Check for updates

SOFTWARE TOOL ARTICLE

REVISED Alrecon: computed tomography reconstruction web application based on Solara

[version 2; peer review: 5 approved, 1 approved with reservations]

Gianluca Iori, Philipp Hans, Ibrahim Foudeh, Mustafa Alzu'bi, Malik Al Mohammad, Salman Matalgah

SESAME - Synchrotron-light for Experimental Science and Applications in the Middle East, Allan, 19252, Jordan

v2 **First published:** 18 Mar 2024, **4**:54
https://doi.org/10.12688/openreseurope.16863.1
Latest published: 28 May 2024, **4**:54
https://doi.org/10.12688/openreseurope.16863.2

Abstract

Synchrotron X-ray computed tomography is a non-destructive 3D imaging technique that offers the possibility to study the internal microstructure of samples with high spatial and temporal resolution. Given its unmatched image quality and acquisition speed, and the possibility to preserve the specimens, there is an increasing demand for this technique, from scientific users from innumerable disciplines. Computed tomography reconstruction is the computational process by which experimental radiographs are converted to a meaningful 3-dimensional image after the scan. The procedure involves pre-processing steps for image background and artifact correction on raw data, a reconstruction step approximating the inverse Radon-transform, and writing of the reconstructed volume image to disk. Several open-source Python packages exist to help scientists in the process of tomography reconstruction, by offering efficient implementations of reconstruction algorithms exploiting central or graphics processing unit (CPU and GPU, respectively), and by automating significant portions of the data processing pipeline. A further increase in productivity is attained by scheduling and parallelizing demanding reconstructions on high performance computing (HPC) clusters.

Nevertheless, visual inspection and interactive selection of optimal reconstruction parameters remain crucial steps that are often performed in close interaction with the end-user of the data. As a result, the reconstruction task involves more than one software. Graphical user interfaces are provided to the user for fast inspection and optimization of reconstructions, while HPC resources are often accessed through scripts and command line interface. We propose

Open Peer Review

Approval Status ✓ ✓ ✓ ? ✓ ✓

	1	2	3	4	5	6
version 2 (revision) 28 May 2024			✓ view	? view	✓ view	✓ view
version 1 18 Mar 2024	✓ view	✓ view				

1. **Viktor Nikitin**, Argonne National Laboratory, Lemont, USA
2. **Edoardo Pasca**, Science Technology Facilities Council, Swindon, UK
3. **Chang-Chieh Cheng**, National Yang Ming Chiao Tung University (Ringgold ID: 34914), Hsinchu, Taiwan
4. **Tristan van Leeuwen**, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
5. **Luisa Vigorelli**, Università di Milano-Bicocca and INFN, Milano, Italy
6. **Vânia Vieira Estrela**, Federal Fluminense University, Niterói, Brazil

Any reports and responses or comments on the article can be found at the end of the article.

Alrecon, a pure Python web application for tomographic reconstruction built using Solara. Alrecon offers users an intuitive and reactive environment for exploring data and customizing reconstruction pipelines. By leveraging upon popular 3D image visualization tools, and by providing a user-friendly interface for reconstruction scheduling on HPC resources, Alrecon guarantees productivity and efficient use of resources for any type of beamline user.

Keywords
computed tomography, tomographic reconstruction, x-ray imaging, image processing, graphical user interface, Python, Solara

H2020 This article is included in the Horizon 2020 gateway.

Corresponding author: Gianluca Iori (gianthk.iori@gmail.com)

Author roles: Iori G: Conceptualization, Data Curation, Formal Analysis, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Hans P**: Data Curation, Software; **Foudeh I**: Data Curation, Resources, Writing – Review & Editing; **Alzu'bi M**: Data Curation, Funding Acquisition, Resources, Supervision, Writing – Review & Editing; **Al Mohammad M**: Data Curation, Writing – Review & Editing; **Matalgah S**: Data Curation, Funding Acquisition, Project Administration, Resources, Supervision, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No [822535] (Beamline for Tomography at SESAME [BEATS]).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2024 Iori G *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Iori G, Hans P, Foudeh I *et al.* **Alrecon: computed tomography reconstruction web application based on Solara [version 2; peer review: 5 approved, 1 approved with reservations]** Open Research Europe 2024, **4**:54 https://doi.org/10.12688/openreseurope.16863.2

First published: 18 Mar 2024, **4**:54 https://doi.org/10.12688/openreseurope.16863.1

REVISED **Amendments from Version 1**

The strengths of the pure Python Solara framework for web app development are introduced in the article introduction. A future developments section was added, reflecting on features of the application currently under development, and important upgrades envisioned for future releases of Alrecon.

Any further responses from the reviewers can be found at the end of the article

Introduction

X-ray Computed Tomography (CT) is a powerful imaging tool mainly known for its impact in the medical field, where it allows healthcare professionals to visualize and analyze the internal anatomy in 3 dimensions, with high precision and clarity. CT scanners emit beams of X-rays that pass through the patient's body. X-ray detectors on the opposite side of the patient measure the intensity of the X-rays after they have passed through the body. Multiple X-ray projections are collected while rotating the X-ray source and detector around the patient. CT reconstruction refers to the computational process used to generate three-dimensional images of the internal structures of an object from sets of raw X-ray projections (also called sinograms).

Synchrotron X-ray Computed Tomography (SXCT) exploits the high brilliance and coherence of synchrotron radiation to provide tomographic microscopy of specimens with spatial and temporal resolution unmatched by medical or laboratory CT scanners. Users from a broad range of disciplines benefit from the capabilities of the several SXCT facilities (beamlines) installed at synchrotrons worldwide. During the normal operation of such laboratories, CT scans of a large variety of samples, with varying geometry and X-ray absorption properties, as well as a multiplicity of in-situ studies of materials can be performed. The process of tomographic reconstruction is performed routinely at SXCT beamlines to provide the users reconstructed volume images at the end of their beamtime and is a crucial step in data processing towards the extraction and publication of meaningful results from raw SXCT datasets.

Compared with laboratory CT, SXCT presents significant differences that are related to the different X-ray source nature, and to the different process of image formation. These differences are reflected in the process of data reconstruction, and in the range of artifacts that can affect SXCT sinograms. They also determine several opportunities that are unique to CT imaging with synchrotron light. The parallel geometry of collimated synchrotron beams implies that SXCT radiographs are generated as parallel projections of the object onto the image-forming, scintillator plane. Accordingly, the reconstruction process involves implementations of the Filtered-Back-Projection or other reconstruction algorithms based on a parallel geometry. On the contrary, image formation and tomographic reconstruction of laboratory or clinical CT are based on X-ray propagation of cone geometry. In SXCT radiographs, bright spots are generated by a variety of factors including dead or damaged detector pixels, dust, imperfections and impurities on the optical surfaces (e.g. mirrors, vacuum containment windows, lenses, scintillator) encountered by the beam along its propagation path, which can be up to hundreds of meters. In particular, the effect of dust and dirt on the scintillator surface is enhanced by the high resolution of the SXCT imaging system. In comparison and due to their large pixel size, the flat-panel detectors used in laboratory and clinical CT are less sensitive to the presence of impurities. In the sinogram space, this family of artifacts is described as stripes with a bright, saturated, or altered detector response. If not treated, such stripes give rise to ring artifacts in the reconstructed 3D image that can undermine image processing. Depending on the cause and features of stripe artifacts, these can be corrected using a variety of different algorithms before 3D reconstruction[1]. Propagation-based phase-contrast X-ray tomography is an advanced imaging technique that exploits the unique coherence properties of X-ray illumination available at synchrotron facilities. Contrary to other approaches for phase-contrast imaging, propagation-based phase-contrast is generated without the need of dedicated equipment, by simply propagating the X-ray beam, after its interaction with the studied specimen, up to several meters before detection[2]. Its straightforward implementation has made the technique an almost omnipresent feature of modern SXCT beamlines. Compared with standard, absorption-based CT, phase-contrast SXCT can achieve superior contrast to low or similarly X-ray absorbing materials, and a general reduction of the radiation dose deposited in the sample[3]. The reconstruction of propagation-based phase-contrast tomograms requires an additional phase-retrieval step[2], which is applied to the sinogram prior to reconstruction. Another key advantage of CT with highly brilliant synchrotron radiation is the significant reduction of acquisition time, which enables time-resolved tomographic studies, with acquisition rates from a few Hz up to the kHz range[4]. Time-resolved, in-situ SXCT has become an invaluable technique for observing in a non-destructive manner phenomena in 3D materials related to energy production and storage[5,6], biomedical engineering[7,8], geology and earth sciences[9,10], and more. The drastic reduction of scan duration poses an additional challenge for CT reconstruction, since the time needed for dataset exploration and visualization can become the bottleneck in the experimental workflow. With data collection becoming faster, there is an increasing need for tools providing fast CT reconstruction and results visualization.

