

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Prototyping X-ray tomographic reconstruction pipelines with Flexbox

A. Kostenko^{a,*}, W.J. Palenstijn^a, S.B. Coban^a, A.A. Hendriksen^a, R. van Liere^a,
K.J. Batenburg^{a,b}^a Centrum Wiskunde & Informatica, Science Park 123, 1098XG, Amsterdam, The Netherlands^b Universiteit Leiden, Rapenburg 70, 2311EZ Leiden, The Netherlands

ARTICLE INFO

Article history:

Received 2 May 2019

Received in revised form 15 September 2019

Accepted 22 November 2019

Keywords:

Tomography

Reconstruction algorithms

Data processing

Acquisition geometry

ABSTRACT

Computer Tomography (CT) scanners for research applications are often designed to facilitate flexible acquisition geometries. Making full use of such CT scanners requires advanced reconstruction software that can (i) deal with a broad range of geometrical scanning settings, (ii) allows for customization of processing algorithms, and (iii) has the capability to process large amounts of data. Flexbox is a Python-based tomographic reconstruction toolbox focused on these three functionalities. It is built to bridge the gap between low-level tomographic reconstruction packages (e.g. ASTRA toolbox) and high-level distributed systems (e.g. Livermore Tomography Tools). Flexbox allows to model arbitrary source, detector and object trajectories. The modular architecture of Flexbox allows to design an optimal reconstruction approach for a single CT dataset. When multiple datasets of an object are acquired (either different spatial regions or different snapshots in time), they can be combined into a larger high resolution volume or a time series of volumes. The software allows to then create a computational reconstruction pipeline that can run without user interaction and enables efficient computation on large-scale 3D volumes on a single workstation.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Code metadata

Current code version	1.0.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2019_163
Legal Code License	GPL 3.0
Code versioning system used	git
Software code languages, tools, and services used	Python
Installation can be done using conda or pip	yes, via conda 'cicwi' channel
If available Link to developer documentation/manual	https://cicwi.github.io/flexDATA/ https://cicwi.github.io/flexTOMO/ https://cicwi.github.io/flexCALC/
Support email for questions	w.j.palenstijn@cwi.nl

Software metadata

Current software version	1.0.0.
Permanent link to executables of this version	N/A
Legal Software License	GPL 3.0
Computing platforms/Operating Systems	Linux, Microsoft Windows
Installation requirements & dependencies	numpy, astra-toolbox, scipy, simpleitk, pygraphviz, numpy-stl, scikit-image, pyqtgraph, matplotlib, tqdm, imageio, tiffio, psutil, toml, transforms3d, paramiko
If available, link to user manual – if formally published include a reference to the publication in the reference list	https://cicwi.github.io/flexDATA/ https://cicwi.github.io/flexTOMO/ https://cicwi.github.io/flexCALC/
Support email for questions	a.kostenko@cwi.nl , w.j.palenstijn@cwi.nl

* Corresponding author.

E-mail address: a.kostenko@cwi.nl (A. Kostenko).<https://doi.org/10.1016/j.softx.2019.100364>2352-7110/© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Motivation and significance

Advanced laboratory cone-beam X-ray CT scanners allow researchers to obtain high quality scans of a wide range of specimens through optimization of the source settings and adjustment of the scan orbit. Multiple scans of the same object can be acquired in different configurations in order to increase the total size of the reconstructed volume [1], improve the contrast [2], reduce the reconstruction artifacts [3] or create a time-lapse series [4]. However, such experiments also require access to reconstruction software supporting adjustable processing routines, variable scanning geometries, and large data volumes.

A considerable number of software packages can be used for tomographic reconstructions [5–13]. These projects provide access to a multitude of effective data processing functions and advanced algorithms. Low-level packages like ODL [5] or ASTRA [6] allow to fully optimize reconstruction of a single scan but lack the functionality needed to process large amounts of data, while high-level systems like LTT [9] or CASToR [12] permit to design fast processing pipelines but may lack the flexibility and transparency of a low-level package. In order to close the gap between algorithm-level reconstruction packages and processing pipeline-level distributed systems, we have designed a light-weight Python-based CT reconstruction toolbox – FlexBox. It is organized as a collection of building blocks that can be used along with Python’s scientific computing package `numpy` on two different levels: (1) to design a reconstruction approach optimized for a specific scan, (2) to prototype a reconstruction pipeline for processing arrays of larger-than-memory data on a single workstation.

