A modular software framework for the design and implementation of ptychography algorithms

Francesco Guzzi[1], George Kourousias[1], Fulvio Billè[1], Roberto Pugliese[1], Alessandra Gianoncelli[1] and Sergio Carrato[2]

[1] Elettra Sincrotrone Trieste, Basovizza, Italy

[2] Dipartimento di Ingegneria e Architettura, Università degli studi di Trieste, Trieste, Friuli Venezia Giulia, Italy

ABSTRACT

Computational methods are driving high impact microscopy techniques such as ptychography. However, the design and implementation of new algorithms is often a laborious process, as many parts of the code are written in close-to-the-hardware programming constructs to speed up the reconstruction. In this article, we present Sci-ComPty, a new ptychography software framework aiming at simulating ptychography datasets and testing state-of-the-art and new reconstruction algorithms. Despite its simplicity, the software leverages GPU accelerated processing through the PyTorch CUDA interface. This is essential for designing new methods that can readily be employed. As an example, we present an improved position refinement method based on Adam and a new version of the rPIE algorithm, adapted for partial coherence setups. Results are shown on both synthetic and real datasets. The software is released as open-source.

INTRODUCTION

Today, modern microscopy fully relies on cutting edge computing. The contemporary presence of a large Field Of View (FOV), high resolution and quantitative phase information is the distinctive characteristic of the ptychography diffraction imaging technique (*Rodenburg & Faulkner, 2004*; *Pfeiffer, 2018*). This phase retrieval scheme is commonly solved through a computationally intensive procedure—*a reconstruction*—which requires algorithms written in a low-level language or following complex HPC paradigms. Writing such programs is typically done by an expert software engineer, and this step may discourage the prototyping or the study of new methods. Ptychography algorithms are complex not only in their implementation, but also *per se*: simple computational systems suffers from setup inconsistencies, which usually takes the form of bad parameter modelling of *positions* (*Zhang et al., 2013*), *distances* (*Guzzi et al., 2021b*), *illumination conditions* (*Thibault & Menzel, 2013*), just to enumerate a few. Such problems produce very noticeable artefacts. To improve the quality of the reconstructions, parameter tweaking

becomes essential (*Reinhardt & Schroer, 2018*), especially for common setups; powerful but even more advanced and complex algorithms (*Enders & Thibault, 2016*; *Mandula et al., 2016*; *Marchesini et al., 2016*) are thus employed.

State of the art

Iterative algorithms such as ePIE (*Maiden & Rodenburg, 2009*) or DM (*Thibault et al., 2008*) are typically employed for the reconstructions. Recently, the rPIE algorithm (*Maiden, Johnson & Li, 2017*) has been proposed and studied: while being reasonably simple, it provides a fast convergence to a good object estimate (*Maiden, Johnson & Li, 2017*). Compared to ePIE, we noticed that the rPIE algorithm, at least in our implementation, also provides a large computational FOV, which is comparable to the one seen in more advanced, but also computational eager, optimisation algorithms (*Guizar-Sicairos & Fienup, 2008*; *Thibault & Guizar-Sicairos, 2012*; *Guzzi et al., 2021b*). Indeed, these latter methods are used to refine a previous reconstruction, as they are more prone to stagnation (*Thibault & Guizar-Sicairos, 2012*). Being new, the rPIE algorithm is currently lacking of: (i) a public implementation; (ii) a model decomposition approach; (iii) a tested position refinement routine. As a whole, a *recipe* is missing for this kind of algorithm.

Proposed framework - SciComPty

In this article, we describe how these three elements can be combined within the SciComPty GPU framework, a new software released as open-source (*Guzzi et al., 2021a*), entirely written in the PyTorch (*Paszke et al., 2019*) Python dialect. Our reconstruction recipe is tested against other solutions and then the results are reported. We designed a fast position refinement technique exploiting Adam (*Kingma & Ba, 2015*) as a feedback controller. To do so, a fast subpixel registration algorithm (*Guizar-Sicairos, Thurman & Fienup, 2008*) has also been implemented *via* a PyTorch GPU code. This latter element has many uses, for example in CT alignment (*Guzzi et al., 2021c*), or super-resolution imaging (*Guarnieri et al., 2021*; *Guzzi et al., 2018*). The details are described in the "method section". The software and algorithms capabilities are illustrated ("Results section") with reconstructions from simulated and real soft-X-ray data acquired at TwinMic (Elettra) (*Gianoncelli et al., 2016*; *Gianoncelli et al., 2021*). Within the "Discussion section", we will elaborate more on the features of the reconstructed images. The datasets and the code can be accessed at *Guzzi et al. (2021a)* and *Kourousias et al. (2022)*.

Background

Ptychography aims at recovering the 2D specimen transmission function $O(x,y) \in \mathbb{C}$ starting solely from a set of diffraction patterns (*Williams et al., 2006*; *Pfeiffer, 2018*). In a transmission setup (Fig. 1), a coherent and monochromatic wavefield $P(x,y) \in \mathbb{C}$ is shined onto the specimen ($z = z_O$), placed between the source and the detector ($z = z_d$). In the thin sample approximation (*Paganin, 2006*), the *exit-wave* $\psi_{exw}(x,y)$ transmitted by the object becomes:

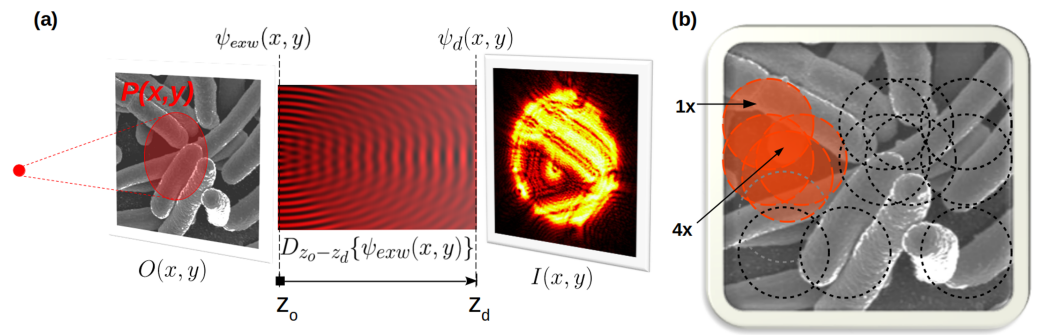$$\psi_{exw}(x,y) = \psi(x,y,z_o) = P(x,y) \cdot |O(x,y)| \cdot e^{j \cdot \phi_{O(x,y)}}. \tag{1}$$

Figure 1 (A) $P(x,y)$ is shined onto an object $O(x,y)$, producing $\psi_{exw}(x,y)$. At the detector we observe $|\psi_d(x,y)|^2$. (B) ptychography uses overlapped illumination (*e.g.*, 4x oversampling) to solve the phasing. The model in Eq. (1) is repeated for each illuminated region.