Due to the unique characteristics of synchrotron tomography, several open-source Python packages have been designed, in part or entirely, for the processing and reconstruction of SXCT datasets. These include TomoPy[11], TomoCuPy[12], Astra[13], Savu[14], and Nabu[15]. These packages provide a range of tools and algorithms for sinogram manipulation, artifact correction, CT reconstruction, and post-processing operations. All these Python implementations of reconstruction software can be installed and scaled relatively easily (e.g. through app containers) for use on high performance computing (HPC) clusters allowing large-scale parallelization of processes across several central processing units (CPU), improving the 3D reconstruction performance dramatically. In addition,

they allow enhancing the performance of particularly computationally intensive algorithms such as iterative reconstruction, by exploiting graphics processing unit (GPU)-accelerated versions of these. Despite extensive online documentation, the user of such packages is expected to be fluent in Python, as much as required by the customization and testing of complete reconstruction pipelines. In this context, the recent development of Solara[16], a framework for the development of large-scale web apps within a pure Python ecosystem, offers a unique opportunity for closing the existing gap between efficient reconstruction Python libraries and non-expert users of SXCT.

In the past, toolkits such as Tofu[17] and STP[18] have been developed with the goal of providing user-friendly interfaces to CT reconstruction software. These are optimized for application on stand-alone workstations equipped with GPU. Dealing with large amounts of data and processes, such as those involved during a SXCT beamtime, using stand-alone equipment, presents several drawbacks:

- First, large chunks of experimental data must be transferred via local area network (LAN) to the reconstruction workstation. This introduces a workflow bottleneck and, in the worst case, can lead to data duplication. State-of-the-art SXCT laboratories, on the contrary, aim at eliminating experimental data copy and transfer. As an example, the data acquisition and computing infrastructure of the recently inaugurated beamline ID10-BEATS of SESAME, deploys high-throughput detectors able to generate massive experimental files on a centralized general parallel file system (GPFS) storage facility[19]. The use of a hierarchical data format (HDF5), and 100 GBs connectivity between store and a hybrid HPC reconstruction cluster, eliminates redundancy and maximizes data transfer rates.

- Second, the necessity to inspect reconstructions immediately after the scan requires computing resources to be permanently available for data exploration and parameter optimization. Even utilizing fast GPU implementations of reconstruction routines, processing massive scans can take several minutes and hundreds of dedicated volatile memory, leading to inevitable multiplication of resources. In this context, a wiser allocation of resources seems to be a workstation dedicated to data exploration, while a queue of reconstruction jobs is handled by a job scheduling system such as Slurm[20] on the nodes of a HPC cluster.

Statement of need

Alrecon (الريكُن), a user-friendly open-source Solara web application for SXCT data reconstruction with TomoPy is presented[21].

- In a single Python web application, Alrecon allows to inspect and optimize CT reconstructions interactively, and submit demanding reconstruction jobs to a HPC cluster, while keeping a flawless log of these operations.

- For 3D sinogram and reconstruction visualization, Alrecon leverages on napari and ImageJ, two powerful toolkits known by a vast community of users.

- The Solara framework makes Alrecon intuitive, while keeping the app robust and flexible.

- The modular, pure Python implementation of Alrecon allows expert users to scale and customize the app to their infrastructure.

Alrecon aims at bridging the gap between the large number of tools offered by reconstruction Python packages, and the routine exploration and processing of a large number and variety of SXCT datasets. The target user of Alrecon is the experienced SXCT beamline staff, as well as researchers or students lacking programming experience.

Methods

Implementation: Alrecon pipeline

Figure 1 illustrates the pipeline for SXCT data reconstruction implemented by Alrecon. During the routine operation of the tomography beamline ID10- of SESAME, hierarchical HDF5 experimental files containing X-ray projections, reference flat and dark images, as well as a metadata archive of instrument and acquisition parameters, are loaded from a centralized GPFS short-term storage[22].

During a scan at BEATS, the raw experimental data (X-ray projections) is streamed by the detector directly to the central GPFS storage[19]. Raw experimental data and metadata is written on a common HDF5 file that is generated by a data acquisition system on the GPFS storage, following the Scientific Data Exchange community standard[23]. Experimental file duplication or modification after conclusion of the scan is not allowed.

Pre-processing and reconstruction of sinograms. The sinogram is loaded as a numpy 3-dimensional array and treated in different steps before reconstruction. Background normalization using the collected flat and dark frames can be generally performed automatically after loading the dataset. Optional modules allow to correct different types of sinogram stripes, and to retrieve the X-ray phase shift from absorption-contrast X-ray projections of the object. An Alrecon page dedicated to the optimization of the center of tomographic rotation (COR) parameter for the reconstruction is available. In most cases, the user is expected to inspect visually and confirm the choice of COR before proceeding to CT reconstruction. Once the COR parameter is found, and the sinogram pre-processed, the reconstruction procedure is launched. At any step of the pipeline, napari and ImageJ can be called to inspect the processed sinogram, or the results of COR optimization and CT reconstruction.

Post-processing of reconstructed data. Two basic operations can be applied before writing data to disk: (i) the dataset can be converted to 8-bit or 16-bit unsigned integer formats, and (ii) a cylinder masking filter can be applied to remove the

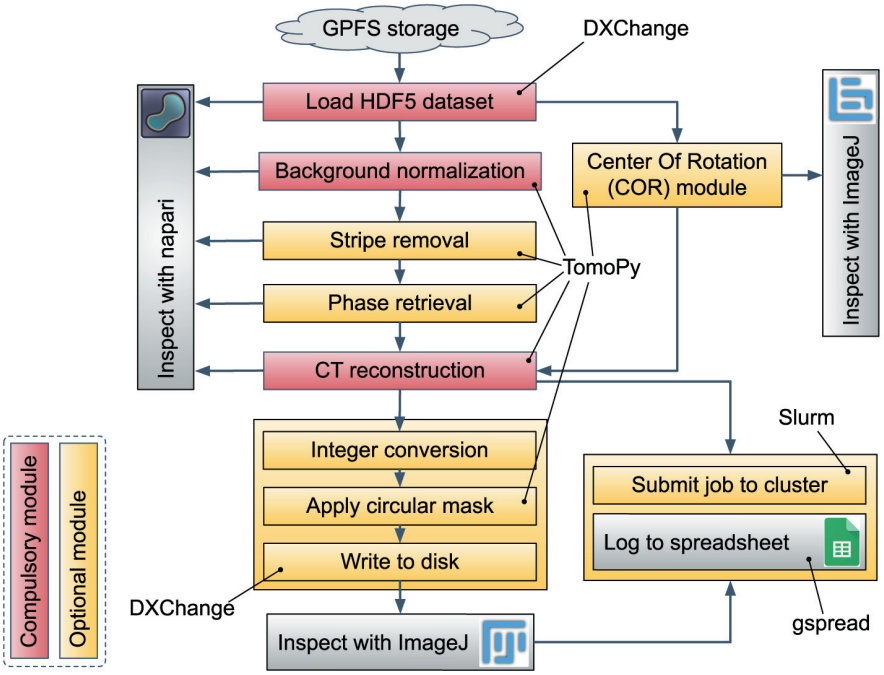


Figure 1. CT reconstruction pipeline implemented in Alrecon. Red: compulsory modules. Yellow: optional modules.

corners of the 3D image. The best way to verify that the process was successful, and that all parameters are correct, is to inspect the final output with ImageJ.

Launching reconstruction jobs on HPC cluster. The pipeline described so far can be executed on a subset of the experimental file in just a few minutes (see next section for a detailed description). Once all parameters are optimized, a complete reconstruction job can be submitted to a HPC reconstruction cluster. At SESAME BEATS, the resources of one hybrid CPU-GPU HPC node equipped with 512 GB RAM are allocated permanently for this task during beamtime. In this way, a queue of large reconstructions can be handled on the main computing facility, taking advantage of the high-speed connection between GPFS storage and reconstruction cluster, while Alrecon remains available to the user for dataset and parameter exploration immediately after a new scan is completed.

The Alrecon ecosystem. Alrecon's functionalities are based on a set of open-source software and Python packages for web application reactiveness, data Input/Output, CPU and GPU tomographic reconstruction, process logging, and 3D image visualization. Table 1 summarizes the software ecosystem that supports Alrecon, with a description of the function provided by external software.

Operation: the Alrecon web application

The main page of the Alrecon web application is shown in Figure 2. On the left, a sidebar (Figure 2-A) displays experimental dataset information, locations for writing of output files and the name of the beamtime master Google spreadsheet where reconstruction parameters will be logged. Two additional cards on the sidebar allow launching ImageJ and napari for visualizing results.

