Live monitoring is not limited to iterative ptychographic reconstruction: Any other processing or reconstruction method that is compatible with LiberTEM can be included with little effort in the preprocessing stage, such as Centre of Mass or even direct ptychographic reconstruction (Strauch et al., 2021; Bangun et al., 2022).

Furthermore, in the future it can be investigated under which conditions the reconstruction is quantitative and how large the errors are at various stages of the scan and reconstruction.

Panning and zooming with a live view may be possible by applying a suitable transformation to an intermediate state of the object in order to generate a new starting value, generating a new configuration and continuing the acquisition and iteration with the new parameters. This can allow live navigation on the specimen and adjustment of parameters at the beamline, effectively creating an interactive microscope for X-rays and other ptychography modalities. The results also show that nonstandard scan patterns can optimize early coverage of the field of view. Previously, Velazco et al. (2020) have shown nonstandard scan strategies, including a space-filling curve. Live reconstruction can also enable adaptive scanning strategies (Ede, 2021).

ePIE extended ptychographical iterative engine
 REPL read-eval-print loop
 UDF user-defined function
 SSB single side-band
 WDD Wigner distribution deconvolution
 ADMM alternating direction method of multipliers
 RAAR relaxed averaged alternating reflections
 DM difference map
 DFT Discrete Fourier Transform

Availability of Data and Materials

The ePIE reconstruction engine and interface for live processing is available upon reasonable request from Andreas Schropp andreas.schropp@desy.de.

Additional movies that show the live reconstruction, the dataset and notebook for simulated offline processing, the Jupyter notebook for live processing at the beamline, and the beamline data recorded during the live processing run are available at <http://doi.org/10.5281/ZENODO.8239052> (Weber et al., 2023).

Supplementary Material

To view supplementary material for this article, please visit <http://doi.org/10.5281/ZENODO.8239052>.

Financial Support

- Parts of this research were carried out at beamline P06 of PETRA III at Deutsches Elektronen-Synchrotron DESY, a member of the Helmholtz Association (HGF).
- We acknowledge support from Helmholtz Association under contract No. ZT-I-0025 (Ptychography 4.0) and

JL-MDMC (Joint Lab on Model and Data-Driven Material Characterization).

- This research was supported in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany
- This work was partly funded by the Center for Advanced Systems Understanding (CASUS) which is financed by Germany's Federal Ministry of Education and Research (BMBF) and by the Saxon Ministry for Science, Culture, and Tourism (SMWK) with tax funds on the basis of the budget approved by the Saxon State Parliament.