2. Software description

FlexBox is highly transparent: it is Python-based, which provides easy access to its internal design. It is integrated with ASTRA toolbox, which enables fast multi-GPU reconstructions on a single workstation. Computational times of ASTRA were previously shown to scale almost linearly with the number of available GPUs, allowing to compute 1000^3 voxel volumes in only a few seconds [6]. Other relevant packages are employed, such as `numpy` (linear algebra and out-of-memory arrays support), `scipy` and `skimage` (image processing and analysis), `xraylib` (spectral simulations), `paramiko` and `imageio` (SCP, data I/O), `SimpleITK` (volume registration), `numpy-stl` (surface models).

2.1. Software architecture

Functionality of the FlexBox is divided between three components with submodules: `flexData`, `flexTomo` and `flexCalc` (Fig. 1). `flexData` can be used stand-alone and is providing basic I/O utilities and geometry definitions. It can be integrated with existing ASTRA-based reconstruction routines. `flexTomo` contains building blocks for simulation and reconstruction scripts, as well as a library of subset-accelerated algebraic algorithms. `flexCalc` component is comprised of a higher-level data processing and analysis routines and a toolbox for pipeline prototyping.

2.2. Software functionalities

We consider the following to be the core functionality of FlexBox: (1) simple definition and manipulation of the acquisition geometry, (2) fast subset-accelerated tomographic reconstruction algorithms, (3) prototyping of large data processing pipelines.

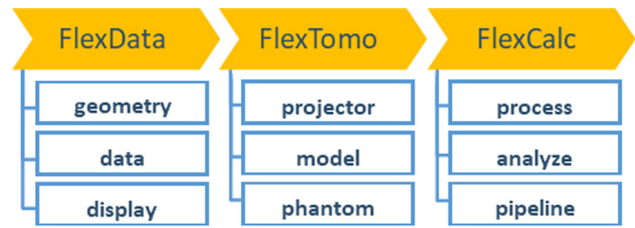


Fig. 1. FlexBox components and their modules.

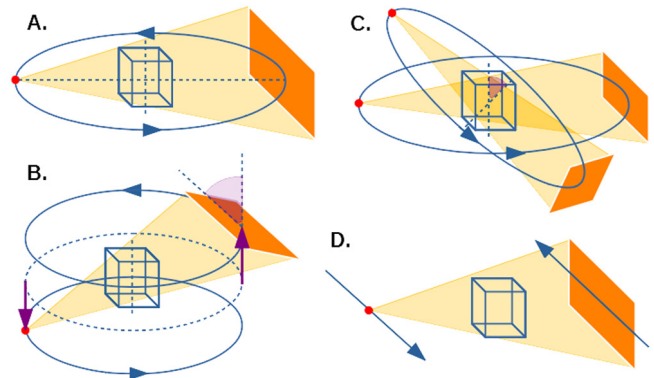


Fig. 2. Geometry definition: (A) a standard circular-orbit cone-beam geometry; (B) laminography geometry achieved by rotating the detector by 45° and introducing a source–detector offset along the axis of rotation; (C) two acquisitions combined via rotation of the reconstruction volume coordinate system; (D) a linear scan geometry.

2.2.1. Geometry definition

While some reconstruction packages can be applied to one of the standard acquisition geometries (e.g. circular orbit cone-beam), ASTRA toolbox provides definition for an arbitrary vector-based geometry. `flexData` enables an intermediate, more intuitive yet flexible definition of an acquisition geometry. It is realized through one of the three geometry classes: *circular* (Fig. 2a), *helical* (Fig. 2b) and *linear* (Fig. 2d). Each class is initialized with a set of basic parameters, e.g. *source to object distance*, *object to detector distance*, *pixel size*, *rotation range* for the *circular* class; and can be adjusted using an extended set of optional parameters e.g. *tilts* and *translations* of the scanning orbit, *detector orientation* etc (Fig. 2b, c).