Experimental raw data stored in HDF5 files can be loaded as a whole or in chunks. The load dataset card (Figure 2-B) allows the user to select sinogram (detector lines) and projections ranges before retrieving the data. Parameter optimization and reconstruction setup can be achieved in most cases on a subset of the sinogram. In this way, the time for file read and computation become practically negligible, and users experience an almost real-time visualization of results. This situation is desirable during the preliminary data exploration phase, in which several reconstruction attempts followed by visual checks of results are needed to find optimal parameters. Background normalization and a preliminary guess of the COR can be executed automatically after loading the data.

A phase retrieval card (Figure 2-C) allows to process the sinogram and extract the phase shift induced by the specimen on the beam front, by applying the filter first described by Paganin *et al.*[2]. Parameters describing the experiment setup required for the phase retrieval operation (i.e. object pixel size, sample-to-detector distance, and beam energy) are extracted automatically from the metadata archive associated with the loaded dataset. Normally, the user will tune the regularization parameter α:

$$\alpha = 1 / (4\pi^2 \delta / \beta)$$

where δ/β is the ratio between the real and imaginary parts of the complex refractive index of the material.

Table 1. Alrecon ecosystem.

Package/ Software name	Function	Reference
Solara	Solara is the backbone of the Alrecon application. It allows running Alrecon as a standalone, React-style web application, or inside Jupyter Notebook. Solara handles components state and reactiveness.	16
TomoPy	Python package for CT data processing and reconstruction. All sinogram pre-processing and reconstruction operations in the pipeline of Alrecon (Figure 1) are implemented using TomoPy.	11
ASTRA	Toolbox for high-performance tomographic reconstruction methods on graphic processing units (GPUs). The ASTRA toolbox is accessed through its TomoPy integration.	13,24
DXChange	Handles data Input/Output following the Scientific Data Exchange schema for HDF5 raw and metadata storage.	23
numpy	Core Python tool for N-dimensional arrays.	25
napari	Pure Python 3D image interactive viewer. Can be used to visualize the sinogram and reconstructed 3D images.	26
ImageJ	Multi-purpose image processing package. At SESAME BEATS, the Fiji distribution of ImageJ2 is used, which bundles several plugins for scientific image analysis.	27
gspread	Python API for Google Sheets.	
pandas	Used for reconstruction parameter logging.	28
Slurm	Workload manager used for reconstruction job scheduling on HPC.	20

The CT reconstruction card (Figure 2-D) allows to launch and inspect the results of tomographic reconstruction processes. CPU and GPU implementations of several algorithms available in TomoPy and ASTRA can be selected. Conversion from floating-point to integer values, and a circular mask filter can be activated before writing the reconstructed dataset to disk. A plot of the intensity histogram of the reconstructed volume is used to select the optimal data range for representation as integers. Once the reconstruction settings are validated, a button allows the user to submit the reconstruction job on a centralized HPC cluster by performing the following:

1. A Slurm batch script file is written with reconstruction parameters taken from the active Alrecon session.
2. The batch script is submitted to Slurm on a dedicated HPC node.
3. The reconstruction parameters are logged to the master Google spreadsheet of the active beamtime.

Alrecon's module for optimization of the reconstruction COR is shown in Figure 3. Two methods are available for estimating the COR automatically[11,29]. The user can verify and optimize the choice of COR interactively, by inspecting a single slice of the 3D volume reconstructed with a range of COR values.

The module for the correction of stripe artifacts from the sinogram is shown in Figure 4A. Four techniques implemented in TomoPy can be used to remove large stripes, fluctuating or unresponsive stripes caused by dead detector pixels, partial stripes, or a combination of methods[1]. Figure 4B shows Alrecon's HPC settings card, allowing expert users to modify parameters of the Slurm job scheduling and control.

Use cases

This section demonstrates the functionalities described in the previous paragraphs, presenting results obtained at the ID10- beamline of SESAME using Alrecon. A Python environment for CT reconstruction with Alrecon was set up on a beamline workstation equipped with 36-core Intel Xeon W-2295CPU @ 3.00GHz, 2× NVIDIA RTX A5000 GPUs, and 512 GB of RAM memory, following the installation instructions included in the home page of the Alrecon GitHub repository. The workstation is connected to the centralized GPFS storage facility through a 10 Gbps switch.

Two cases of CT reconstruction performed with Alrecon are presented in Figure 5. Anatomical sections through the body of a dead wasp are shown. Figure 5A-C show the result of a standard, absorption-contrast image, while in Figure 5D-F the sample phase shift was reconstructed. SXCT scan and reconstruction settings are presented in Table 2. Reconstructions were performed on one of the GPU nodes of the beamline HPC cluster (2× Intel Xeon Gold 5220R @ 2.20GHz with 24 cores each, 576 GB RAM, Nvidia A100 40GB GPU), and can be reproduced downloading a repository of the raw data and the full metadata archive of the SXCT scan[30] (see the download link in the Data availability section). The online repository also contains a 3D video rendering of the reconstructed sample.

Figure 6 shows the result after optimizing the reconstruction COR for a SXCT scan of an archaeological incisive tooth

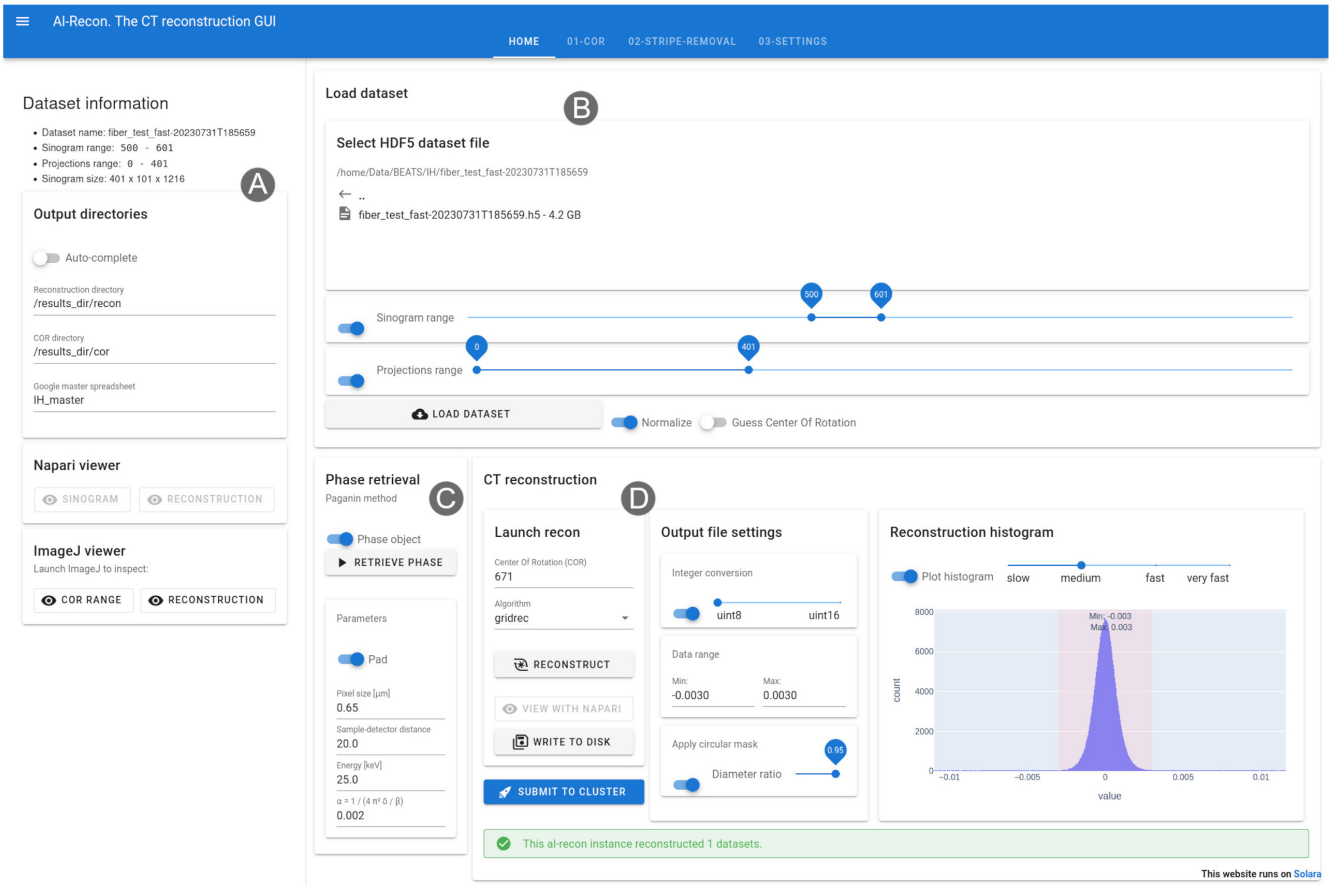


Figure 2. Home page of the Alrecon web application. (**A**) Sidebar displaying dataset information and locations for output files and logs. ImageJ and napari can be launched from the sidebar. (**B**) File load card. (**C**) Phase retrieval card. (**D**) CT reconstruction card.

