BEATS

Work Package 7
Data analysis and management

Report on the successful adaptation of the tomography software

D7.4

June 2023

Funded by the EU's H2020
framework programme under

PROJECT DETAILS

PROJECT ACRONYM	PROJECT TITLE
BEATS	**BEAmline for Tomography at SESAME**

GRANT AGREEMENT NO:
822535

THEME

START DATE
2019

DELIVERABLE DETAILS

WORK PACKAGE ID: 7

EXPECTED DATE: 30/06/2023

WORK PACKAGE TITLE: DATA ANALYSIS AND MANAGEMENT

DELIVERABLE TITLE: REPORT ON THE SUCCESSFUL ADAPTATION OF THE TOMOGRAPHY SOFTWARE

WORK PACKAGE LEADER: CH. CHRYSOSTOMOUS /CYI

DELIVERABLE DESCRIPTION: REPORT, PUBLIC

DELIVERAVLE ID: D 7.4

PERSON RESPONSIBLE FOR THE DELIVERABLE: CH. CHRYSOSTOMOU

NATURE

☒ R - Report □ P - Prototype □ D - Demonstrator □ O - Other

DISSEMINATION LEVEL

☒ P - Public

REPORT DETAILS

VERSION: 1.0

DATE: 30.06.2023

NUMBER OF PAGES: 20

DELIVERABLE REPORT AUTHOR(S): CH. CHRYSOSTOMOU, M. ALYU'BI, G. IORI, S. MATALGAH

FOR MORE INFO PLEASE CONTACT: C. CHRYSOSTOMOU@CYI.AC.CY

STATUS

□ Template □ Draft

☒ Final ☒ Released to the EC

CONTENTS

DESCRIPTION OF THE DELIVERABLE

The adaptation of the tomography software project has seen significant success, primarily anchored on enhancements to the computing infrastructure. A significant development comprised boosting the Random Access Memory (RAM) of the acquired HPC infrastructure for more efficient data processing, while the acquisition of Dragonfly VizServer has upgraded remote data analysis and collaboration capabilities. Solutions such as the Short Term Storage (STS), the Long Term Storage (LTS), and the ICAT metadata catalogue have been deployed for efficient data management to address escalating storage needs. The IBM TS4300 Tape Library system is a noteworthy addition, a scalable solution catering to future storage requirements and ensuring secure, automated data handling.

Data acquisition for the BEATS project has been revolutionised by introducing TomoScan, a robust, Python-based software application specifically designed for tomography data collection. Its innate flexibility makes it ideally suited for the demanding data acquisition needs of the BEATS beamline. To overcome the performance bottleneck encountered with the PCO camera during step scan mode, we developed a Raspberry Pi 4-based triggering system. Utilising TTL (Transistor-Transistor Logic) triggering, this system bypasses the arming time associated with each single frame collection, enhancing the system's overall efficiency. A critical advancement has been developing an HDF5 DXFile writer based on the ZeroMQ (ZMQ) asynchronous messaging library. This writer has significantly improved data collection and processing, overcoming previous limitations and enhancing the overall performance of the TomoScan system.

The tomography software, Tomopy successfully reconstructed a full X-ray dataset, demonstrating various CPU and GPU reconstruction algorithms' effectiveness. It has been instrumental in establishing a swift and accurate image reconstruction pipeline, reducing reconstruction time considerably. The Slurm Workload Manager, integrated into the system, has significantly improved overall productivity and resource efficiency by effectively managing job scheduling, resource allocation, and balancing workload distribution. Several advanced tools, including ImageJ, Paraview, Dragonfly, 3D Slicer, PALABOS, Quanfima, Calculix, and iMorph, are made available to enrich the quality of work and foster detailed, high-quality visualisations for a wide range of applications.

In conclusion, the project has achieved its goals, successfully upgrading the computational infrastructure, establishing an efficient reconstruction pipeline, and effectively managing data storage needs, aiming to adapt tomography software successfully.

REPORT ON THE SUCCESSFUL ADAPTATION OF THE TOMOGRAPHY SOFTWARE

Upgrades to Computing Infrastructure

Several enhancements to the computing infrastructure have fundamentally underpinned the project's success. To optimise the efficiency of the reconstruction processes further, the system's Random Access Memory has undergone significant augmentation to meet the high computational needs intrinsic to these processes. Every GPU node within the system is now equipped with 512GB of RAM, representing a substantial increase from the preceding 192 GB. This RAM expansion facilitates enhanced and direct access, leading to expedited processing times. Consequently, this enhancement is predicted to substantially amplify the performance of the tomography software by reducing processing times and augmenting BEATS' capacity to manage more extensive data sets efficiently.

A Dragonfly VizServer, a dedicated 3D visualisation and analysis dedicated workstation that will enable interactive exploration of large-scale data, has been acquired alongside RAM augmentation, a cutting-edge tool conceived to refine remote data analysis post-beamtime. Serving as a dedicated visualisation server, the Dragonfly VizServer empowers remote, real-time examination and visualisation of large-scale data. This server enables team members to execute high-calibre, interactive visualisations and analyses from any geographic location, bolstering collaboration and augmenting productivity. By deploying the Dragonfly VizServer, the aim is to render the post-beamtime data analysis process more efficient and interactive, thereby hastening research outcomes. Integrating this server into the system will significantly heighten team members' productivity by reducing downtime and enabling more effective data utilisation.

SESAME Experimental Data (SED) Archiving

The storage of BEATS experimental data, especially concerning its rapidly increasing storage capacity, necessitates maintaining the GPFS storage, also known as the Short Term Storage (STS), in an orderly and functional condition. The Long Term Storage (LTS) provision has been established to undertake data archiving tasks, ensuring a cost-effective solution for safely storing the immense quantity of BEATS data. ICAT, an open-source community solution, offers a metadata catalogue alongside supporting components for efficient experimental data management. Web-based access is accommodated by ICAT, in addition to managing STS and LTS file access. The ICAT system has been integrated to oversee STS and LTS file access, assuring seamless transition between different storage media. The ICAT Data Service (IDS) is a key component in scientific data management, having been developed to store, oversee, and provide efficient access to large data files.