2.2.2. Tomographic reconstruction

`flexTomo` provides reconstruction and simulation routines for monochromatic or polychromatic cone-beam CT. It contains subset-based back- and forward-projection operators, an implementation of a single gradient-descent step and a few complete algebraic algorithms such as: SIRT, EM, FISTA, PWLS [14]. Splitting the data into subsets serves a double purpose: (1) facilitates reconstruction of a dataset that does not fit into the workstation memory by projecting one subset at a time, (2) speed-up the convergence of gradient-descent-based reconstruction algorithms, making them practically applicable to large datasets.

2.2.3. Pipeline prototyping

A pipeline can be created to either apply same processing and reconstruction steps to multiple datasets or to combine a number of datasets into a larger reconstructed volume or a series of aligned volumes. This is done by scheduling a queue of operations that should be applied to a sequence of projection data arrays. A tree of nodes is automatically created (see Fig. 3a), where each node represents an operation call applied to a single dataset. A

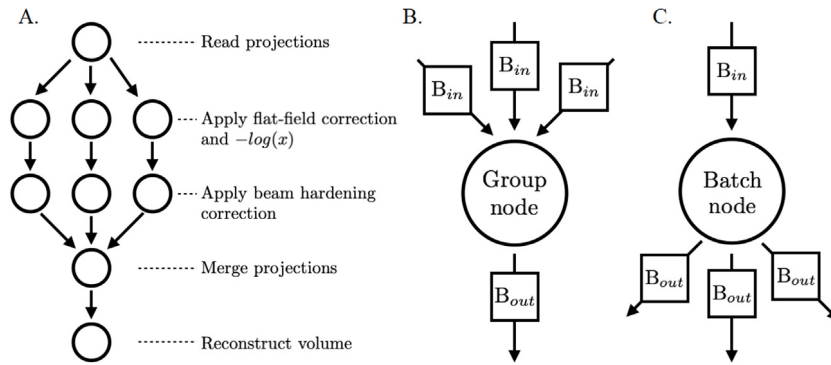


Fig. 3. Batch scheduler: (A) a pipeline for merging several scans into a single reconstruction; (B) group node with multiple input buffers; (C) batch node with one input buffer and several output buffers;