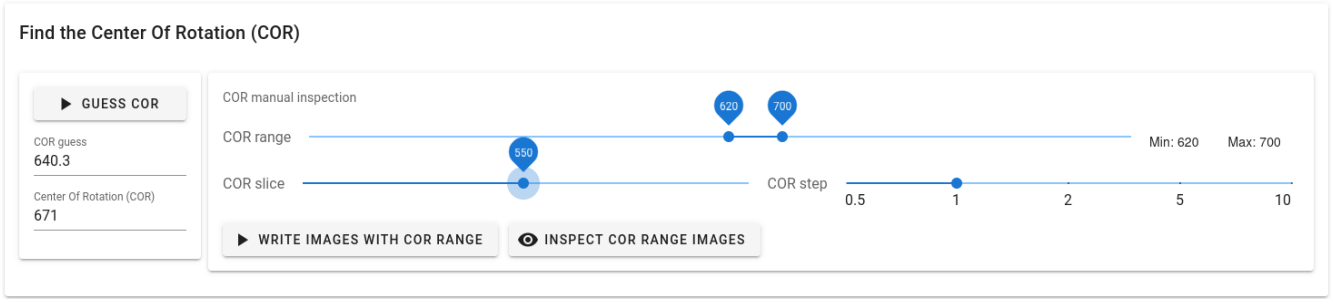


Figure 3. Center Of Rotation (COR) module. The page allows the user to find interactively the optimal COR parameter before CT reconstruction.

sample. The detail of one axial slice through the tooth is shown, after reconstruction with different COR values. The reconstruction was performed with the Gridrec algorithm implementation of TomoPy[11,31]. Through the Alrecon application it is possible to reconstruct a slice with different COR values and identify the optimal COR by inspecting with ImageJ the result of the operation.

Figure 7A shows the results of CT reconstruction of a high X-ray energy scan of a sample of Nb_3Sn (Niobium-Tin) superconducting wire with a Tantalum barrier[32]. Without stripe artifact correction, the reconstructed image is affected by ring artifacts. In Figure 7B, a pre-processing step was applied to the normalized sinogram in order to remove full and partial stripe artifacts. Stripe removal was performed using Nghia Vo's

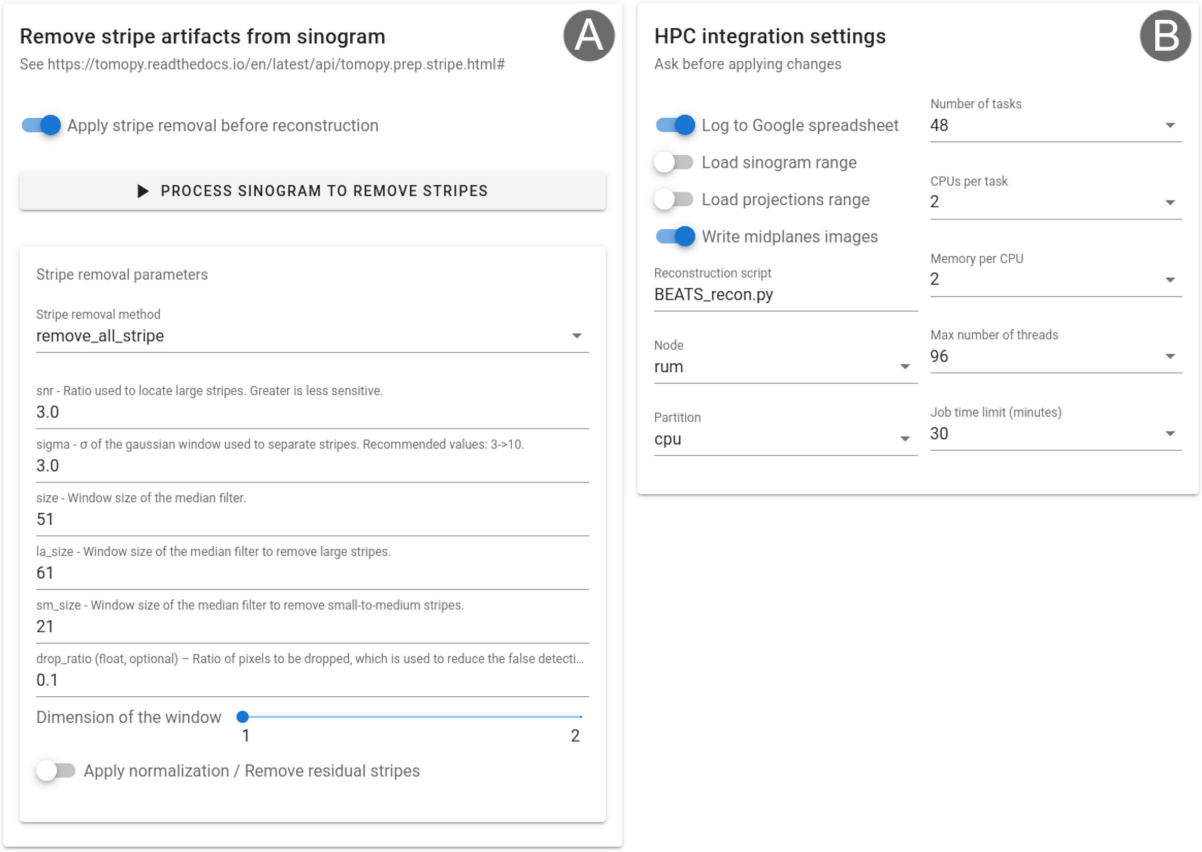


Figure 4. (**A**) Stripe removal module for the correction of stripe artifacts from the sinogram. (**B**) Settings card for HPC integration, allowing to customize the behavior of Alrecon and Slurm job files written by the application.

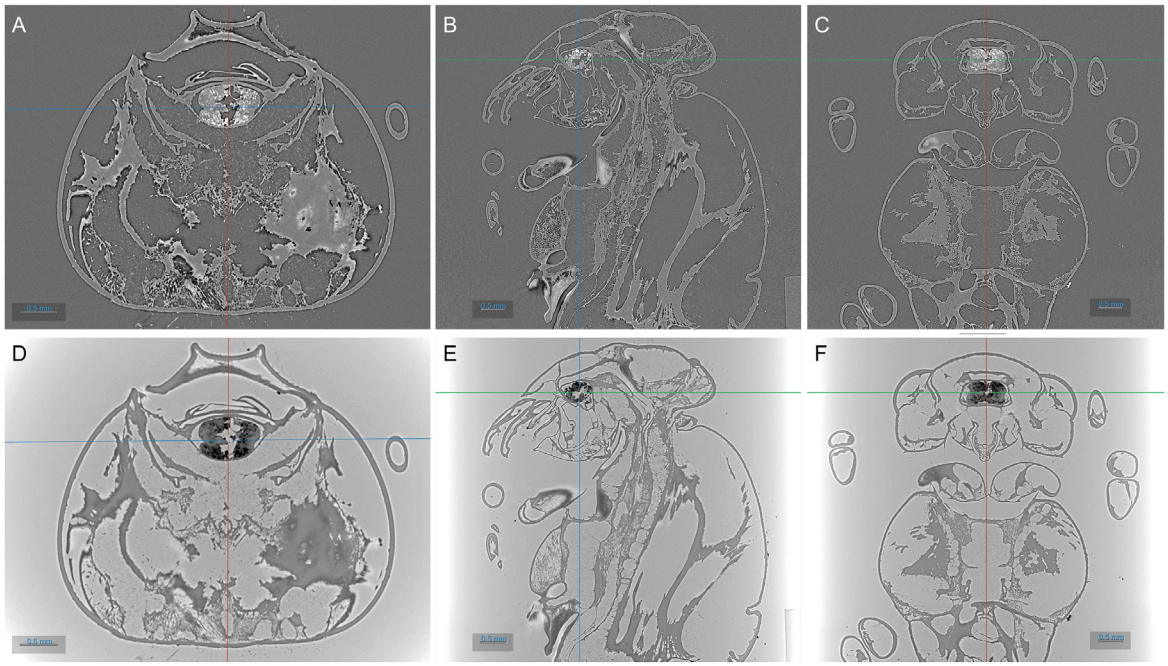


Figure 5. 3D reconstruction of SXCT scan of a wasp performed at the ID10- beamline of SESAME. Top: absorption-contrast reconstruction. Bottom: phase-contrast reconstruction. From left to right: transverse, sagittal, and coronal planes through the insect's body. Images were obtained with ORS Dragonfly.