Full-size 🖼 DOI: 10.7717/peerjcs.1036/fig-1

During the path from the sample to the detector, free-space propagation occurs (*Paganin, 2006*), modelled by the D_z operator (*Schmidt, 2010*):

$$\psi_d(x,y) = \psi(x,y,z_d) = D_{z_d - z_o}\{\psi_{exw}(x,y)\} = D_{z_d - z_o}\{P(x,y,z_o) \cdot O(x,y)\}. \tag{2}$$

Finally, a 2D photon detector is used to reveal the intensity (magnitude squared) of the incident radiation, producing a real, discrete, and quantised image $I(x,y)$, which is the diffraction pattern.

$$I(x,y) = \psi_d(x,y) \cdot \psi_d^*(x,y). \tag{3}$$

Ptychography employs translation diversity to over-condition the problem; a large illumination (*probe*) is indeed scanned across the sample, by imposing a large overlap on each illuminated region (Fig. 1B); as a result, the object is highly oversampled. A ptychography dataset (real or simulated) is thus composed of multiple diffraction patterns, paired with their corresponding translation coordinates. Geometry and illumination metadata complete the description of a particular instance of the computational forward model.

During the reconstruction process, the model described by Eq. (3) must be inverted to gather back the object transmission function $O(x,y)$. Since this inversion cannot, in general, be done analytically, an iterative procedure is employed. Modern versions of the ptychography reconstruction algorithms (*Thibault et al., 2008*; *Maiden & Rodenburg, 2009*) automatically perform the factorisation of the $P(x,y)$ and $O(x,y)$ functions. In a typical imaging-oriented ptychography experiment like ours, the recovery of $P(x,y)$ is just a by-product, though mandatory for the O–P factorisation.

METHODS

Virtual experiment
Reconstructions apart, SciComPty offers a simple method to perform virtual experiments. To simulate a ptychography dataset, one can implement the general transmission model

Guzzi et al. (2022), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.1036

3/17

of Eq. (3), and assign particular values to its setup parameters. In the case of a far-field setup (*Schmidt, 2010*), by fixing the detector pixel size δ_D, the corresponding pixel size at the sample plane δ_s can be readily calculated (equation 1 in the supplementary material). The recurring parameters in both simulations and reconstructions are W (detector size in pixels), S_d (detector lateral dimension) and z_{do} (sample-detector distance). In a simulation, δ_s normally defines the basis for the scanning movements, that in the simplest case follows a grid pattern. Random jitter is added to the (x, y) coordinates, to prevent the *raster scanning pathology* (*Thibault et al., 2008*; *Dierolf et al., 2010*; *Edo et al., 2013*). The overlap factor is defined instead by the maximum step movement. The object function $O(x, y)$ and the illumination function $P(x, y)$ are assembled in magnitude and phase providing two images each. Defining such virtual experiment parameters follows what is actually done during a real experiment (see Listing 1 in the Simulation section of supplementary material).

M-rPIE algorithm

In order to describe partial coherece beams (*Chen et al., 2012*) the wavefield is decomposed into a set of mutually incoherent probe modes (*Thibault & Menzel, 2013*; *Odstrcil et al., 2016*; *Li et al., 2016*). For each sample position (x_j, y_j), the *jth* diffraction pattern $I_j(x, y)$ can be modeled by:

$$I_j(x, y) = \sum_{p=1}^{M} |D_z\{P_p(x, y) \cdot O(x, y, x_j, y_j)\}|^2 = \sum_{p=1}^{M} |D_z P_p(x, y) \cdot o_j(x, y)|^2 \tag{4}$$

where P_p with $p \in \{1, \ldots, M\}$ is a particular probe mode; $\mathbf{o_j} = o_j(x, y) = O(x, y, x_j, y_j)$ is the cropped region of the object, corresponding to the *jth* illuminated area in the scan sequence. To reconstruct an object with the model in Eq. (4), we solve for $O(x, y)$ and *a set* of M mutually incoherent probe functions $\mathbf{P_p} = P_p(x, y)$. A public implementation of rPIE is currently missing and its multiprobe variant has not been reported yet. We implemented it in SciComPty, calling it M-rPIE. The procedure to design the method follows what has been done for one of the multimode-ePIE (M-ePIE) versions (*Thibault & Menzel, 2013*), producing the following new update steps (2D arrays depending on x, y are in **bold**):

$$\mathbf{P'_p} = \mathbf{P_p} + \alpha_P \frac{(\boldsymbol{\psi}'_{exw_{p,j}} - \mathbf{P_p} \cdot \mathbf{o_j}) \cdot \mathbf{o_j^*}}{\beta |\mathbf{o_j}|^2_{\max} + (1-\beta) |\mathbf{o_j}|^2}; \tag{5}$$

$$\mathbf{o'_j} = \mathbf{o_j} + \alpha_O \frac{\sum_{p=1}^{M} (\boldsymbol{\psi}'_{exw_{p,j}} - \mathbf{P_p} \cdot \mathbf{o_j}) \cdot \mathbf{P_p^*}}{\gamma (\sum_{p=1}^{M} |\mathbf{P_p}|^2)_{\max} + (1-\gamma) \sum_{p=1}^{M} |\mathbf{P_p}|^2}. \tag{6}$$

As usual, $\mathbf{o'_j} = o'_j(x, y)$ and $\mathbf{P'_p} = P'_p(x, y)$ represent the updated quantities respectively for the current object box and the current probe, while

$$\psi'_{exw_{p,j}} = D_{-z}\{\sqrt{\mathbf{I}_j} \frac{\psi_{exw_{p,j}}}{\sqrt{\sum_{n=1}^{N}|\psi_p|^2}}\} \tag{7}$$

is the magnitude-corrected scattering wave produced by the *pth* illumination mode; α_p and α_o represent the update rates of the probe and object estimates and are typically set to [0.5, 1]. The role of each denominator is to weight the update where the exit wave is not bright (*Rodenburg & Faulkner, 2004*; *Maiden, Johnson & Li, 2017*), reducing thus the noise amplification. β and γ are regularisation parameters: when set to 1, M-rPIE simplifies to M-ePIE (*Thibault & Menzel, 2013*; *Maiden, Johnson & Li, 2017*). The net effect of the modifications, weighted by β and γ, is to slow down the update, by introducing a loss cost given by $o_j(x,y)$ and $P_p(x,y)$. Increasing the number of modes increases the iteration time by the same factor: that is why it is extremely important the fact that SciComPty is actually working on GPUs.