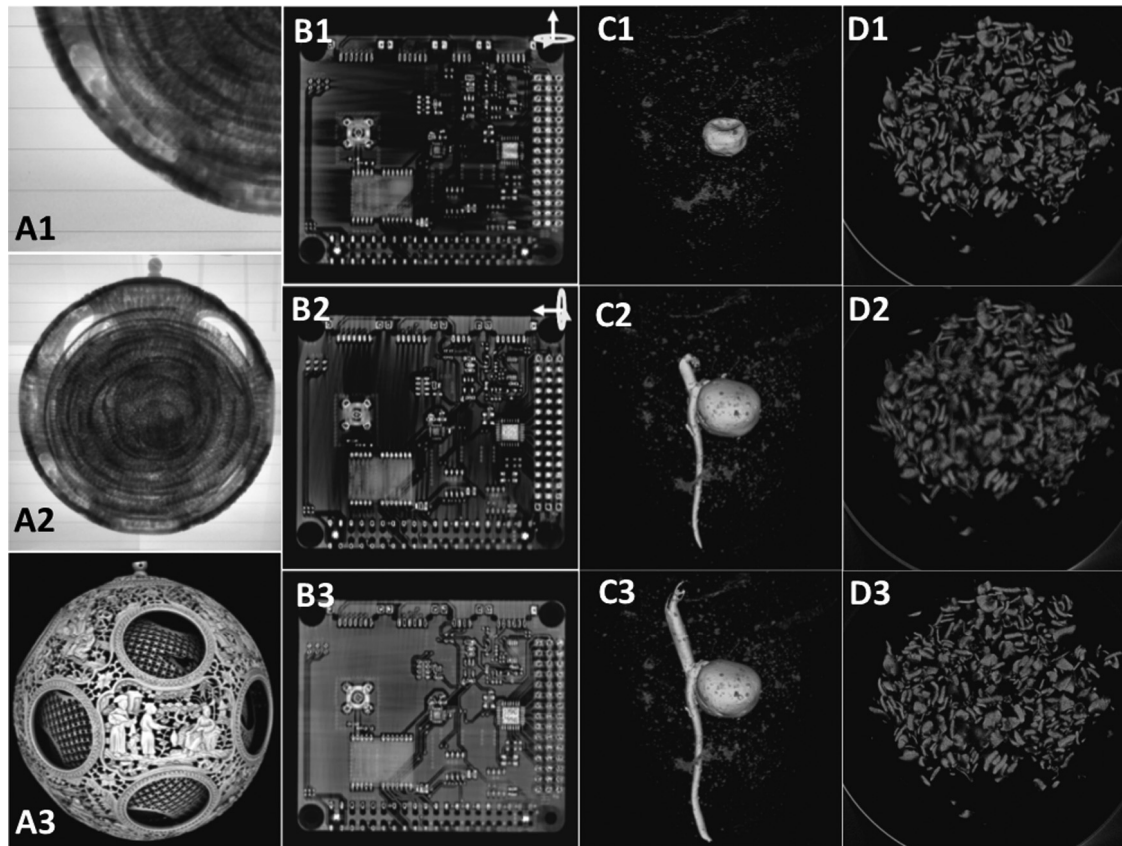


Fig. 4. Example use cases. (A) high-resolution reconstruction through data tiling of cone-beam data: single projection (A1), six stitched projections (A2), rendering of the reconstructed volume (A3). (B) multi-axis reconstruction of a circuit board: B1 and B2 filtered back-projections based on two scans with orthogonal rotation axes; an algebraic reconstruction based on two datasets (B3). (C) time-lapse reconstruction of a growing green pea: C1–C3 day 1, day 5 and day 10 respectively. (D) comparison of the oatmeal reconstructions with: Flexbox (D1), TIGRE (D2), TIGRE, corrected detector tilt (D3)

node may have one or more input buffers and one or more output buffers that reside on disk (Fig. 3b, c). This way the amount of memory used at any point is limited to the amount needed to process a single dataset. If the system crashes, reconstruction can be restarted from the last completed node, once the cause of the failure is addressed.

2.2.4. Other utilities

flexCalc enables the user to rapidly develop a single script for pre-processing, reconstruction and post-processing. It contains such utilities as: single material beam-hardening correction, thresholding, interpolation, volume registration, shape and size measurements, geometry optimization based on auto-focusing.

3. Illustrative examples

Flexbox was initially developed to process data generated by the Flex-Ray scanner [15]; it was used to demonstrate advantages of multi-axis CT acquisition in [3] and is extensively used for reconstructions of tiled high-resolution volumes in the so-called See-Through Museum Project [16].

4. Impact

Different use cases of Flexbox are demonstrated in Fig. 4. First column shows a high-resolution reconstruction of a Chinese

ivory ball [16]. In this example, six scans with a projection size of 1943×1535 pixels were aligned and combined into larger projections to reconstruct a 4000^3 voxel volume.

For this reconstruction around 100 GB (16 bit) of projections were processed, resulting in a reconstructed volume of 220 GB (32 bit) on a workstation equipped with an NVIDIA 1080Ti GPU (10 GB), 128 GB RAM, 1TB SSD hard drive. The disk-footprint of the reconstructed volume was reduced by downgrading to 8 bit and applying ZIP compression. Second column demonstrates a result of a standard single axis filtered backprojection (Fig. 4b1, b2) versus a multi-axis algebraic reconstruction available in flexTomo (Fig. 4b3) [3]. Third column shows an example of a time-lapse, where a green pea was scanned once a day, the results of each reconstruction were aligned using volume registration available in flexCalc. The last column shows a comparison between a reconstruction of a bottom of an oatmeal jar [17] made in Flexbox (Fig. 4d1) with a result obtained using TIGRE package [18] (Fig. 4d2, d3). This data exhibits a small detector tilt that was corrected using geometry parameters optimization in flexCalc (Fig. 4d1). Uncorrected reconstruction that was made using TIGRE package is slightly defocused (Fig. 4d2). However, once the detector tilt estimated by flexCalc was substituted into the TIGRE geometry, a sharp reconstruction was achieved (Fig. 4d3).