Table 2. SXCT scan and reconstruction settings for the wasp sample of Figure 5. The insect was dead when collected for scanning.

Scan settings	
X-ray beam modality	Filtered white beam
Energy (mean)	25 keV
N. Projections	2001
Exposure time	0.02 s
Scan modality	Continuous rotation - 360° rotation
Detector	PCO edge 5.5 sCMOS; 2560 × 2160 image format
Pixel size	3.1 µm
Reconstruction settings	
Center Of Rotation	1301
Stripe removal method	none
Algorithm	Filtered Back Projection (CUDA ASTRA implementation)
Filter	Ram-Lak
Output file format	8-bit unsigned integer
Floating point representation range	Absorption contrast: [-0.0015, 0.0015] Phase contrast: [-0.0006, 0.004]
Reconstructed image size	2560 × 2560 × 2160
Phase retrieval settings	
Algorithm	Paganin *et al.*[2] (TomoPy implementation)
Sample to detector distance	115 mm
α	0.0002

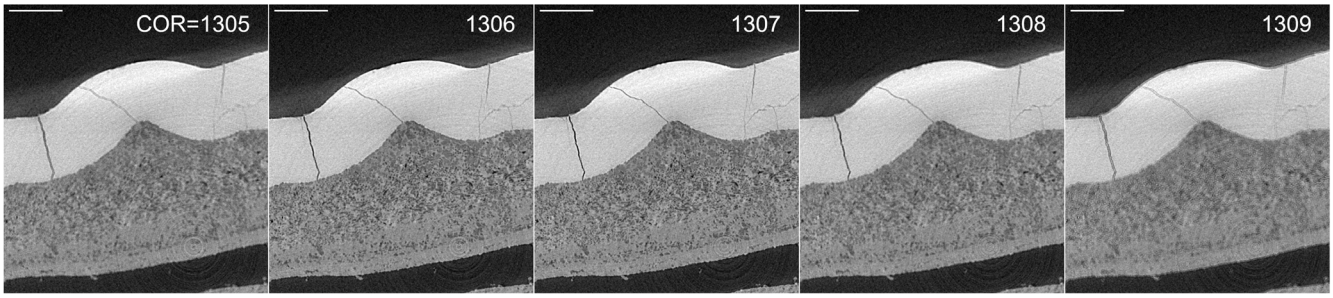


Figure 6. Slice (detail) of archaeological tooth scan reconstructed with different Center Of Rotation (COR). The image in the center (COR=1307) shows the better contrast and resolution of small features. After visual inspection of one slice, the whole dataset is reconstructed with COR=1307. Scale bar on the top-left of each image: 0.5 mm.

approach[1] using the *remove_stripe_based_sorting* method implemented in TomoPy. Both reconstructions were performed with TomoPy's implementation of the *gridrec* algorithm[31]. All steps were performed in Alrecon.

Discussion

We have presented Alrecon, an open-source, pure Python web application that provides a modern and reactive graphical user interface for the reconstruction of synchrotron X-ray computed tomography scans using TomoPy. Alrecon is designed with a general beamline user in mind, and its use does not require programming skills. At the same time, Alrecon provides unique features that can benefit not only general users, but also experienced beamline staff. These include:

- Control over the sinogram portion to be loaded and processed

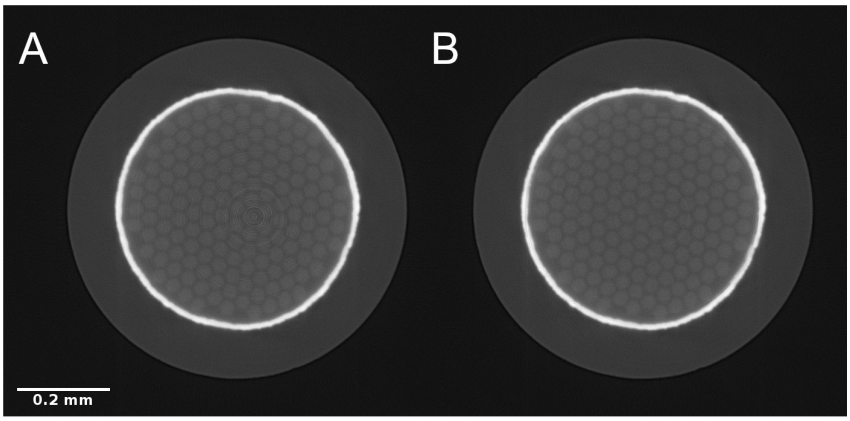


Figure 7. Reconstructed SXCT slice (detail) through a Nb_3Sn superconducting wire without (**A**) and with (**B**) stripe artifact correction pre-processing step. Scale bar: 0.2 mm.

- Module for the optimization of the tomographic center of rotation for CT reconstruction
- Module for the correction of stripe artifacts from the sinogram
- Different CT reconstruction algorithms, including iterative methods
- GPU-accelerated CT reconstruction using TomoPy and ASTRA
- Output controls for the correction of the reconstructed volume data range and integer conversion
- Possibility to schedule reconstruction jobs on HPC cluster
- Automatic logging of reconstruction jobs to Google spreadsheet

Advantages of Alrecon

The possibility to render multidimensional images directly within a Python notebook using Jupyter Widgets has motivated the development of tools similar to Alrecon such as TomoPyUI: a Jupyter-based TomoPy reconstruction graphical user interface[33]. Alrecon profits from the advantages provided by Python and ipywidgets, but uses Solara to deploy a web application with a clean, intuitive and user-friendly layout.

The use of Solara which, in turn, is based on React, makes Alrecon reactive and trustworthy, while keeping the source code extremely simple. Common beamline users can launch and control CT reconstruction tasks from a simple web page, focusing exclusively on the optimization of the reconstruction parameters. While they do so, Alrecon takes care of scheduling jobs on the beamline HPC facility and keeps an automatic online log of the submitted jobs.

The Alrecon user interface is designed with the specific goal to enable seamless inspection of both raw sinogram data, and reconstructed 3D images, immediately after the completion of SXCT scans. The possibility to attempt CT reconstruction of a dataset directly after data has been captured can be crucial to optimize experimental conditions and data collection parameters. Often, settings such as the X-ray beam energy, and the detector position or configuration, influence the quality and contrast of the SXCT image, determining the possibility to visualize and segment features of interest. Visualizing CT reconstructions immediately after the scan is the best way to obtain feedback on the quality of the experiment and allows quick identification of problems.

Alrecon exploits two wide-spread image visualization and analysis tools: napari and ImageJ. The pure python implementation of the Alrecon reconstruction environment allows the use of napari for 3D volume data visualization. Sinograms and reconstructed datasets living in multidimensional numpy arrays can be displayed by napari within Python, removing intermediate write-to-disk operations. At the same time, a launcher allows open results written to disk using ImageJ, which provides the user an extensive set of image analysis capabilities.

The simplicity of the source code, together with the modularity provided by Solara, makes Alrecon easily customizable. We have developed a HPC settings card that can be used to customize the Slurm job files written by Alrecon (Figure 4B). We expect this to help different laboratories in the adoption and set-up of Alrecon for use on their HPC facility. Other examples of possible customization include the use of beamline-specific input/output methods, depending on the HDF5 file architecture in use, and on the desired file format for the reconstructed datasets, or the use of Alrecon with CT reconstruction software other than TomoPy. Among the future developments that we envision, is the possibility to perform CT reconstructions using Tomocupy, a python package specifically developed for efficient GPU reconstruction implementing both 32-bit and 16-bit arithmetics[12].

Finally, the Alrecon Solara app can be run within Jupyter (e.g. as a Jupyter Notebook) or as a standalone server. The Alrecon reconstruction environment can be set up on a local workstation, or on a server accessing computing resources on demand. The possibility to process subsets of collected sinograms allows Alrecon to run on a machine with minimal computing resources.

Future developments

Future releases of Alrecon will extend the application versatility in dealing with diverse SCXT data acquisition and reconstruction modalities. Tools for the scan field of view extension via sinogram stitching and for custom normalization of the detector flat-field response are currently being implemented. Another important release with a focus on easy deployment of Alrecon will provide instructions on how to deploy and publish a container version of the application using Docker[34], and allow users to access the ImageJ functionalities through a browser-based platform such as ImJoy[35]. Finally, the integration of real-time reconstruction tools such as Recast3D[36] and TomoStream[37] into Alrecon would extend the field of application of the web app to time-resolved SXCT experiments.