The IBM TS4300 Tape Library system has been chosen and deployed to cater to long-term storage needs, given the exponentially increasing volume of data. Comprising a base unit supplemented by an expansion unit, this state-of-the-art system offers a scalable infrastructure capable of adapting to future storage requirements. The 60 LTO 9 data cartridges embedded in this configuration, each a secure and reliable data storage medium, contribute to a substantial total storage capacity of 1 petabyte (PB), equivalent to 1,000 terabytes (TB). This LTS system is operated via a dedicated archiving server that ensures data preservation and availability through key tasks like data backup, restoration, and retrieval. The physical tape cartridges are stored securely within a tape library, a device specifically designed for automated data handling. This system promotes efficient data management and reduces the risk of manual handling errors, featuring barcode reading for quick tape identification and advanced inventory management for tracking stored data.

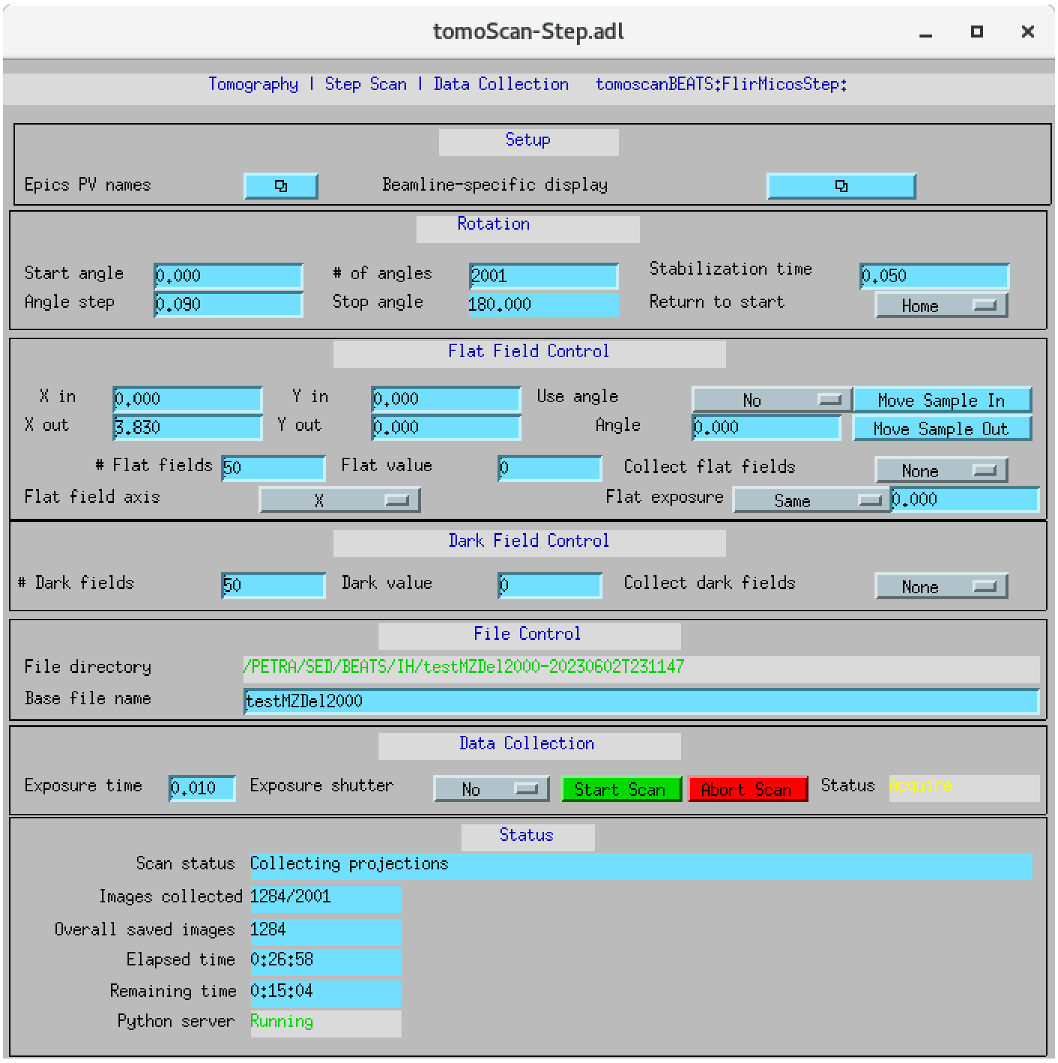
Data Acquisition of BEATS

The data acquisition procedure within the BEATS infrastructure is fundamentally anchored on the use of TomoScan, an efficient Python-based, object-oriented software application primarily conceived for tomography data collection at the Advanced Photon Source (APS). The comprehensive capabilities of TomoScan, including its inherent flexibility and robustness, make it an ideal fit for the intense data acquisition demands of BEATS.

TomoScan features an implemented base class that generalises the necessary operations for any tomography beamline. This base class serves as a backbone for the system, providing the essential functionalities required for efficient data collection. Leveraging the versatility of this base class and the object-oriented nature of TomoScan, BEATS has developed specific modules tailored to its unique needs.

For BEATS, these customised modules are built on TomoScan, enhancing the system's capabilities to accommodate data acquisition in two distinct scan modes: step scan and continuous scan. These modules enable seamless integration with two types of detectors utilised at the BEATS beamline: the FLIR and the PCO camera. FLIR (Forward Looking Infrared) and PCO (Pioneering Camera Optics) are specialized cameras used at BEATS. FLIR captures thermal or infrared radiation, which is useful for detecting heat differences. PCO cameras, known for their high resolution and speed, are ideal for high-speed and low-light imaging. Their integration into the BEATS system enhances its data acquisition capabilities. Moreover, the developed modules ensure smooth operation with other beamline components, including the sample stage, thus contributing to a robust data acquisition system.