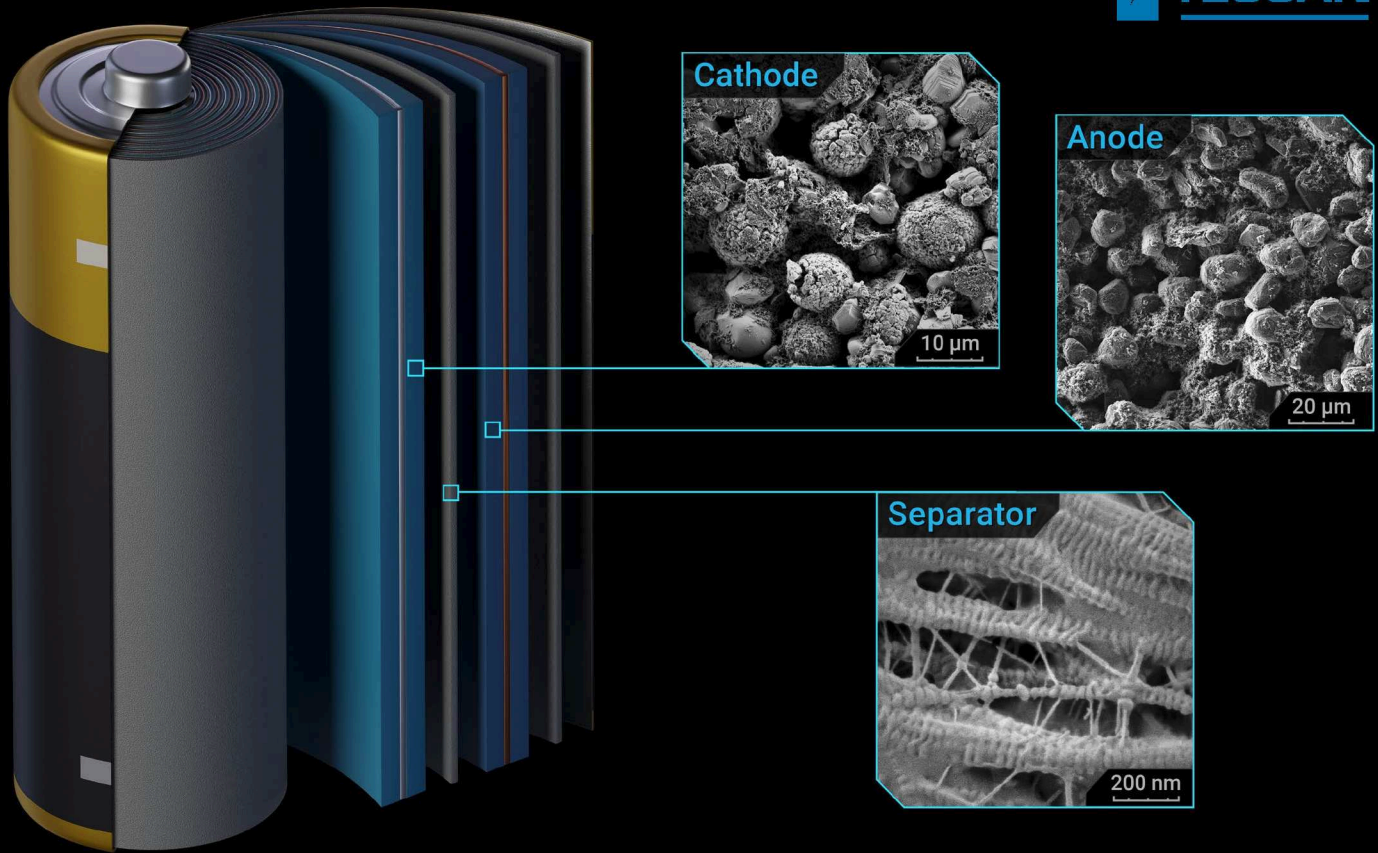
Conflict of Interest

The authors declare no competing interests.

References

- Bangun A, Baumeister PF, Clausen A, Weber D & Dunin-Borkowski RE (2022). Wigner distribution deconvolution adaptation for live ptychography reconstruction, <https://arxiv.org/abs/2212.01309>.
- Ede JM (2021). Adaptive partial scanning transmission electron microscopy with reinforcement learning. *Mach Learn: Sci Technol* 2(4),045011.
- Enders B & Thibault P (2016). A computational framework for ptychographic reconstructions. *Proc R Soc A: Math Phys Eng Sci* 472, 20160640. <https://doi.org/10.1098/rspa.2016.0640>
- Hoppe W (1969a). Beugung im inhomogenen primärstrahlwellenfeld. I. Prinzip einer phasenmessung von elektronenbeugungsinterferenzen. *Acta Crystallogr Sec A* 25(4),495–501. <https://doi.org/10.1107/S0567739469001045>.
- Hoppe W (1969b). Beugung im inhomogenen primärstrahlwellenfeld. III. Amplituden- und phasenbestimmung bei unperiodischen objekt. *Acta Crystallogr Sec A* 25(4),508–514. <https://doi.org/10.1107/S0567739469001069>.
- Hoppe W & Strube G (1969). Beugung in inhomogenen primärstrahlwellenfeld. II. Lichtoptische analogieversuche zur phasenmessung von gitterinterferenzen. *Acta Crystallogr Sec A* 25(4),502–507. <https://doi.org/10.1107/S0567739469001057>.
- Kleinrock L (1964). Analysis of a time-shared processor. *Nav Res Logist Q* 11(1),59–73. <https://doi.org/10.1002/nav.v11:1>
- Könnecke M, Akeroyd FA, Bernstein HJ, Brewster AS, Campbell SI, Clausen B, Cottrell S, Hoffmann JU, Jemian PR, Männicke D, Osborn R, Peterson PF, Richter T, Suzuki J, Watts B, Wintersberger E & Wuttke J (2015). The nexus data format. *J Appl Crystallogr* 48(1),301–305. <https://doi.org/10.1107/S1600576714027575>
- Maiden AM & Rodenburg JM (2009). An improved ptychographical phase retrieval algorithm for diffractive imaging. *Ultramicroscopy* 109(10),1256–1262. <https://www.sciencedirect.com/science/article/pii/S0304399109001284>. <https://doi.org/10.1016/j.ultramic.2009.05.012>
- Matthes A, Widera R, Zenker E, Worpitz B, Huebl A & Bussmann M (2017). Tuning and optimization for a variety of many-core architectures without changing a single line of implementation code using the alpaka library. In *High Performance Computing*, Kunkel JM, Yokota R, Taufer M, & Shalf J (Eds.), pp. 496–514, Cham: Springer International Publishing.
- Melnik O (2023). Convergence properties of gradient methods for blind ptychography. arXiv.
- Nashed YSG, Vine DJ, Peterka T, Deng J, Ross R & Jacobsen C (2014). Parallel ptychographic reconstruction. *Opt Express* 22(26),32082. <https://doi.org/10.1364/OE.22.032082>
- Pelz PM, Johnson I, Ophus C, Ercius P & Scott MC (2021). Real-time interactive 4D-STEM phase-contrast imaging from electron event representation data: Less computation with the right representation. *IEEE Signal Process Mag* 39(1),25–31. <https://doi.org/10.1109/MSP.2021.3120981>