Adam-based position refinement

As the resolution increases, the effects of backlash and limited mechanical precision become more and more noticeable. The positions vector is indeed populated by the open-loop commands given to the stage. A correction method must then be introduced *a posteriori*, increasing the reconstruction time. In *Guizar-Sicairos & Fienup (2008)*; *Guzzi et al. (2021b)* positions are corrected through an optimisation method, within the gradient-based reconstruction; in *Mandula et al. (2016)*, instead, the authors propose to use a gradient-less method based on *Powell (1964)* to guide the position refinement procedure. In this work, we consider a fast and reliable way to correct translations. Similarly to *Zhang et al. (2013)*; *Tripathi, McNulty & Shpyrko (2014)*; *Loetgering et al. (2015)*; *Dwivedi et al. (2018)*, it employs a 2D cross correlation signal, calculated at some points in the reconstruction chain. The same principle is used also for CT alignment in *Gürsoy et al. (2017)*; *Guzzi et al. (2021c)*, as the synthesised projection $o_j(x,y)$ are inevitably *centred*; when confronted with refined estimates ($o'_j(x,y)$, $XCORR_A$), or measured data ($I(x,y)$, $XCORR_B$), geometrical shifts can then be measured. Figure 2 shows how such position refinement scheme can be introduced in the canvas of a PIE reconstruction algorithm: *2D weighted phase correlation* (*Guizar-Sicairos, Thurman & Fienup, 2008*) here is used to determine the shift between two different estimates $\mathbf{o_j}$ and $\mathbf{o'_j}$ (switch in position $XCORR_A$ in Fig. 2, more details later) belonging to the same crop-box (x_j, y_j).

In this class of position refinement methods, the error signal (the 2D argmax of the 2D cross-correlation) is extremely small and provides a correction factor that is cumulated at each iteration. A gain factor η_j is indeed used to provide a correction. Here we propose to use a *spatially variant* and *adaptive* gain factor, that can change iteration after iteration and is probe-dependent. Our subpixel shift detection is based on the work (*Guizar-Sicairos, Thurman & Fienup, 2008*) and its implementation in SciPy (*Jones et al., 2001*). We adapted the same algorithm in PyTorch, allowing for fast GPU computation and avoiding costly move operation between the two memory systems, as the core of the reconstruction
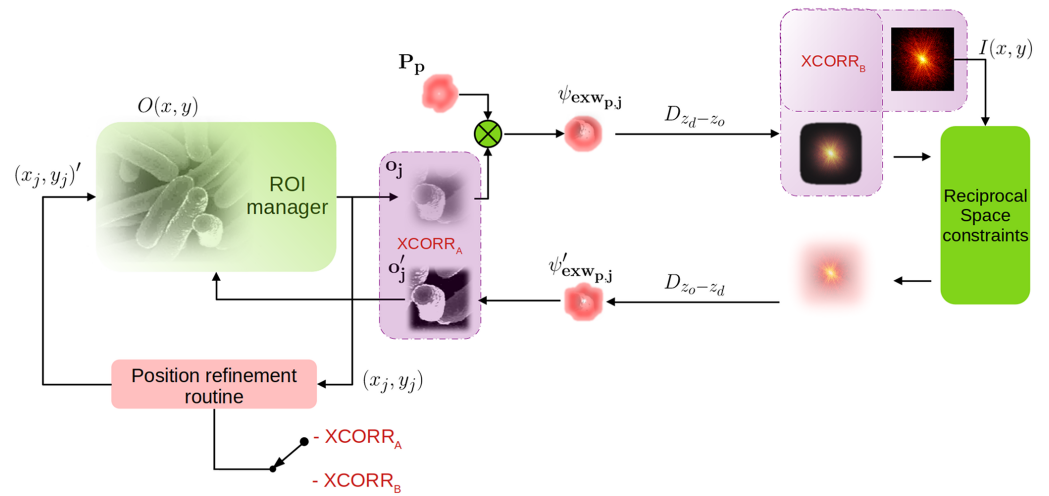
Figure 2   Position refinement routine integrated into a PIE algorithm; the position error signal (argmax of 2D cross-correlation) can be calculated meaningfully in two points, *XCORR_A* or *XCORR_B* (purple boxes). The estimated positions (x, j) are then updated to $(x_j, y_j)'$ iteration by iteration.

Full-size 🖼 DOI: 10.7717/peerjcs.1036/fig-2

(M-rPIE) is already working on the GPU. To speed up the procedure for subpixel scales, the matrix multiplication version of the 2D DFT is employed. An upsampled version of the cross-correlation can then be computed within just a neighbourhood of the coarse peak estimate, without the need for zero-padding. Each probe position is updated at each iteration by the following expressions:

$$x'_j = x_j + \eta_{x,j} \cdot \underset{x}{\mathrm{argmax}}\{XCORR_{i \in [A,B]}\}, \tag{8}$$

$$y'_j = y_j + \eta_{y,j} \cdot \underset{y}{\mathrm{argmax}}\{XCORR_{i \in [A,B]}\}, \tag{9}$$

where $\eta_{x,j}$ and $\eta_{y,j}$ are the gain factors ($>> 1$). The variable gain is calculated by using the Adam optimization algorithm (*Kingma & Ba, 2015*): Adam is an extension to stochastic gradient descent, and is nowadays widely used in deep learning (*Zhang, 2018*). While in Stochastic Gradient Descent (SGD) the same learning rate is kept constant for all the variables, Adam provides a per-parameter factor that is separately adapted as learning unfolds. This is achieved by considering the evolution of each parameter. In our method, the argmax of the 2D cross-correlation takes the place of the gradient in the Adam algorithm, because its value will maximise the position error gradient. The resulting algorithm is then reported in Alg. 1:

Data: The cropped object $\mathbf{o_j}$ and its refined estimate $\mathbf{o_j'}$
Result: Gain parameters $\eta_{x,j}$ and $\eta_{y,j}$ for the *jth* crop-box (the *jth* position vector
$\qquad (x,y)_j$)
Initialization as in Alg. 1 of Adam;
while *reconstruction iteration > 0* **do**
$\quad$ t = t+1;
$\quad$ $g_t = \text{argmax}\{XCORR_{i\in[A,B]}\}$;
$\quad$ proceed as from row 3 of Alg. 1 of Adam;
$\quad$ [...]
end

Algorithm 1: The modified Adam algorithm uses the argmax of the 2D phase correlation as the gradient of the error, then the exponentially damped moving averages are updated as in Adam.