The primary Graphical User Interface (GUI) of BEATS_TomoScan, embodies the culmination of these elements, presenting an intuitive and efficient platform for managing and overseeing the data acquisition process. The following is a depiction of the main GUI of BEATS_TomoScan:



tomoScan-Step.adl

Tomography | Step Scan | Data Collection tomoscanBEATS:FlirMicosStep:

Setup

Epics PV names

Beamline-specific display

Rotation

Start angle 0.000 # of angles 2001 Stabilization time 0.050

Angle step 0.090 Stop angle 180.000 Return to start Home

Flat Field Control

X in 0.000 Y in 0.000 Use angle No Move Sample In

X out 3.830 Y out 0.000 Angle 0.000 Move Sample Out

Flat fields 50 Flat value 0 Collect flat fields None

Flat field axis X Flat exposure Same 0.000

Dark Field Control

Dark fields 50 Dark value 0 Collect dark fields None

File Control

File directory /PETRA/SED/BEATS/IH/testMZDel2000-20230602T231147

Base file name testMZDel2000

Data Collection

Exposure time 0.010 Exposure shutter No Start Scan Abort Scan Status Acquire

Status

Scan status Collecting projections

Images collected 1284/2001

Overall saved images 1284

Elapsed time 0:26:58

Remaining time 0:15:04

Python server Running

TomoScan, a vital tool within the BEATS control system, accommodates the inclusion of metadata pertinent to the experiment in progress. This metadata encompasses various information, effectively providing a comprehensive experiment overview. The details incorporated in the metadata comprise the following key elements:

Principal Investigator: This is the primary individual responsible for the design, execution, and management of the experiment. The name of the Principal Investigator lends accountability to the research.

Sample Description: A thorough description of the sample involved in the experiment is included. This provides a clear understanding of the material under investigation, its properties, and its relevance to the study.

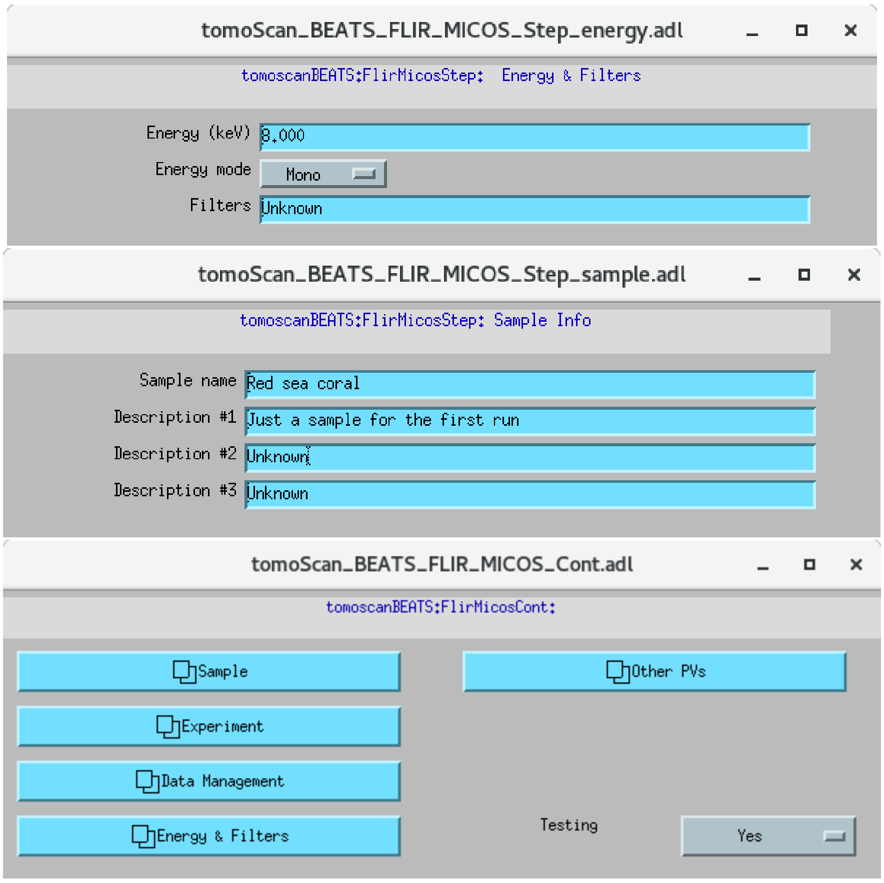
Beam Energy: The energy level of the beam utilised during the experiment is documented. This is a critical parameter that determines the experiment's overall results.

Proposal Number: The specific number associated with the proposal for the experiment is recorded. This aids in traceability and serves as a reference for future studies or audits.

Beam Mode: The mode of the beam used in the experiment is also documented. This information is crucial in understanding the beam's operational status during the experiment.

Additional Information: Various other information relevant to the experimental setup and execution are also included in the metadata.

Following are a few illustrative examples represented by screenshots, providing a visual overview of the incorporation of metadata within Tomoscan:



tomoScan_BEATS_FLIR_MICOS_Step_energy.adl

tomoscanBEATS:FlirMicosStep: Energy & Filters

Energy (keV) 8.000

Energy mode Mono

Filters Unknown

tomoScan_BEATS_FLIR_MICOS_Step_sample.adl

tomoscanBEATS:FlirMicosStep: Sample Info

Sample name Red sea coral

Description #1 Just a sample for the first run

Description #2 Unknown

Description #3 Unknown

tomoScan_BEATS_FLIR_MICOS_Cont.adl

tomoscanBEATS:FlirMicosCont:

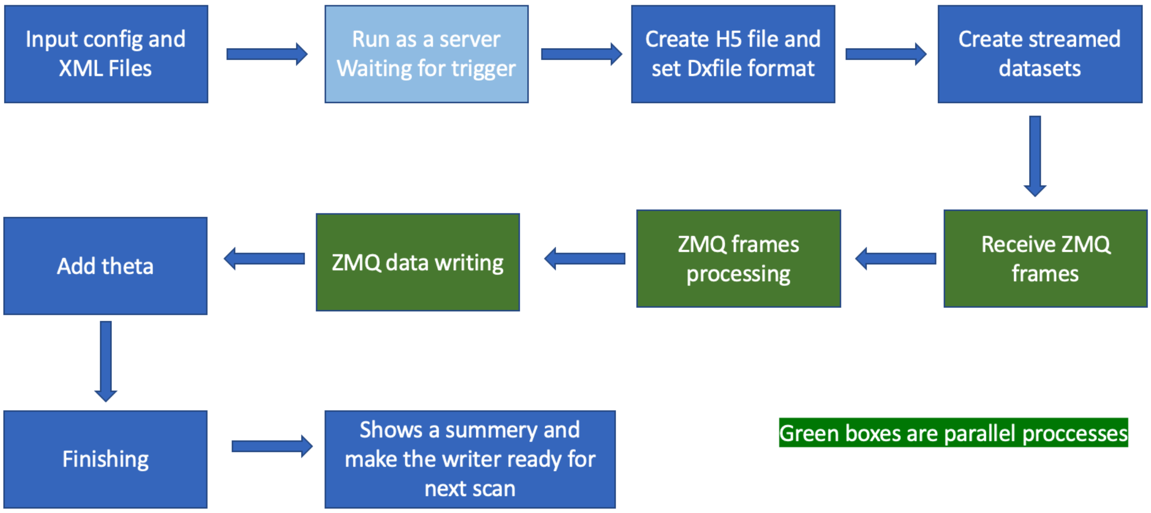
Testing Yes

The TomoScan system, the beamline's major tool for tomographic data collection and processing, is designed to operate exclusively with the EPICS control system hosted on the Linux operating system. The prerequisite for this functioning is that the EPICS support for the beamline components, including the detector and the sample stage, should be preconfigured and fully operational. It is noteworthy that TomoScan does not undertake the task of writing experimental files. Instead, it leverages the EPICS driver of the detector to accomplish this. Such reliance on the EPICS driver provides a straightforward and efficient implementation strategy. However, it also carries certain drawbacks, predominantly regarding time consumption. The dataset has to be initially collected on the EPICS driver server at the beamline before being transferred to the main storage at the data centre. This process can be particularly time-consuming when the data must be copied from a Windows operating system, such as the one used by the PCO driver server, to the GPFS-based main storage. The file copying performance in such a scenario is limited to protocol speed, regardless of network connectivity.

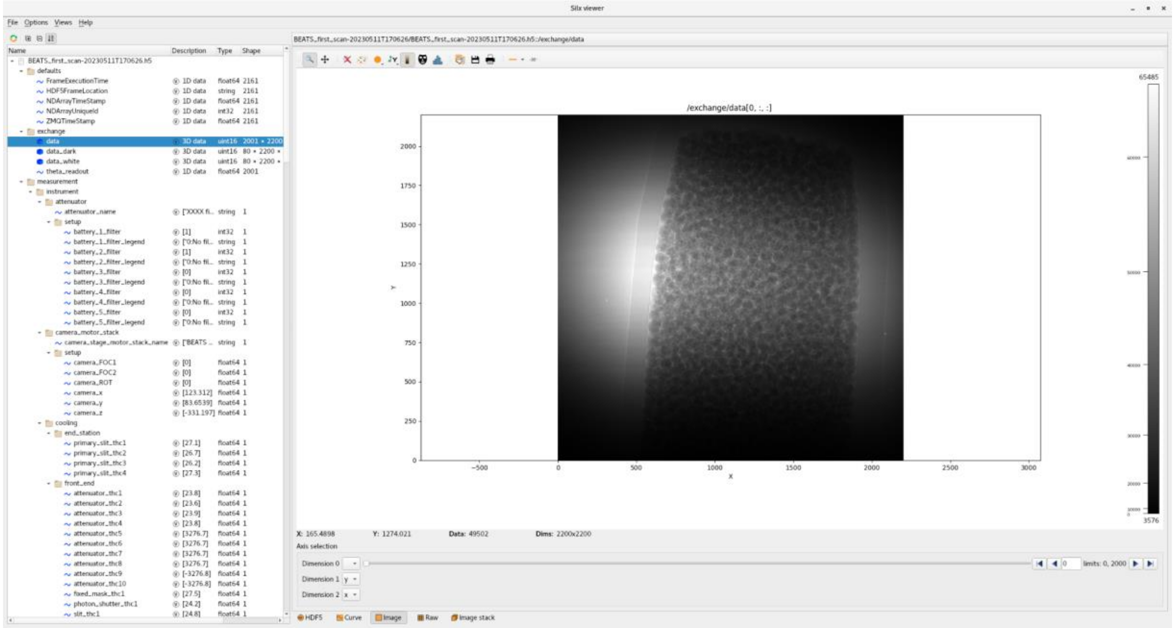
Furthermore, since the HDF5 DXFile file format has been selected for this beamline in order to have all information compiled in a single file, all systems - including the EPICS camera driver, motion driver, and TomoScan itself - must be installed and configured on a single Linux server. This is necessary as all the information collected from different Experimental Physics and Industrial Control System Input/Output (EPICS IOCs) should be accessible to the server that writes the file. However, this presents a challenge since BEATS' current setup includes the PCO detector operating on a Windows-based system, while TomoScan is Linux-based. A comprehensive problem description was shared with project partners to address these issues, culminating in the decision to develop an HDF5 DXFile writer. This writer can record all the information inside the file, irrespective of the operating system hosting the EPICS detector driver.

The design of the HDF5 DXFile writer is based on the publisher-subscriber sockets connection type of the ZeroMQ (ZMQ) asynchronous messaging library. In this setup, the camera driver functions as a publisher, streaming the camera frames and any other information from the camera driver. Conversely, the writer acts as a subscriber, receiving all data, including the camera frames and additional data, such as the theta readout from the motion EPICS IOC. It then consolidates all the received data into the HDF5 DXFile. This approach ensures a robust and efficient data collection and processing pipeline, overcoming any limitations and enhancing the overall performance of the TomoScan system. The general block diagram below provides an overview of the writer's operation:

- **BEATSH5Writer:** Components


Input config and XML Files
Run as a server Waiting for trigger
Create H5 file and set Dxfile format
Create streamed datasets
Receive ZMQ frames
ZMQ frames processing
ZMQ data writing
Add theta
Finishing
Shows a summery and make the writer ready for next scan
Green boxes are parallel proccesses

A sample file generated from the writer is shown below:



BEATS_first_scan-20230511T170626/BEATS_first_scan-20230511T170626.h5::/exchange/data