Data availability

Underlying data

Zenodo: Underlying data for 'Alrecon: computed tomography reconstruction web application based on Solara', 'Synchrotron X-ray Computed Tomography scan of a wasp',

https://doi.org/10.5281/zenodo.10075277[30]

This project contains the following underlying data:

- bee_yazeed-20231001T170032.h5. (SXCT scan of a wasp performed at the ID10-BEATS beamline of SESAME. The HDF5 file contains raw data and metadata following the Scientific Data Exchange standard.)
- SESAME_wasp_yazeed.avi. (3D video rendering of phase-contrast CT reconstruction of bee_yazeed-20231001T170032. The dataset was reconstructed using Alrecon. The video was created using ORS Dragonfly.)

Data are available under the terms of the Creative Commons Attribution 4.0 International license (CC-BY 4.0)

Software availability

Source code available from: https://github.com/gianthk/alrecon/tree/master

Archived source code available from: https://zenodo.org/records/10535212[33]

License: MIT License.

Acknowledgements

The authors acknowledge Dr Alexander Rack from the Structure of Materials Group of the European Synchrotron Radiation Facility, Dr. Christian Barth, Dr. Tommaso Bagni, and Prof. Carmine Senatore from the University of Geneva for the contribution of a superconducting wire sample. We also acknowledge Dr Anis Fatima and Dr Kirsi O. Lorentz of the Cyprus Institute for the contribution of archaeological samples and for the reuse of a scan of archaeological teeth for this work.

References

1. Vo NT, Atwood RC, Drakopoulos M: **Superior techniques for eliminating ring artifacts in X-ray micro-tomography.** *Opt Express.* 2018; **26**(22): 28396–412.
PubMed Abstract | Publisher Full Text
2. Paganin D, Mayo SC, Gureyev TE, *et al.*: **Simultaneous phase and amplitude extraction from a single defocused image of a homogeneous object.** *J Microsc.* 2002; **206**(Pt 1): 33–40.
PubMed Abstract | Publisher Full Text
3. Cloetens P, Barrett R, Baruchel J, *et al.*: **Phase objects in synchrotron radiation hard x-ray imaging.** *J Phys Appl Phys.* 1996; **29**(1): 133–46.
Publisher Full Text
4. García-Moreno F, Kamm PH, Neu TR, *et al.*: **Tomoscopy: Time-Resolved Tomography for Dynamic Processes in Materials.** *Adv Mater.* 2021; **33**(45): e2104659.
PubMed Abstract | Publisher Full Text
5. Schröder D, Bender CL, Osenberg M, *et al.*: **Visualizing current-dependent morphology and distribution of discharge products in sodium-oxygen battery cathodes.** *Sci Rep.* 2016; **6**(1): 24288.
PubMed Abstract | Publisher Full Text | Free Full Text
6. Gondek Ł, Selvaraj NB, Czub J, *et al.*: **Imaging of an operating $LaNi_{4.8}Al_{0.2}$-based hydrogen storage container.** *Int J Hydrog Energy.* 2011; **36**(16): 9751–7.
Publisher Full Text
7. Madi K, Staines KA, Bay BK, *et al.*: **In situ characterization of nanoscale strains in loaded whole joints via synchrotron X-ray tomography.** *Nat Biomed Eng.* 2020; **4**(3): 343–54.
PubMed Abstract | Publisher Full Text | Free Full Text
8. Le Cann S, Tudisco E, Perdikouri C, *et al.*: **Characterization of the bone-metal implant interface by Digital Volume Correlation of *in-situ* loading using neutron tomography.** *J Mech Behav Biomed Mater.* 2017; **75**: 271–8.
PubMed Abstract | Publisher Full Text
9. Baker DR, Brun F, O'Shaughnessy C, *et al.*: **A four-dimensional X-ray tomographic microscopy study of bubble growth in basaltic foam.** *Nat Commun.* 2012; **3**(1): 1135.
PubMed Abstract | Publisher Full Text
10. Andrew M, Menke H, Blunt MJ, *et al.*: **The imaging of dynamic multiphase fluid flow using synchrotron-based X-ray microtomography at reservoir conditions.** *Transp Porous Med.* 2015; **110**: 1–24.
Publisher Full Text
11. Gürsoy D, De Carlo F, Xiao X, *et al.*: **TomoPy: a framework for the analysis of synchrotron tomographic data.** *J Synchrotron Radiat.* 2014; **21**(Pt 5): 1188–93.
PubMed Abstract | Publisher Full Text | Free Full Text
12. Nikitin V: ***TomocuPy* - efficient GPU-based tomographic reconstruction with asynchronous data processing.** *J Synchrotron Radiat.* 2023; **30**(Pt 1): 179–91.
PubMed Abstract | Publisher Full Text | Free Full Text
13. van Aarle W, Palenstijn WJ, Cant J, *et al.*: **Fast and flexible X-ray tomography using the ASTRA toolbox.** *Opt Express.* 2016; **24**(22): 25129–47.
PubMed Abstract | Publisher Full Text
14. Kazantsev D, Wadeson N, Basham M: **High performance Savu software for fast 3D model-based iterative reconstruction of large data at Diamond Light Source.** *SoftwareX.* 2022; **19**: 101157.
Publisher Full Text
15. Paleo P, Payno H, Vigano N, *et al.*: **Nabu 2021.2.0.** *Zenodo.* 2022.
Publisher Full Text

16. **Solara documentation.** [cited 2023 Nov 22].
Reference Source

17. Faragó T, Gasilov S, Emslie I, *et al.*: ***Tofu*: a fast, versatile and user-friendly image processing toolkit for computed tomography.** *J Synchrotron Radiat.* 2022; **29**(Pt 3): 916–27.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

18. Brun F, Massimi L, Fratini M, *et al.*: **SYRMEP Tomo Project: a graphical user interface for customizing CT reconstruction workflows.** *Adv Struct Chem Imaging.* 2017; **3**(1): 4.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

19. Iori G, Matalgah S, Chrysostomou C, *et al.*: **Data Acquisition and Analysis at the X-ray Computed Tomography Beamline of SESAME.** In: *2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. 2021; 134–9.
Publisher Full Text

20. Yoo AB, Jette MA, Grondona M: **SLURM: Simple Linux Utility for Resource Management.** In: Feitelson D, Rudolph L, Schwiegelshohn U, editors. *Job Scheduling Strategies for Parallel Processing*. Berlin, Heidelberg: Springer; (*Lecture Notes in Computer Science*), 2003; 44–60.
Publisher Full Text

21. Iori G, Foudeh I, Alzu'bi M, *et al.*: **gianthk/alrecon: Open Research Europe submission (v1.0.0).** *Zenodo.* 2024.
Publisher Full Text

22. Alzubi M, Abbadi A, Abdellatief M, *et al.*: **Experimental Data Collection Standards at SESAME Synchrotron.** In: *JACOW Publishing*. Geneva, Switzerland, 2023; 116–20.
Publisher Full Text

23. De Carlo F, Gürsoy D, Marone F, *et al.*: **Scientific data exchange: a schema for HDF5-based storage of raw and analyzed data.** *J Synchrotron Radiat.* 2014; **21**(Pt 6): 1224–30.
PubMed Abstract | **Publisher Full Text**

24. Pelt DM, Gürsoy D, Palenstijn WJ, *et al.*: **Integration of TomoPy and the ASTRA toolbox for advanced processing and reconstruction of tomographic synchrotron data.** *J Synchrotron Radiat.* 2016; **23**(Pt 3): 842–9.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

25. Harris CR, Millman KJ, van der Walt SJ, *et al.*: **Array programming with NumPy.** *Nature.* 2020; **585**(7825): 357–62.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

26. Ahlers J, Althviz Moré D, Amsalem O, *et al.*: **napari: a multi-dimensional image viewer for Python.** *Zenodo.* 2023; [cited 2023 Aug 18].
Publisher Full Text

27. Schindelin J, Arganda-Carreras I, Frise E, *et al.*: **Fiji: an open-source platform for biological-image analysis.** *Nat Methods.* 2012; **9**(7): 676–82.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

28. McKinney W: **Data structures for statistical computing in python.** In: *Proceedings of the 9th Python in Science Conference*. Austin, TX, 2010; 51–6.
Publisher Full Text

29. Vo NT, Drakopoulos M, Atwood RC, *et al.*: **Reliable method for calculating the center of rotation in parallel-beam tomography.** *Opt Express.* 2014; **22**(16): 19078–86.
PubMed Abstract | **Publisher Full Text**

30. Iori G: **Synchrotron X-ray Computed Tomography scan of a wasp.** *Zenodo.* [Data set], 2023.
http://www.doi.org/10.5281/zenodo.10075277

31. Marone F, Stampanoni M: **Regridding reconstruction algorithm for real-time tomographic imaging.** *J Synchrotron Radiat.* 2012; **19**(Pt 6): 1029–37.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