RESULTS

Experiments on simulated and real data were performed. We compared the proposed reconstruction recipe in SciComPty with other state of the art algorithms, also implemented in the advanced *PyNx* software (*Mandula et al., 2016*; *Favre-Nicolin et al., 2020*). Simulated data experiments are reported in the supplementary material and confirm that $XCORR_A$ (see Fig. 2) is the best point to estimate the position error. From those results we defined our strategy for the analysis of real datasets.

Synchrotron soft-X-ray experiments

Real data have been acquired at the TwinMic beamline of the Elettra Synchrotron facility. We used a 1,020 eV (Fig. 3 and Fig. S1) and 1,495 eV (Fig. 4) soft-X-ray beam obtained from a secondary source of 15 μm. The zone plate has a diameter of 600 μm with a smallest ring width of 50 nm. At those energies the focus length is respectively 36 and 24 mm. The sample was placed at 370 μm from the virtual point source, providing a probe size of roughly 9 μm. The sample-detector distance is set at roughly 75 cm. The Princeton camera is based on a peltier-cooled CCD sensor with a resolution of 1,300 × 1,340 pixels, with a pixel size of 20 μm.

Figure 3 shows a series of phase reconstructions of a group of chemically fixed Mesenchymal-Epithelial Transition (MET) cells, grown in silicon nitride windows and exposed to asbestos fibres (*Cammisuli et al., 2018*). The reconstruction in Fig. 3A has been produced by the DM algorithm, while the one in Fig. 3B is the output of the ML algorithm (*Thibault & Guizar-Sicairos, 2012*). In both, the position refinement method (*Mandula et al., 2016*) is enabled. In the same configuration, similar results can be obtained by the AP method (*Marchesini, Tu & Wu, 2016*), that can be seen as a parallelised version of the ePIE algorithm, implemented in PyNx. The reconstruction programs for these images are listed in the Supplemental Information. Even if stunning, images in Figs. 3A and
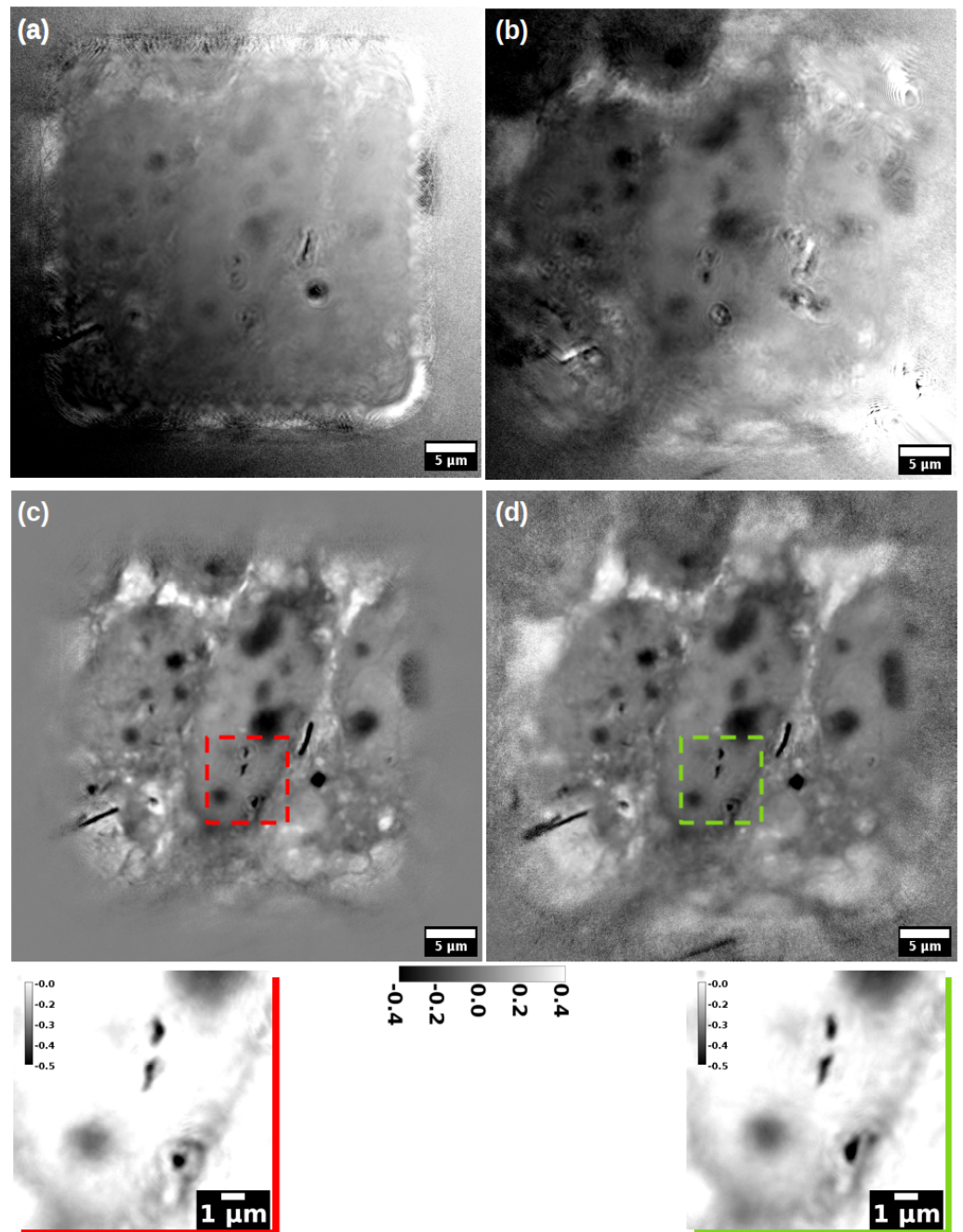
Figure 3 Reconstruction of a MET cells sample. (A) and (B) show respectively the output of DM and ML, paired with the position correction algorithm (*Mandula et al., 2016*). (C) and (D) show the reconstruction with the proposed recipe, using M-ePIE (C) and M-rPIE (D). The insets show how the latter gives fewer ringing artefacts.