Name	Description	Type	Shape
BEATS_first_scan-20230511T170626.h5			
defaults			
FrameExecutionTime	1D data	float64	2161
HDF5FrameLocation	1D data	string	2161
NDArrayTimeStamp	1D data	float64	2161
NDArrayUniqueId	1D data	int32	2161
ZMQTimeStamp	1D data	float64	2161
exchange			
data	3D data	uint16	2001 × 2200
data_dark	3D data	uint16	80 × 2200 ×
data_white	3D data	uint16	80 × 2200 ×
theta_readout	1D data	float64	2001
measurement			
instrument			
attenuator			
attenuator_name	["XXXX fi...	string	1
setup			
battery_1_filter	[1]	int32	1
battery_1_filter_legend	["0:No fil...	string	1
battery_2_filter	[1]	int32	1
battery_2_filter_legend	["0:No fil...	string	1
battery_3_filter	[0]	int32	1
battery_3_filter_legend	["0:No fil...	string	1
battery_4_filter	[0]	int32	1
battery_4_filter_legend	["0:No fil...	string	1
battery_5_filter	[0]	int32	1
battery_5_filter_legend	["0:No fil...	string	1
camera_motor_stack			
camera_stage_motor_stack_name	["BEATS ...	string	1
setup			
camera_FOC1	[0]	float64	1
camera_FOC2	[0]	float64	1
camera_ROT	[0]	float64	1
camera_x	[123.312]	float64	1
camera_y	[83.6539]	float64	1
camera_z	[-331.197]	float64	1
cooling			
end_station			
primary_slit_thc1	[27.1]	float64	1
primary_slit_thc2	[26.7]	float64	1
primary_slit_thc3	[26.2]	float64	1
primary_slit_thc4	[27.3]	float64	1
front_end			
attenuator_thc1	[23.8]	float64	1
attenuator_thc2	[23.6]	float64	1
attenuator_thc3	[23.9]	float64	1
attenuator_thc4	[23.8]	float64	1
attenuator_thc5	[3276.7]	float64	1
attenuator_thc6	[3276.7]	float64	1
attenuator_thc7	[3276.7]	float64	1
attenuator_thc8	[3276.7]	float64	1
attenuator_thc9	[-3276.8]	float64	1
attenuator_thc10	[-3276.8]	float64	1
fixed_mask_thc1	[27.5]	float64	1
photon_shutter_thc1	[24.2]	float64	1
slit_thc1	[24.8]	float64	1

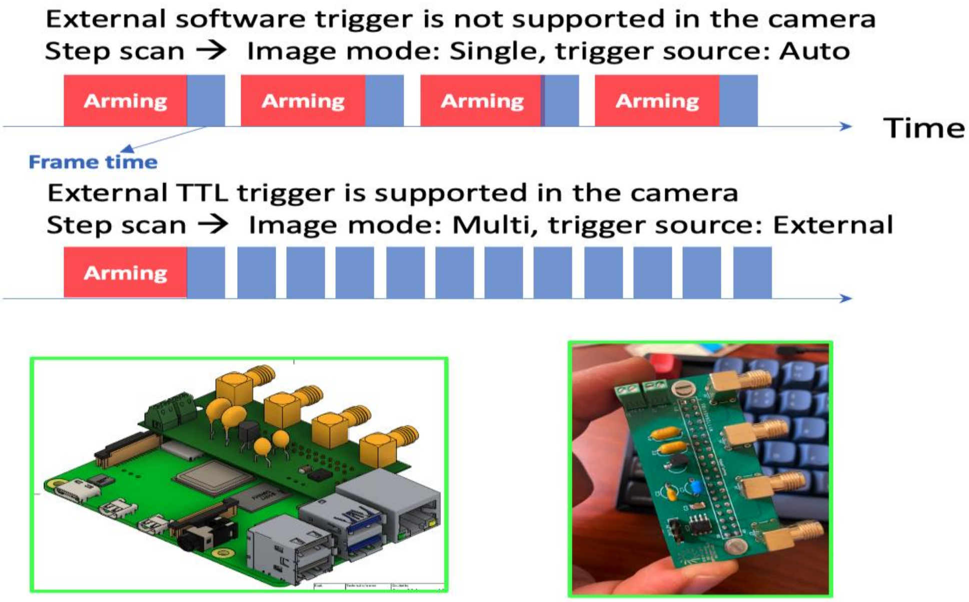
X: 165.4898 Y: 1274.021 Data: 49502 Dims: 2200x2200

Axis selection

Dimension 0 | 0 | limits: 0, 2000
Dimension 1 y
Dimension 2 x

During the system commissioning phase, we encountered a performance issue with the PCO camera operating in step scan mode. The issue concerned the camera's arming time, approximately 1.1 seconds for every frame collected. This significantly impacted the overall efficiency of the system. We identified a solution to rectify this performance bottleneck: utilising TTL (Transistor-Transistor Logic) triggering for the camera instead of the default software single image mode. TTL triggering is an electronic signalling method that offers a more direct control mechanism, which can help bypass the imposing arming time associated with each single frame collection.

We designed and developed a simple yet effective triggering system to pursue the identified solution. This system is built around the Raspberry Pi 4, a small and powerful single-board computer that offers significant computational capabilities and versatility. The Raspberry Pi 4-based triggering system, implemented to overcome the performance issue, is illustrated in the following section. The depiction of the system offers a clear understanding of its structure, functionality, and how it interacts with the more extensive system to alleviate the identified camera performance issue.



External software trigger is not supported in the camera

Step scan → Image mode: Single, trigger source: Auto

External TTL trigger is supported in the camera

Step scan → Image mode: Multi, trigger source: External

RPTrigServ: PCO edge 5.5 Simple triggering system for step scan ***Eliminates arming time between Frames***

Tomography Software Adaptation and Reconstruction Algorithms

The tomography software, Tomopy has demonstrated its proficiency by successfully reconstructing a full X-ray dataset collected at BEATS on both CPU and GPU nodes. This achievement proves the efficiency of the system. Various CPU and GPU reconstruction algorithms have been evaluated, with the primary pipeline leveraging the fastest algorithm, gridrec, on the CPU. Utilising 96 threads has reduced the gridrec reconstruction time to as low as 60 seconds. Tomopy provides additional methods for image reconstruction. The table below provides an overview of these methods:

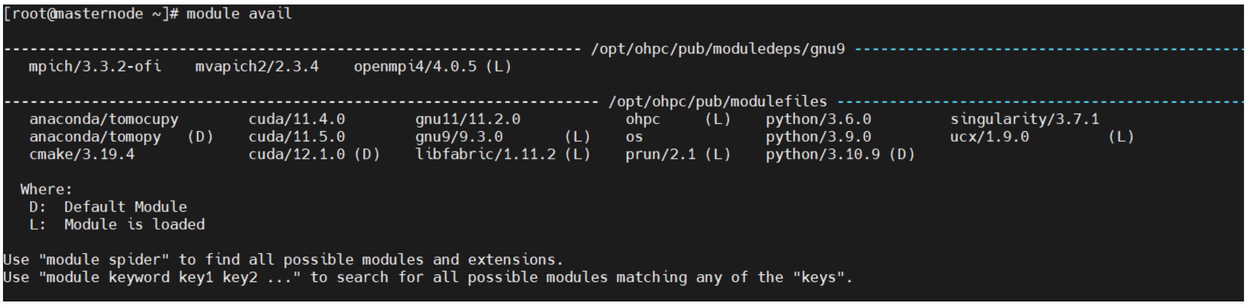
Reconstruction Method	**Description**
Algebraic Reconstruction Technique (ART)	An iterative reconstruction method that uses a projection-based technique to compute tomographic images
Block Algebraic Reconstruction Technique (BART)	A variant of the ART algorithm that processes blocks of data rather than individual data points
Filtered Back-Projection (FBP)	A standard technique for tomographic imaging, using a filtering process followed by a sum-back projection to reconstruct the image