32. Bagni T, Bovone G, Rack A, *et al.*: **Machine learning applied to X-ray tomography as a new tool to analyze the voids in RRP Nb_3Sn wires.** *Sci Rep.* 2021; **11**(1): 7767.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

33. **TomoPyUI — tomopyui.** [cited 2023 Nov 3].
https://tomopyui.readthedocs.io/en/stable/

34. Merkel D: **Docker: Lightweight linux containers for consistent development and deployment.** *Linux Journal.* 2014; **2014**(239): 2.
Reference Source

35. Ouyang W, Mueller F, Hjelmare M, *et al.:* **ImJoy: an open-source computational platform for the deep learning era.** *Nat Methods.* 2019; **16**(12): 1199–1200.
PubMed Abstract | **Publisher Full Text**

36. Buurlage JW, Kohr H, Palenstijn WJ, *et al.:* **Real-time quasi-3D tomographic reconstruction.** *Meas Sci Technol.* 2018; **29**(6): 064005.
Publisher Full Text

37. Nikitin V, Tekawade A, Duchkov A, *et al.:* **Real-time streaming tomographic reconstruction with on-demand data capturing and 3D zooming to regions of interest.** *J Synchrotron Rad.* 2022; **29**(Pt 3): 816–828.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Open Peer Review

Current Peer Review Status: ✔ ✔ ✔ ? ✔ ✔

Version 2

Reviewer Report 03 January 2025

https://doi.org/10.21956/openreseurope.19270.r48211

© **2025 Estrela V.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Vânia Vieira Estrela**
Federal Fluminense University, Niterói, Brazil

Firstly, commendations to the authors for presenting Alrecon, a Python-based web application GUI that reconstructs synchrotron XCT that functions as a graphical user interface for widely used tools within the community, including TomoPy and ASTRA. The application offers streamlined access to a conventional processing pipeline, encompassing data loading, pre-processing, reconstruction, and storing functionalities with compression and information recording. Scalability is achieved by executing cluster computations using the SLURM task scheduler to confer scalability to Alrecon. The user can examine the data at every processing level via external apps like ImageJ and Napari, which can start from the application.

The following remarks are more related to the whole cyber-physical system (CPS) as perceived by users like health care professionals and biomedical/computer engineers while bearing in mind the project nature, the reviewers' remarks, and the authors' replicas. SXCT, despite some caveats in nuclear engineering, is quite good and promising in terms of extremely high resolution. Organ-to-cellular and organ-to-molecule scale imaging are crucial in neurological studies, the blood-brain barrier (BBB), and connectome mapping. These tasks are in their infancy, but they can be decisive in developing future medications targeting the brain, which more than justifies SXCT. Of course, other applications exist, but this is the genesis of the questions below.

1. What about the software scalability and its ability to incorporate other pieces of code or suites from other synchrotron- and non-synchrotron facilities via cloud computing under the assumption that cybersecurity (CS) and reliable data (e.g., protection via blockchain) are robust so the interface's problems will only come from detectors' data?
2. Alrecon leads to two huge computational software challenges: image pre-processing and 3-D, ..., N-D reconstruction, where N is an integer greater than 3. For instance, super-resolution (SR) and hyperspectral imaging can be operational, considerably increasing the feature-vector dimension per voxel.
3. Can Alrecon handle and accommodate current and future number-crunching and storage-demanding algorithms adaptively? This question stems from the fact that pre-processing synchrotron images can partially correct the scanning results of an in vivo organism. The other part will be addressed by nuclear equipment specialists working with photon-

counting detectors (PCD) in place of energy-integrating detectors (EID) to enhance SXCT imaging, which will eventually affect the data pre-processing step.

4. As is known, motion can affect SXCT even if the subject is still, and again, this is a problem that can be partially corrected through software. Here, Dr. E. Pasca's comments and Alrecon's design are sources of insights. People who develop hardware and software could extend the life cycle of their creations by leaving provisions for both hardware and software scalability and room for future virtual machines that will blend in new CPSs. Times are tough, and lots of paradigms coexist. Nevertheless, obsolescence should be postponed the most because of the costs involved in the detector's end and software engineering.
5. The questions above will also probably benefit from AI (e.g., deep learning) and quantum machines to help obtain high-resolution, multi-, and hyperspectral spectral voxels adequate for more advanced non-destructive experiments, particularly in vivo.

In order to corroborate the issues mentioned above, the authors can peruse some of these references given with this report.

References

1. Hauschild AC, Martin R, Holst SC, Wienbeck J, et al.: Guideline for software life cycle in health informatics.*iScience*. 2022; **25** (12): 105534 PubMed Abstract | Publisher Full Text
2. Systems and software engineering - Life cycle management. *International Standard*. Reference Source
3. Duan X, Ding XF, Li N, Wu FX, et al.: Sparse2Noise: Low-dose synchrotron X-ray tomography without high-quality reference data.*Comput Biol Med*. 2023; **165**: 107473 PubMed Abstract | Publisher Full Text
4. Shin S, Kim MW, Jin KH, Yi KM, et al.: Deep 3D reconstruction of synchrotron X-ray computed tomography for intact lungs.*Sci Rep*. 2023; **13** (1): 1738 PubMed Abstract | Publisher Full Text
5. Brunet J, Cook AC, Walsh CL, Cranley J, et al.: Multidimensional Analysis of the Adult Human Heart in Health and Disease Using Hierarchical Phase-Contrast Tomography.*Radiology*. 2024; **312** (1): e232731 PubMed Abstract | Publisher Full Text
6. Pu,S., Song, J., Li, L: Benchtop cone-beam X-ray fluorescence CT system with high sensitivity and high spatial resolution. *IEEE*. Publisher Full Text
7. Tavakoli, C., Cuccione, E., Dumot, C., Cormode, D., Wiart, M., Elleaume, H, Brun , E: Tracking cells in the brain of small animals using synchrotron multi-spectral phase contrast imaging. *HAL Open Science*. Publisher Full Text

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Electronic and computer engineering, cyber-physical systems, Industry 5.0, multidimensional signal processing, biomedical engineering, multimedia, neuroinformatics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 26 December 2024

https://doi.org/10.21956/openreseurope.19270.r47665

© 2024 Vigorelli L. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Luisa Vigorelli
Università di Milano-Bicocca and INFN, Milano, Italy

In this paper, the authors present a new software, called Alrecon, developed for synchrotron computed tomography reconstruction. The software is realized using Solara, a tool that allow to build web apps based on pure Python language. The software is designed and created trying to build a graphical interface easy to understand and use even for those with less programming experience, separating the tools for the different reconstruction phases. Synchrotron users will be able to use this tool for tomographic reconstructions immediately after acquisitions and verify step by step the quality of the reconstruction, also thanks to the use of visualization tools such as ImageJ or Napari. In this way, the user is guided through all the work phases, having the possibility of adopting the best parameters to obtain high-quality reconstruction (centre of rotation research, ring artefacts correction). The software gives the possibility of viewing and reconstructing both absorption and phase contrast images, using different phase retrieval algorithms for the latter.
The authors provide a test database for users in order to become familiar with the software and the link for the source code. The authors report also some consideration about aspects and functionalities that can be integrated as future development of the software.
Overall, the paper is well written and easy to read and understand, I certainly recommend it for approval.

- This new tool seems very promising; in the current configuration, could it be employed for non-synchrotron CT reconstruction too? Do some code parts have to be added/modified, for example beam geometry? (just a curiosity)
- Does it make sense to add a tool for beam hardening correction in AIrecon (if not already present)?

Is the rationale for developing the new software tool clearly explained?
The authors give the rationale description in the Introduction section. Some already developed toolkits are optimized for application on stand-alone GPU workstations, however with some

drawbacks, as data dimensions, manipulation and storage. Moreover, more than one software or tool are often use for CT reconstruction, sometimes needing some expertise and resulting time consuming. Alrecon can be used by SXCT beamline staff, as well as researchers or students lacking programming experience.

Is the description of the software tool technically sound?
In the paper the software construction is described, with single procedures and the general CT reconstruction pipeline. All the parts are well described, with detailed figures. The authors do not report the architecture of the web application or provide code examples.

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
The authors describe principally the functional and graphical aspects of the software rather than the baseline code; however, this make sense as the aim of the paper is to present a new software thought for enabling users to access functionality from multiple packages. They also describe the single modules reporting some application examples, useful for a better understanding.