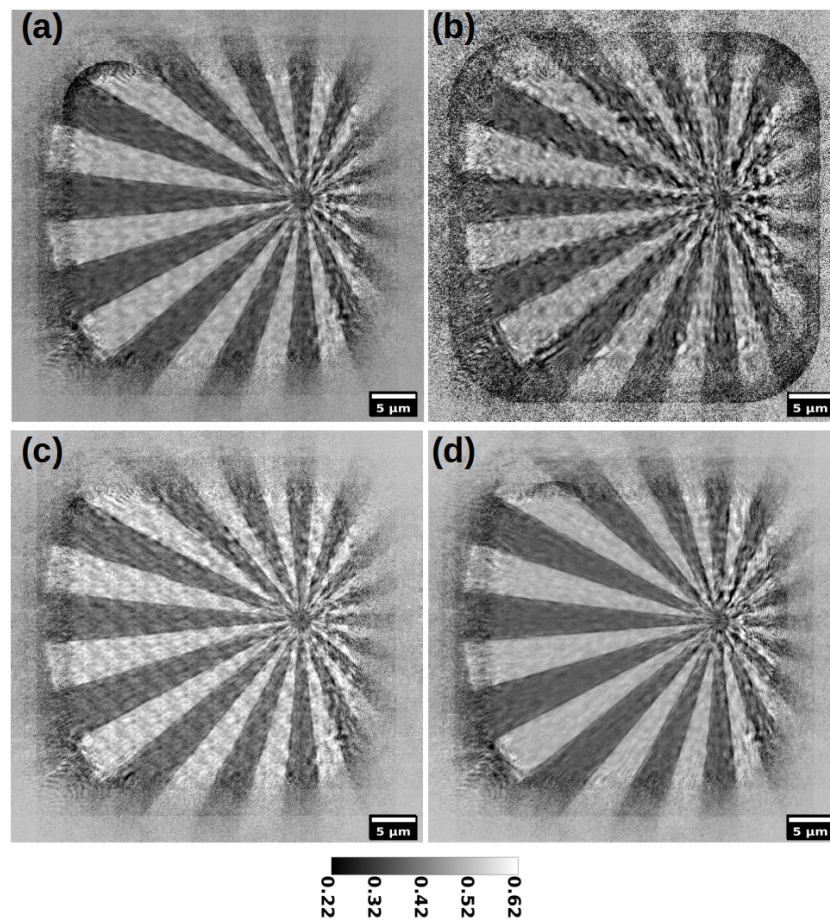Full-size 🖾 DOI: 10.7717/peerjcs.1036/fig-3

Guzzi et al. (2022), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.1036

8/17

Figure 4 Reconstruction of a siemens star (magnitude) with different position refinement methods; (A) *Zhang et al., 2013*; (B) *Mandula et al., 2016*; (C) no correction at all; (D) proposed method.

Full-size 🖼 DOI: 10.7717/peerjcs.1036/fig-4

3B appear blurry and full of artefacts. As expected, the ML algorithm provides a better reconstruction than DM; a larger FOV can also be observed. In both, AP was essential to create a meaningful $P(x, y)$, that allowed the object to appear as "reconstructed". At the end of these reconstructions, it was also required to remove a phase modulation; many details about this post-processing can be found in the supplementary material.

Conversely, Figs. 3C and 3D show the reconstructions obtained with SciComPty, for the same set of parameters and pre-processed data; in either cases, the proposed Adam-based position refinement method is used. Even the simple M-ePIE (Fig. 3C) is able to provide a high quality reconstruction, if paired with the proposed position refinement method. Reconstruction quality can be even increased if the proposed M-rPIE method is engaged (Fig. 3D): the proposed recipe allows to resolve a large FOV, that becomes not only comparable to the one in Fig. 3B, but in some cases (*e.g.*, in the top left corner) even larger: the bottom left fibre can be now observed in its entire length, as well as other cell organelles. No post-processing is required at the end of these reconstructions. The reconstruction for a different region of the MET sample can be found in Fig. S4.
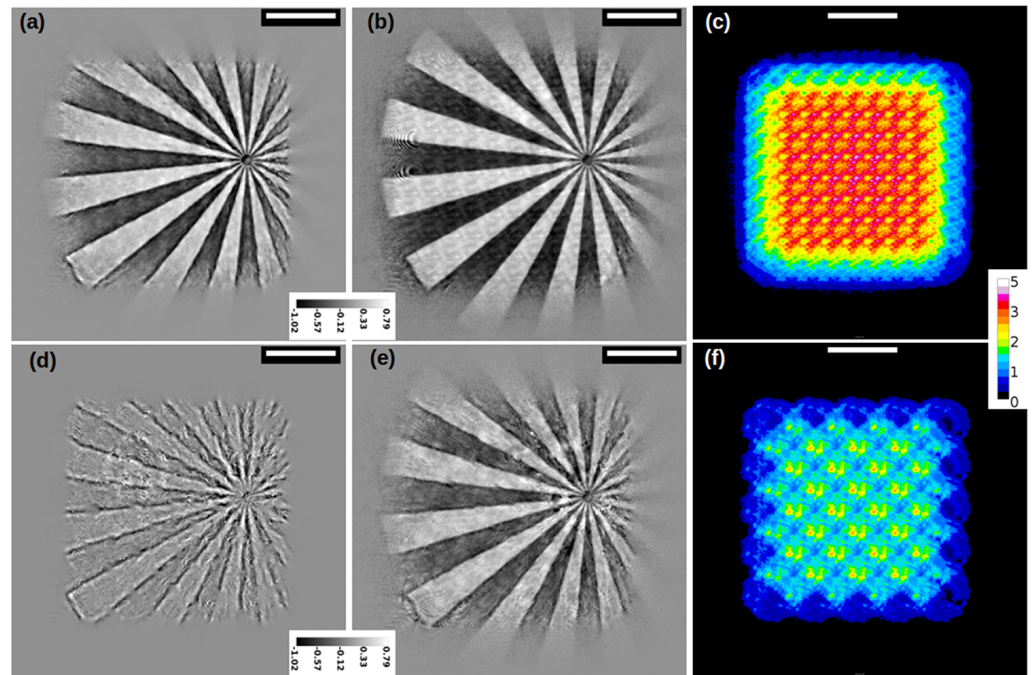
Figure 5 **Reconstruction (phase), obtained using the full dataset (A, B) or a synthetic sparser version (D, E).** M-ePIE is used in (A, D), M-rPIE in (B, E). (C) and (F) show the canvas map for each dataset. The white bar represents 10 μm.