Ftheier Grid Reconstruction	A method that utilises the Ftheier slice theorem to reconstruct the image from the Ftheier space of its projections
Maximum-Likelihood Expectation Maximisation (MLEM)	An iterative method that uses statistical techniques to estimate the image, providing a probabilistic approach to reconstruction
Ordered-Subset Expectation Maximisation (OSEM)	A variant of the MLEM algorithm, designed to improve computational efficiency by dividing data into subsets

Customised Modules Implementation

Based on the Beamline scientist requirements, customised modules have been implemented on the RUM (BEATS HPC cluster). The table below lists the reconstruction codes, and Table (2) and Figure 1, shows the available modules on the RUM cluster.

Package	Instructions URL
astra	https://www.astra-toolbox.com/docs/install.html
tomopy	https://tomopy.readthedocs.io/en/latest/install.html
tomopy-cli	https://tomopycli.readthedocs.io/en/latest/sthece/install.html
tomocupy	https://tomocupy.readthedocs.io/en/latest/install.html
exchange	https://dxchange.readthedocs.io/en/latest/sthece/install.html
tifffile	https://github.com/cgohlke/tifffile
matplotlib	https://matplotlib.org/
numpy	https://numpy.org/
pypng	https://pypng.readthedocs.io/en/latest/png.html#installation-and-overview
scipy	https://scipy.org/install/
scikit-image	https://scikit-image.org/docs/stable/install.html



```
[root@masternode ~]# module avail

------------------------------------------------------------------ /opt/ohpc/pub/moduledeps/gnu9 ---------------------------------------------
   mpich/3.3.2-ofi    mvapich2/2.3.4    openmpi4/4.0.5 (L)

-------------------------------------------------------------------- /opt/ohpc/pub/modulefiles -----------------------------------------------
   anaconda/tomocupy        cuda/11.4.0        gnu11/11.2.0            ohpc     (L)    python/3.6.0         singularity/3.7.1
   anaconda/tomopy   (D)    cuda/11.5.0        gnu9/9.3.0       (L)    os              python/3.9.0         ucx/1.9.0         (L)
   cmake/3.19.4             cuda/12.1.0 (D)    libfabric/1.11.2 (L)    prun/2.1 (L)    python/3.10.9 (D)

  Where:
   D:  Default Module
   L:  Module is loaded

Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

Implementation of the Slurm Scheduling System

The Slurm Workload Manager, a free and open-source job scheduler for Linux and Unix-like kernels, has been successfully integrated into the system. This potent tool enables advanced scheduling capabilities designed for the efficient allocation of resources to tasks. As an integral component of the computational infrastructure, Slurm manages, schedules, and oversees the execution of jobs, facilitating optimal resource utilisation. It possesses the ability to queue tasks, allocate resources, manage workloads, and to provide an interface for user and administrative access, all while ensuring a balanced distribution of resources among the submitted tasks. Moreover, the Slurm system has been developed to handle a broad spectrum of job types, from single-process tasks to parallel jobs spanning numerous processors. It is highly configurable, enabling administrators to fine-tune system parameters to optimise the scheduling and management of jobs based on specific workload characteristics and system architecture.

The successful implementation of the Slurm Workload Manager provides a robust and efficient framework for job scheduling, significantly improving overall productivity and resource efficiency. The system now benefits from streamlined job submission and execution, leading to reduced idle times, maximised throughput, and enhanced performance. It is also expected that the integration of Slurm will contribute to a more seamless user experience, with simplified job submission and monitoring processes.

The scheduling system Slurm, has been successfully implemented and is now fully operational

Reconstruction Pipeline and Tests

The process of image reconstruction is a crucial part of the beamline’s computational infrastructure and represents a crucial part of the activities of work package 7. It involves several important steps and the use of key software tools to ensure that the images are accurately and effectively reconstructed from the data collected during scans.

Overview of the Reconstruction Pipeline