- Rodenburg JM & Faulkner HM (2004). A phase retrieval algorithm for shifting illumination. *Appl Phys Lett* 85(20),4795–4797. <https://doi.org/10.1063/1.1823034>
- Schroer CG, Kurapova O, Patommel J, Boye P, Feldkamp J, Lengeler B, Burghammer M, Riekel C, Vincze L, van der Hart A & Kuchler M (2005). Hard X-ray nanoprobe based on refractive X-ray lenses. *Appl Phys Lett* 87(12),124103. http://apl.aip.org/applab/v87/i12/p124103_s1. <https://doi.org/10.1063/1.2053350>
- Schropp A, Boye P, Feldkamp JM, Hoppe R, Patommel J, Samberg D, Stephan S, Giewekemeyer K, Wilke RN, Salditt T, Gulden J, Mancuso AP, Vartanyants IA, Weckert E, Schöder S, Burghammer M & Schroer CG (2010). Hard X-ray nanobeam characterization by coherent diffraction microscopy. *Appl Phys Lett* 96(9),091102. <http://link.aip.org/link/?APPLAB/96/091102/1>, NULL. <https://doi.org/10.1063/1.3332591>
- Schropp A, Döhrmann R, Botta S, Brückner D, Kahnt M, Lyubomirskiy M, Ossig C, Scholz M, Seyrich M, Stuckelberger ME, Wiljes P, Wittwer F, Garrevoet J, Falkenberg G, Fam Y, Sheppard TL, Grunwaldt JD & Schroer CG (2020). PtyNAMi: Ptychographic nano-analytical microscope. *J Appl Crystallogr* 53(4),957–971. <https://doi.org/10.1107/S1600576720008420>.
- Schropp A & Schroer CG (2010). Dose requirements for resolving a given feature in an object by coherent X-ray diffraction imaging. *New J Phys* 12(3),035016. <https://doi.org/10.1088/1367-2630/12/3/035016>.
- Strauch A, Weber D, Clausen A, Lesnichaia A, Bangun A, März B, Lyu FJ, Chen Q, Rosenauer A & Dunin-Borkowski R (2021). Live processing of momentum-resolved STEM data for first moment imaging and ptychography. *Microsc Microanal* 27(5), 1078–1092. <https://doi.org/10.1017/S1431927621012423>
- Velazco A, Nord M, Béch e A & Verbeeck J (2020). Evaluation of different rectangular scan strategies for STEM imaging. *Ultramicroscopy* 215, 113021. <https://www.sciencedirect.com/science/article/pii/S0304399120300334>. <https://doi.org/10.1016/j.ultramic.2020.113021>
- Weber D, Ehrig S, Schropp A, Clausen A, Achilles S, Hoffmann N, Busmann M, Dunin-Borkowski RE & Schroer CG (2023). Live iterative ptychography. <https://doi.org/10.5281/zenodo.10390322>
- You S, Lu PH, Schachinger T, Kovács A, Dunin-Borkowski RE & Maiden AM (2023). Lorentz near-field electron ptychography. *Appl Phys Lett* 123(19), 192406. <https://doi.org/10.1063/5.0169788>



TESCAN Solutions for Battery Industry

Power your Advanced Battery Technology
and Research with TESCAN Solutions

info.tescan.com/batteries



Scan for more information