Figure 4 shows the effects of different position correction methods applied on another real dataset, acquired at 1495 eV. Diffraction patterns are acquired following a regular raster grid. Figure 4A shows the reconstruction with the position refinement algorithm in *Zhang et al. (2013)*; in Fig. 4B the algorithm in *Mandula et al. (2016)* is applied; Fig. 4C is the reconstruction output with no position correction applied at all, while in Fig. 4D the proposed Adam-based position correction is applied. As can be seen, even if with simple features, this dataset is quite challenging to correct. While in Fig. 4A a form of correction can be seen, the reconstruction in Fig. 4B is even worse than the one with no correction at all (Fig. 4C). The proposed method (Fig. 4D) estimates the best correction, while being fast. Note that in Fig. 4D the raster grid pathology is quite absent (*actual* positions are not regularly spaced).

A large FOV is the most visible feature in any reconstruction employing M-rPIE (Figs. 3D, 5B, 5E and S4). To better analyse this effect, and to disentangle it from the position correction, a sparser dataset has been synthesised from the one of Fig. 4, producing Figs. 5A and 5D which are respectively the reconstructions with M-ePIE with the full and a sparse dataset (half of the probes); the canvas map (*Kourousias et al., 2016*) in Figs. 5C and 5F shows the two different densities. Figures 5B and 5E show instead the reconstruction with the M-rPIE algorithm: for the same amount of sparsity, the reconstruction is way more resolved in Fig. 5E than in Fig. 5D.
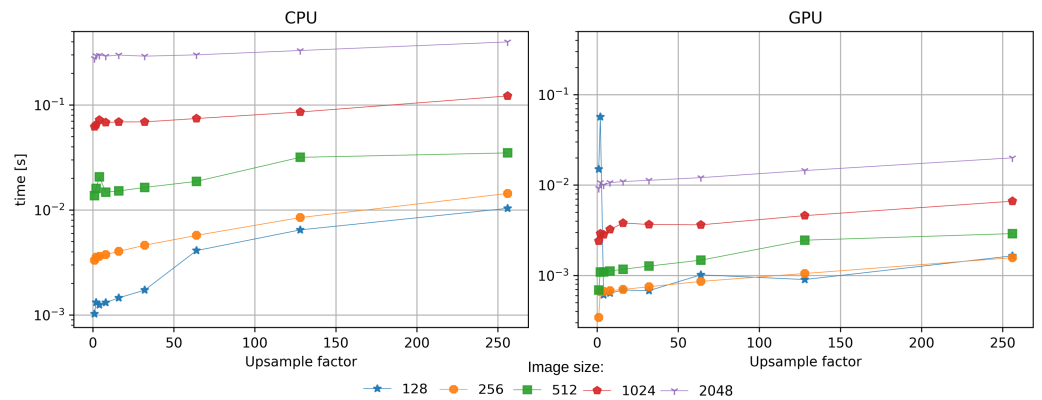
Figure 6 **Benchmark for the position refinement procedure implemented on CPU (*Jones et al., 2001*) or on GPU (proposed method).** The performance gain in terms of speed is about 10x.

Full-size 🖼 DOI: 10.7717/peerjcs.1036/fig-6

Implementation details

SciComPty software framework leverages GPU-based computing for the entire process. The test configuration is reported in the supplementary material. The performance gain obtained by carrying the reconstruction algorithm from the CPU to the GPU is about 10x (0.1 s *vs* 1.3 s). The reason for such a large gain is the high number of GPU cores which allow to parallelize array calculations. With no position correction, the performance in terms of speed are comparable to other CUDA based solvers implemented in PyNx, which is currently the fastest alternative. Regarding the position correction, at the best of our knowledge, no implementation of *Guizar-Sicairos, Thurman & Fienup (2008)* is readily available for GPU computing; consequently, we implemented it in PyTorch GPU. Having part of the algorithm on the GPU(reconstruction) and part on the CPU (registration) would have been detrimental, due to the overhead given by the data transfer. It is indeed extremely important that all the required arrays reside in the same domain, minimising copy operations. Figure 6 shows the computation time measured for the registration algorithm implemented on the two devices: the performance gain is of about an order of magnitude. The acceleration is significant, as during the reconstruction the same operation has to be performed for any position in the dataset.

DISCUSSION

A larger FOV is the most noticeable effect in the reconstructions obtained with M-rPIE (see Fig. 3D and Fig. S4). A well-designed regularisation is considered to be the main candidate for a successful phase-retrieval, that is shown to work better for under-sampled areas (Fig. 5). This effects automatically translates in a better reconstruction in the case of sparser scanning, as the higher resolving power can effectively cope with the missing information between different subsequent measures. Sparse scanning means reduced acquisition and reconstruction times, as well as lower radiation dose, and that is why the implementation of the rPIE algorithm in both single and multi-probe versions is essential for a modern

beamline. The fact that these algorithms are implemented on GPU, in a simple way and using open-source tools, effectively allows to use these algorithms.

The positioning error induced naturally by the sample stage mitigates the raster grid pathology (see Fig. 4), as the actual movements performed by the motors differs from the position commands sent by the control software. Indeed, as backlash, irregular friction and limited mechanical precision introduce a small irregular jitter, it is not required to add it during a scan. Raster grid artefacts, however, *are not* an effect due to position error; the pathology exists and is a consequence of sampling, as they appear also for perfectly ideal simulations with regular grids (Fig. S2). A position refinement method applied *a posteriori* may greatly reduce the artefact, if the actual positions are not regular. That is why a fast implementation of a position refinement algorithm (as deployed by this work) is essential. The Powell method in *Mandula et al. (2016)* tends to fail, as the loss function may be incorrectly estimated by the optimiser. Two are the supposed main characteristics which describe the performances of the Adam-based method: (i) being based on the evolution of the error through many iterations, the gain estimation is more robust than a simple look-back at the previous iteration; (ii) different regions of the sample may require different gains, that have to be adjusted separately.