The reconstruction pipeline is a comprehensive series of necessary steps, each designed to refine and improve the data as it progresses through the system. These steps are

1. I/O (Input/Output): This is the fundamental step of handling data where the information is read from or written into the system.
2. Normalisation: This step involves adjusting the values measured to a common scale without distortion. It is crucial for making meaningful comparisons.
3. Center of Rotation (COR) Detection: The COR is a key parameter in tomography reconstruction. It is accurately determined to ensure the image is reconstructed properly.
4. Artefact Correction: Various artefacts may appear in the raw data due to mechanical issues or detector errors. The correction of these artefacts is essential to ensure high-quality reconstructions.
5. Phase Retrieval: This optional step is used in phase contrast imaging to extract phase information from the projections.
6. Reconstruction (Recon): This is the main phase of the pipeline, where the pre-processed data is reconstructed into a 3D image.
7. Extended FOV Reshape: This step is essential when the sample's field of view (FOV) extends beyond the detector size.
8. 8-bit Conversion: The final reconstructed data is converted to 8-bit values for further analysis or visualisation, which enhances compatibility with other systems or platforms.

Each of these functionalities has been rigorously tested to ensure accuracy and reliability.

The Use of Tomopy in the Reconstruction Process

The Python library TomoPy is critical to the reconstruction process. With its versatile features, it enables advanced algorithms for image reconstruction, correction, and analysis. It blends the flexibility of Python with the computational efficiency of compiled languages, making it a perfect tool for tomography research. Here is a brief overview of how TomoPy is utilised:

- **Setting up the Environment:** First step is to establish the Python environment and importing necessary data processing and visualisation libraries.
- **Loading Experimental Data:** Experimental data is then loaded from a .h5 file, which includes projections, flat fields, and dark fields collected from an X-ray tomography experiment.
- **Data Inspection:** The loaded data is examined using the Napari library. This tool provides interactive visualisation of the projections, allowing for immediate review and adjustments if needed.
- **Flat Field Correction:** In this step, projections are normalised using flat and dark field images to account for variations in the X-ray beam.
- **Center of Rotation (COR) Detection:** This step determines the COR, which is critical for accurate CT reconstruction.

- **Reconstruction:** The data is reconstructed using the gridrec algorithm as the default.
- **Post-processing:** Following reconstruction, a series of post-processing steps are applied to improve the final image quality.

The Role of Jupyter Notebook in the Reconstruction Process

The Jupyter Notebook platform is central to the BEATS reconstruction pipeline. Its user-friendly interface enables interactive coding, making executing code in individual blocks possible. This allows for detailed step-by-step analysis and instant review of outputs. Additionally, the platform's support for Markdown, a lightweight markup language for creating formatted text, is invaluable. Markdown in Jupyter Notebook allows for the creation of rich text documents that can include formatted text, images, links, and even pieces of code. This feature is particularly useful for documenting code, explaining what the code is doing, and providing instructions to other users.

The sharing capabilities of Jupyter Notebook also foster team collaboration, and its capacity for conversion into various formats makes it an excellent tool for reporting and presentations. In the BEATS reconstruction pipeline, the support for Markdown allows the team to effectively document their processes, integrating code with rich text elements for clarity and ease of understanding.

Tomography 3D Data Processing and Visualisation

Tomography visualisation holds critical importance in extracting essential insights into collected data's structural and functional characteristics. High-quality visualisation is a cornerstone for in-depth analysis, accurate interpretation, and informed decision-making. Furthermore, it plays a crucial role in simplifying the conveyance of complex information, a paramount factor for effective communication of findings within a team and external entities.