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
The authors provided some examples of different software functionalities, showing the potential of the tool; the results are presented with explicative images and parameters.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computed tomography, material science

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 11 December 2024

https://doi.org/10.21956/openreseurope.19270.r48204

© 2024 van Leeuwen T. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Tristan van Leeuwen
Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

The authors introduce a web-based front-end for tomographic reconstruction of Synchrotron X-ray CT data. The tool combines several steps in the typical data-processing workflow in a way that is designed to be easily accessible for beamline scientists. The functionalities include preprocessing, reconstruction, and some post processing and scheduling of compute-heavy jobs to a connected HPC cluster as well as logging of the experimental parameters and processing steps. Overall, it seems a very relevant tool that can be used to address the gap between experimentalists, who collect the data, imaging scientists, who interpret the data, and computational scientists, who develop the algorithms. As such, it is a great step forward in making such involved workflows reproducible.

Generally, the paper is well-written and serves as a good overview of the corresponding software. I don't think the paper is meant as a step-by-step tutorial but it seems to contain sufficient detail to get started. I have not tried to run the accompanying examples nor tried to install the code. so I cannot say it the steps in the paper and documentation suffice. I was also wondering if there is a way to user another cloud-based service for logging if one does not want to rely on Google. I am not an expert, but I can imagine that for some users / applications this may cause legal problems if meta-data related to their project is stored in certain geographical locations.

I have a few minor textual remarks that could help improve readability.

1. The abstract is rather verbose and contains quite a lot of technical jargon (Radiograph, Radon transform. Solara). as well as in my view unnecessarily verbose explanations (such as mentioning that the images are written to disk).

2. On page 3 a few examples of artifacts are mentioned (stripes, rings); it would be nice to have an illustration of this to help the reader.

3. On page 4, statement of need, it is mentioned that Alrecon is user-friendly; this seems a subjective statement. I understand that it is design that way, but has it's user-friendliness been evaluated? if so, how?
Similarly, the 'flawless log' seems strange; a flawed log would be useless. Perhaps replace by detailed log? Even then, what does this mean? Which details are logged?

4. On page 4 Pre-processing and reconstruction... it is mentioned that the sinogram is loaded as a 3D numpy array -- is this crucial information? If so, what is the type used? float32? Otherwise, just replace with '3D array' or 'loaded in memory'?

5. On page 10 it states 'the simplicity of the source code', which is a subjective term. I would replace it by 'the source code has been designed to facilitate customisation' or such and refer to detailed documentation of the API.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computational Imaging

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 10 December 2024

https://doi.org/10.21956/openreseurope.19270.r46389

© 2024 Cheng C. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chang-Chieh Cheng
Information Technology Service Center, National Yang Ming Chiao Tung University (Ringgold ID: 34914), Hsinchu, Taiwan

This paper introduces Alrecon, a web-based system with an intuitive graphical user interface designed to support synchrotron X-ray computed tomography (SXCT). By integrating widely used tools such as TomoPy, ASTRA, napari, and ImageJ, it offers a streamlined solution for SXCT workflows. Alrecon also incorporates DXChange and Slurm, enabling users to schedule tomographic reconstruction tasks on high-performance computing (HPC) systems with minimal programming expertise. Built with Python and released as open source, the system is highly customizable and extensible.
The paper is well-structured and presents its ideas in a logical and coherent manner. However, I have several recommendations:

1. For Figures 6 and 7, consider including enlarged views for each subimage to emphasize regions

affected by significant COR or stride artifacts.
2. I suggest that the authors provide a series of images demonstrating how napari can be used to visualize a reconstructed volume.
3. COR and ring artifacts are common issues in SXCT. Many methods for correcting these artifacts have been proposed. I recommend that the authors cite recent works, such as References 1-4, to provide a more detailed background. Additionally, it would be helpful to include an example illustrating how Alrecon could be extended to incorporate external programs for COR or ring artifact correction.

Refereces:
[Ref 1], [Ref 2], [Ref 3] and [Ref 4]

References

1. Cheng CC, Ching YT, Ko PH, Hwu Y: Correction of center of rotation and projection angle in synchrotron X-ray computed tomography.*Sci Rep*. 2018; **8** (1): 9884 PubMed Abstract | Publisher Full Text
2. Topal E, Löffler M, Zschech E: Deep Learning-based Inaccuracy Compensation in Reconstruction of High Resolution XCT Data.*Sci Rep*. 2020; **10** (1): 7682 PubMed Abstract | Publisher Full Text
3. Zhao Y, Ma C, Ji DJ, Peng Y, et al.: Structure-aware guided filtering for a ring artifact correction in synchrotron x-ray microtomography.*Appl Opt*. 2023; **62** (28): 7400-7410 PubMed Abstract | Publisher Full Text
4. Fu T, Wang Y, Zhang K, Zhang J, et al.: Deep-learning-based ring artifact correction for tomographic reconstruction.*J Synchrotron Radiat*. 2023; **30** (Pt 3): 620-626 PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Image processing, computational imaging, visualization

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 07 May 2024

https://doi.org/10.21956/openreseurope.18220.r39375

© **2024 Pasca E.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Edoardo Pasca
Science Technology Facilities Council, Swindon, UK

The authors present Alrecon an OS python web application GUI for reconstruction of synchrotron XCT. The application serves as a GUI for popular tools in the community such as tomopy. The app provides simplified access to a standard processing pipeline with data loading, pre-processing (normalisation, centre of rotation correction, ring artifact/stripe remover, phase retrieval filter), reconstruction (with tools provided by the underlying packages tomopy and ASTRA) and, saving capabilities with compression and meta data logging. Scalability is obtained by being able to run the computation on clusters via SLURM job scheduler. At each stage of the processing the user is able to inspect the data via external applications such as ImageJ and napari which can be launched from the application itself.

- Is the rationale for developing the new software tool clearly explained?

The authors introduce Alrecon is a GUI for synchrotron XCT which relies on other OS packages used by the community, such as tomopy, ASTRA, dxchange and leverages on scalable processing via SLURM.
The need for developing the UI is well explained: processing the XCT data often requires multiple software and expert users. The UI is designed to provide a simple intuitive tool for the XCT pipeline.

- Is the description of the software tool technically sound?

The authors describe the UI and the tools it provides in relative detail. They do not describe the architecture of the web application or provide code snippets rather they describe the pipeline it implements and they substantiate it with 3 with use cases.

- Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

As the presented software tool is a GUI, the authors focus the description on the functional and graphical aspects rather than the underlying code. The authors exploit 3 use cases to describe the phase retrieval, centre of rotation and ring/stripe removal procedures that are available in Alrecon. For each case, they provide detailed configuration of the procedure with links to the data utilised for one of the use cases.

- Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Examples of output of the tool are described with figures and parameters. This is considered sufficient as the aim of the publication is to discuss a GUI that allows users access functionality from other packages rather than presenting the actual functionality or technical details of the application.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computed tomography, software

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 06 May 2024

https://doi.org/10.21956/openreseurope.18220.r39380

© 2024 Nikitin V. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Viktor Nikitin
Argonne National Laboratory, Lemont, Illinois, USA

In this paper, the authors report their recently developed software package AIrecon for CT data reconstruction and visualisation. The main feature of the package is its user-friendly web interface based on Solara, which allows users to build web apps from pure Python using ipywidgets or a React-like API on top of ipywidgets. This web interface makes the package usable for experienced beamline staff, researchers, and students without programming experience. The interface allows for selecting reconstruction parameters, starting visualisation packages (ImageJ and Napari), submitting a job to an HPC cluster, and other functionality. Reconstruction methods are called from publicly available and already well-functioning TomoPy and Astra packages.
The AIrecon package is publicly available at https://github.com/gianthk/alrecon/tree/master and can be easily installed and tested. The tests can be performed on an experimental h5 dataset acquired at ID10 beamline of SESAME synchrotron. The authors provided access to these data via Zenodo.
The paper is well-structured and easy to read, and I certainly recommend it for approval. There are just a couple of comments:

1. It would be nice to mention Solara in the Introduction since not all people are familiar with it.
2. I recommend mentioning real-time reconstruction tools like Recast3D (Buurlage, et al., 2018 [Ref 1]) and TomoStream (Nikitin, et al., 2022 [Ref 2]) and commenting on whether something like that can be implemented within AIrecon. Also, in the current version, ImageJ is called from the local computer.
3. There is an ImageJ working in the browser at https://ij.imjoy.io/. Does it make sense to adapt it for AIrecon? Just a recommendation.

References

1. Buurlage J, Kohr H, Jan Palenstijn W, Joost Batenburg K: Real-time quasi-3D tomographic reconstruction. *Measurement Science and Technology*. 2018; **29** (6). Publisher Full Text
2. Nikitin V, Tekawade A, Duchkov A, Shevchenko P, et al.: Real-time streaming tomographic reconstruction with on-demand data capturing and 3D zooming to regions of interest.*J Synchrotron Radiat*. 2022; **29** (Pt 3): 816-828 PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computed tomography, software development

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.