In CT alignment, the position error estimation is carried out at position $XCORR_B$ (see Fig. 2). In supplementary material we briefly analysed the dynamics of this kind of error signal, concluding that this configuration performs poorly in a ptychograpy environment, compared to the proposed $XCORR_A$: the use of complex quantities provides a more robust estimate than its real-only counterpart; in addiction, images at the object plane tend to be way more intercorrelated than the one at the detector plane, reducing the noise in the cross-correlation.

CONCLUSION

In this article, we presented solutions for three major flaws in ptychography: complex software architectures, partial coherence and position errors. By combining together the proposed solutions, we provide a recipe that is giving good results in many ptychography experiments performed at synchrotron and FEL laboratories (Elettra Sincrotrone Trieste). The development of a multi-mode variant for rPIE is essential to reduce the overlap condition, allowing for sparser measurements and thus providing shorter acquisition and reconstruction time. This is critical for dynamics experiments (*e.g.*, with many energies and pump-probe delays). Reducing the lag between measurements and reconstruction is a critical research path, which has a significant impact on all the fields that use this kind of microscopy. We designed these solutions by employing our new GPU-accelerated ptychography software framework, SciComPty, which can be used to easily study and develop both state-of-the-art and new reconstruction algorithms. Throughout the text, we presented several reconstruction examples from real and synthetic datasets, comparing them to other state-of-the art solutions. The position refinement procedure relies on a fast subpixel registration algorithm that also runs on GPU. The entire software is provided to the research community as open source and can be downloaded from *Kourousias et al. (2022)* and *Guzzi et al. (2021a)*.

ACKNOWLEDGEMENTS

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

Grant Disclosures

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Francesco Guzzi conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- George Kourousias conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Fulvio Billè conceived and designed the experiments, performed the experiments, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Roberto Pugliese conceived and designed the experiments, performed the experiments, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Alessandra Gianoncelli conceived and designed the experiments, performed the experiments, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Sergio Carrato conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:
The raw data and code are available in the Supplemental Files and at Zenodo: Francesco Guzzi, George Kourousias, Fulvio Billè, Roberto Pugliese, Alessandra Gianoncelli, & Sergio

Carrato. (2021). Material concerning a manuscript about two algorithms implemented in the SciCoMPty framework. https://doi.org/10.5281/zenodo.6524326.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.1036#supplemental-information.

REFERENCES

Cammisuli F, Giordani S, Gianoncelli A, Rizzardi C, Radillo L. 2018. Iron-related toxicity of single-walled carbon nanotubes and crocidolite fibres in human mesothelial cells investigated by Synchrotron XRF microscopy. *Scientific Reports* **8(1)**:706 DOI 10.1038/s41598-017-19076-1.

Chen B, Abbey B, Dilanian R, Balaur E, van Riessen G. 2012. Diffraction imaging: the limits of partial coherence. *Physical Review B* **86**:235401 DOI 10.1103/PhysRevB.86.235401.

Dierolf M, Thibault P, Menzel A, Kewish CM, Jefimovs K. 2010. Ptychographic coherent diffractive imaging of weakly scattering specimens. *New Journal of Physics* **12(3)**:035017 DOI 10.1088/1367-2630/12/3/035017.

Dwivedi P, Konijnenberg A, Pereira S, Urbach H. 2018. Lateral position correction in ptychography using the gradient of intensity patterns. *Ultramicroscopy* **192**:29–36 DOI 10.1016/j.ultramic.2018.04.004.

Edo TB, Batey DJ, Maiden AM, Rau C, Wagner U. 2013. Sampling in x-ray ptychography. *Physical Review A* **87**:053850 DOI 10.1103/PhysRevA.87.053850.

Enders B, Thibault P. 2016. A computational framework for ptychographic reconstructions. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **472(2196)**:20160640 DOI 10.1098/rspa.2016.0640.

Favre-Nicolin V, Girard G, Leake S, Carnis J, Chushkin Y, Kieffer J, Paleo P, Richard M-I. 2020. *PyNX*: high-performance computing toolkit for coherent X-ray imaging based on operators. *Journal of Applied Crystallography* **53(5)**:1404–1413 DOI 10.1107/S1600576720010985.

Gianoncelli A, Bonanni V, Gariani G, Guzzi F, Pascolo L, Borghes R, Billè F, Kourousias G. 2021. Soft X-ray microscopy techniques for medical and biological imaging at TwinMic–Elettra. *Applied Sciences* **11(16)**:7216 DOI 10.3390/app11167216.

Gianoncelli A, Kourousias G, Merolle L, Altissimo M, Bianco A. 2016. Current status of the TwinMic beamline at Elettra: a soft X-ray transmission and emission microscopy station. *Journal of Synchrotron Radiation* **23(6)**:1526–1537 DOI 10.1107/S1600577516014405.

Guarnieri G, Fontani M, Guzzi F, Carrato S, Jerian M. 2021. Perspective registration and multi-frame super-resolution of license plates in surveillance videos. *Forensic Science International: Digital Investigation* **36**:301087 DOI 10.1016/j.fsidi.2020.301087.

Guizar-Sicairos M, Fienup JR. 2008. Phase retrieval with transverse translation diversity: a nonlinear optimization approach. *Optics Express* **16(10)**:7264–7278 DOI 10.1364/OE.16.007264.

Guizar-Sicairos M, Thurman ST, Fienup JR. 2008. Efficient subpixel image registration algorithms. *Optics Letters* **33**(2):156–158 DOI 10.1364/OL.33.000156.

Gürsoy D, Hong YP, He K, Hujsak K, Yoo S. 2017. Rapid alignment of nanotomography data using joint iterative reconstruction and reprojection. *Scientific Reports* **7**(**1**):11818 DOI 10.1038/s41598-017-12141-9.

Guzzi F, Kourousias G, Billè F, Pugliese R, Gianoncelli A. 2021a. Material concerning a manuscript about two algorithms implemented in the SciComPty framework. *Zenodo. Available at* https://doi.org/10.5281/zenodo.6524326 (accessed on 15 January 2021).