A variety of software tools, each with unique capabilities, play a significant role in enriching the quality of work by offering versatile methods to analyse, visualise, and interpret data. Here is a detailed list of visualisation tools that will be accessible:

- **ImageJ**: This open-source tool, designed for scientific image analysis, enjoys widespread usage owing to its powerful features and user-friendly interface.
- **Paraview**: Paraview, an open-source, multi-platform data analysis and visualisation application, is constructed to process large data sets and execute parallel processing, making it an ideal fit for large-scale projects.
- **Dragonfly**: Offering state-of-the-art 3D visualisation and analysis functionality, Dragonfly enables interactive exploration of large-scale data, significantly enriching analytical and interpretive capabilities.

- **3D Slicer**: An open-source software, 3D Slicer specialises in medical image processing and three-dimensional visualisation of medical images, rendering it useful for projects involving biomedical data.
- **PALABOS**: PALABOS delivers a comprehensive solution for the numerical simulation of fluid flows, applicable when understanding flow dynamics or fluid-structure interactions is involved.
- **Quanfima**: Quanfima is an open-source Python package developed for quality and fidelity assessment of 3D imaging data, equipped with features for visualisation, characterisation, and quantification.
- **Calculix**: As an open-source finite element analysis software, Calculix proves instrumental when a project requires understanding how a system behaves under specific conditions.
- **iMorph**: iMorph, a cross-platform 3D image analysis software, is purpose-built to handle large 3D datasets, making it appropriate for morphometric analyses.

These tools, in combination with advanced computational infrastructure, ensure detailed, high-quality visualisations for a wide array of applications can be produced.

#	name	URL	open source	licence type
	ImageJ	https://fiji.sc/	Yes	
	Paraview	https://www.paraview.org/	Yes	
	Dragonfly	https://www.theobjects.com/dragonfly/index.html	No	Academic; 2 seats
	3D Slicer	https://www.slicer.org/	Yes	
	PALABOS	https://palabos.unige.ch/	Yes	
	Quanfima	https://github.com/rshkarin/quanfima	yes	
	Calculix	http://www.calculix.de/	yes	
	iMorph	http://imorph.sourceforge.net/index.html	no	free

First Reconstruction Tests

Initial tests have been performed to validate the reconstruction pipeline. These include tests on different samples such as Glass beads, Dana Terracotta, and Red Sea Coral. The tests showcase various aspects of the reconstruction process, from absorption vs phase-contrast reconstructions to handling extended FOV scans and comparisons between different scanning modalities. The successful results from these tests confirm the functionality and reliability of the BEATS data pipeline.

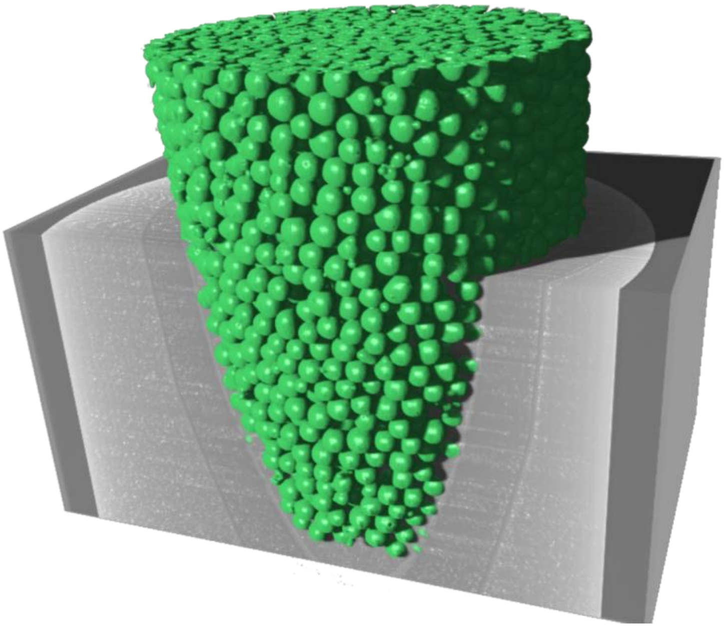
Glass beads

Sample: Glass spheres diameter 300 – 400 µm

Detector: Hasselblad lenses (1x magnification) with ORYX FLIR camera; 4.5 µm voxel size

Scan settings:

- filtered white beam
- Phase contrast reconstruction
- 1000 projections
- 12 ms exposure time
- Scan time: 12 s



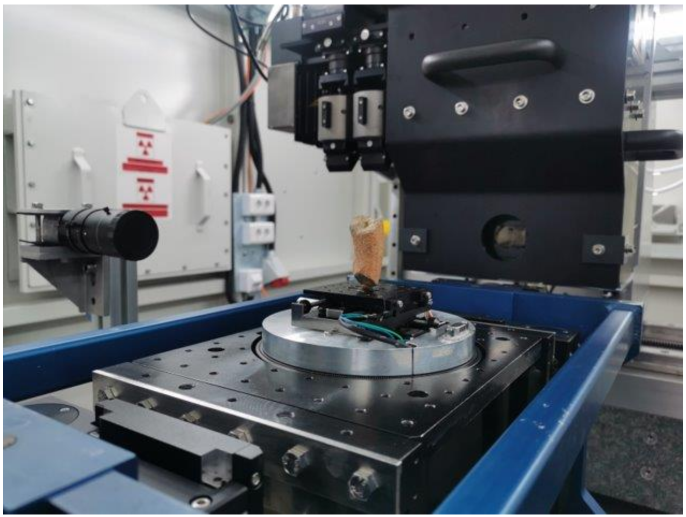
Terracotta

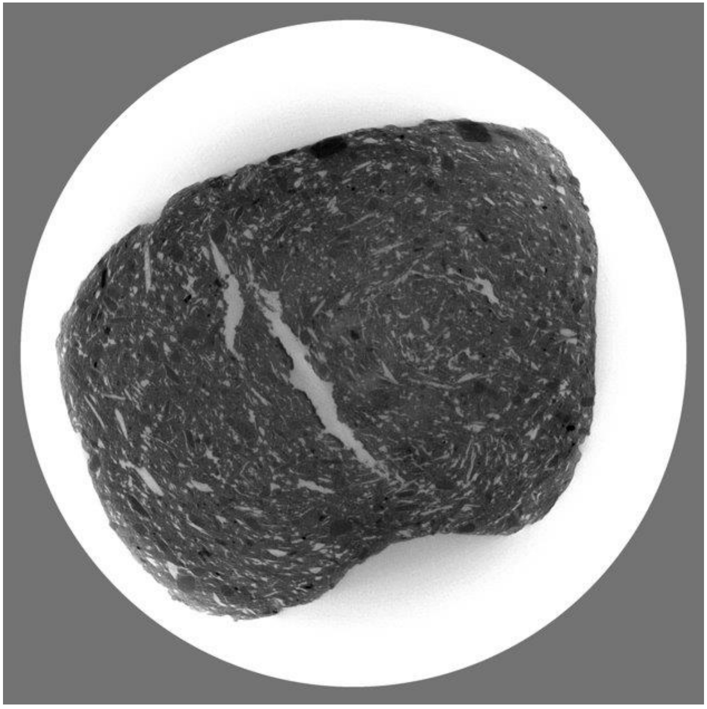
Sample: Terracotta specimen; diameter ~ 20 mm

Detector: Hasselblad lenses (1x magnification) with ORYX FLIR camera; 4.5 µm voxel size

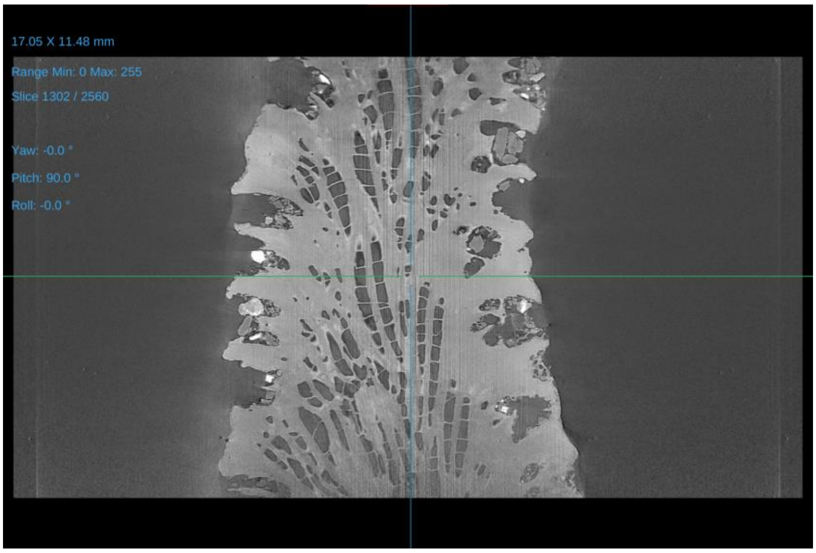
Scan settings:

- filtered white beam
- Phase contrast reconstruction
- 1800 projections; 360 degree scan
- 600 ms exposure time
- Scan time: ~20 mins





Red Sea Coral



17.05 X 11.48 mm

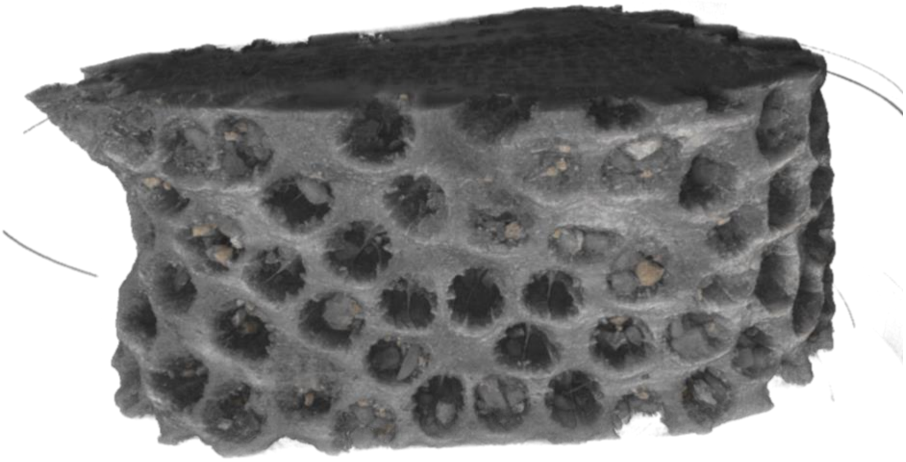
Range Min: 0 Max: 255

Slice 1302 / 2560

Yaw: -0.0 °

Pitch: 90.0 °

Roll: -0.0 °



CONCLUSION

In summary, substantial progress has been made in the project aimed at the adaptation of tomography software. The successful introduction of a sophisticated computing infrastructure, the establishment of a robust and efficient reconstruction pipeline, and the successful execution of a series of reconstruction tests highlight the promising future of this endeavour. The team is looking forward to sharing the upcoming breakthroughs in the field of tomography.