Guzzi F, Kourousias G, Billè F, Pugliese R, Reis C. 2018. Refining scan positions in Ptychography through error minimisation and potential application of machine learning. *Journal of Instrumentation* **13**(06):C06002–C06002 DOI 10.1088/1748-0221/13/06/c06002.

Guzzi F, Kourousias G, Gianoncelli A, Billè F, Carrato S. 2021b. A parameter refinement method for ptychography based on deep learning concepts. *Condensed Matter* **6**(**4**):36 DOI 10.3390/condmat6040036.

Guzzi F, Kourousias G, Gianoncelli A, Pascolo L, Sorrentino A, Billè F, Carrato S. 2021c. Improving a rapid alignment method of tomography projections by a parallel approach. *Applied Sciences* **11**(**16**):7598 DOI 10.3390/app11167598.

Jones E, Oliphant T, Peterson P. 2001. SciPy: Open source scientific tools for Python. *Available at* http://www.scipy.org/ (accessed on 29 April 2022).

Kingma DP, Ba J. 2015. Adam: a method for stochastic optimization. In: *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings.*

Kourousias G, Bozzini B, Gianoncelli A, Jones MW, Junker M. 2016. Shedding light on electrodeposition dynamics tracked in situ via soft X-ray coherent diffraction imaging. *Nano Research* **9**(**7**):2046–2056 DOI 10.1007/s12274-016-1095-9.

Kourousias G, Guzzi F, Billè F, Pugliese R, Gianoncelli A. 2022. Datasets. Trieste: Elettra Sincrotrone Trieste. *Available at* http://dx.doi.org/10.34965/i10053 (accessed on 29 April 2022).

Li P, Edo T, Batey D, Rodenburg J, Maiden A. 2016. Breaking ambiguities in mixed state ptychography. *Optics Express* **24**(**8**):9038–9052 DOI 10.1364/OE.24.009038.

Loetgering L, Hammoud R, Juschkin L, Wilhein T. 2015. A phase retrieval algorithm based on three-dimensionally translated diffraction patterns. *EPL (Europhysics Letters)* **111**(**6**):64002 DOI 10.1209/0295-5075/111/64002.

Maiden A, Johnson D, Li P. 2017. Further improvements to the ptychographical iterative engine. *Optica* **4**(**7**):736–745 DOI 10.1364/OPTICA.4.000736.

Maiden AM, Rodenburg JM. 2009. An improved ptychographical phase retrieval algorithm for diffractive imaging. *Ultramicroscopy* **109**(**10**):1256–1262 DOI 10.1016/j.ultramic.2009.05.012.

Mandula O, Elzo Aizarna M, Eymery J, Burghammer M, Favre-Nicolin V. 2016. *PyNX.Ptycho*: a computing library for X-ray coherent diffraction imaging of

nanostructures. *Journal of Applied Crystallography* **49(5)**:1842–1848 DOI 10.1107/S1600576716012279.

Marchesini S, Krishnan H, Daurer BJ, Shapiro DA, Perciano T. 2016. *SHARP*: a distributed GPU-based ptychographic solver. *Journal of Applied Crystallography* **49(4)**:1245–1252 DOI 10.1107/S1600576716008074.

Marchesini S, Tu Y-C, Wu H-T. 2016. Alternating projection, ptychographic imaging and phase synchronization. *Applied and Computational Harmonic Analysis* **41(3)**:815–851 DOI 10.1016/j.acha.2015.06.005.

Odstrcil M, Baksh P, Boden SA, Card R, Chad JE. 2016. Ptychographic coherent diffractive imaging with orthogonal probe relaxation. *Optics Express* **24(8)**:8360–8369 DOI 10.1364/OE.24.008360.

Paganin D. 2006. *Coherent X-ray Optics.* Oxford: Oxford University Press.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J. 2019. PyTorch: an imperative style, high-performance deep learning library. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R, eds. *Advances in neural information processing systems 32: NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 8024–8035.

Pfeiffer F. 2018. X-ray ptychography. *Nature Photonics* **12(1)**:9–17 DOI 10.1038/s41566-017-0072-5.

Powell MJD. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal* **7(2)**:155–162 DOI 10.1093/comjnl/7.2.155.

Reinhardt J, Schroer C. 2018. Quantitative ptychographic reconstruction by applying a probe constraint. *Journal of Instrumentation* **13(04)**:C04016–C04016 DOI 10.1088/1748-0221/13/04/c04016.

Rodenburg JM, Faulkner HML. 2004. A phase retrieval algorithm for shifting illumination. *Applied Physics Letters* **85(20)**:4795–4797 DOI 10.1063/1.1823034.

Schmidt JD. 2010. *Numerical simulation of optical wave propagation with examples in MATLAB.* Bellingham, Washington: SPIE.

Thibault P, Dierolf M, Menzel A, Bunk O, David C. 2008. High-resolution scanning X-ray diffraction microscopy. *Science* **321(5887)**:379–382 DOI 10.1126/science.1158573.

Thibault P, Guizar-Sicairos M. 2012. Maximum-likelihood refinement for coherent diffractive imaging. *New Journal of Physics* **14(6)**:063004 DOI 10.1088/1367-2630/14/6/063004.

Thibault P, Menzel A. 2013. Reconstructing state mixtures from diffraction measurements. *Nature* **494(7435)**:68–71 DOI 10.1038/nature11806.

Tripathi A, McNulty I, Shpyrko OG. 2014. Ptychographic overlap constraint errors and the limits of their numerical recovery using conjugate gradient descent methods. *Optics Express* **22(2)**:1452–1466 DOI 10.1364/OE.22.001452.

Williams GJ, Quiney HM, Dhal BB, Tran CQ, Nugent KA. 2006. Fresnel coherent diffractive imaging. *Physical Review Letters* **97**:025506 DOI 10.1103/PhysRevLett.97.025506.

Zhang Z. 2018. Improved Adam Optimizer for Deep Neural Networks. In: *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*. 1–2 DOI 10.1109/IWQoS.2018.8624183.

Zhang F, Peterson I, Vila-Comamala J, Diaz A, Berenguer F. 2013. Translation position determination in ptychographic coherent diffraction imaging. *Optics Express* **21(11)**:13592–13606 DOI 10.1364/OE.21.013592.