In the oatmeal example, we reconstructed a cropped 1000^3 voxel volume from a $1944 \times 1536 \times 1001$ projection dataset. This computation was performed on a workstation with an NVIDIA GTX 970 GPU (4 GB), 64 GB RAM, 6-core Intel Xeon 3.5 GHz CPU. Calculation using Flexbox (ASTRA-based) took 70 s, while the calculation based on TIGRE took 250 s.

5. Conclusions

Advanced X-ray CT systems such as XRE UniTOM, Xradia Versa, SKYSCAN 1273 as well as some open-design systems such as [19] are highly adjustable and can generate tomographic data in a wide range of unconventional geometries. Flexbox can provide aid in utilization of such systems as it is focused on three objectives: (1) flexible geometry definition, (2) large data array support, (3) pipeline prototyping.

By using Flexbox, one can quickly prototype an optimal reconstruction pipeline customized for different applications: high resolution tiled scans (limited to 5000^3 voxel volumes in our practice); reduction of reconstruction artifacts through accelerated iterative algorithms (moderate size volumes of 1000^3 – 2000^3 voxel); reduction of metal artifacts through multi-axis scanning; and time-lapse reconstructions (combining multiple data into a time series).

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgments

We are grateful to XRE, Ghent for providing us with the Flex-ray scanner; to the Rijksmuseum and the Naturalis Bio-Diversity

Centre in Leiden for providing us with extraordinary specimens for scanning. We acknowledge financial support from The Netherlands Organization for Scientific Research (NWO) (project 341-60-001).

References

- [1] Vescovi R, Du M, de Andrade V, Scullin W, Gursoy D, Jacobsen C. Tomosaic: efficient acquisition and reconstruction of teravoxel tomography data using limited-size synchrotron X-ray beams. *J Synchrotron Radiat* 2018;25(Pt 5):1478–89. <http://dx.doi.org/10.1107/S1600577518010093>.
- [2] McCollough CH, Leng S, Yu L, Fletcher JG. Dual- and multi-energy CT: Principles, technical approaches, and clinical applications. *Radiology* 2015;276(3):637–53. <http://dx.doi.org/10.1148/radiol.2015142631>.
- [3] Kostenko A, Andriashen V, Batenburg KJ. Registration-based multi-orientation tomography. *Opt Express* 2018;(26):28982–95. <http://dx.doi.org/10.1364/OE.26.028982>.
- [4] Lowe T, Garwood RJ, Simonsen TJ, Bradley RS, Withers PJ. Metamorphosis revealed: time-lapse three dimensional imaging inside a living chrysalis. *J R Soc Interface* 2013;10. 201303. <http://dx.doi.org/10.1098/rsif.2013.0304>.
- [5] Operator Discretization Library. <https://odlgroup.github.io/odl/>. [Accessed 01 April 2019].
- [6] Palenstijn WJ, Bédorf J, Sijbers J, Batenburg KJ. A distributed ASTRA toolbox. *J Adv Struct Chem Imag* 2016;23(3):842–9.
- [7] Gursoy D, De Carlo F, Xiao X, Jacobsen C. Tomopy: a framework for the analysis of synchrotron tomographic data. *J Synchrotron Radiat* 2014;21(5):1188–93.
- [8] Vogelgesang M, Chilingaryan S, dos Santos Rolo T, Kopmann A. UFO: a scalable GPU-based image processing framework for on-line monitoring. *IEEE HPCC & ICESS* 2012;(6):824–9.
- [9] Livermore Tomography Tools for image reconstruction, <https://nci.llnl.gov/content/assets/docs/ChampleyALSmeeting.pdf>. [Accessed 01 April 2019].
- [10] Thielemans K, Mustafovic S, Tsoumpas C. STIR: Software for tomographic image reconstruction release 2. In: *IEEE nuc. sci. sympos. conf. rec.* 2006, p. 2174–6. <http://dx.doi.org/10.1109/NSSMIC.2006.354345>.
- [11] Rit S, Vila Oliva M, Brousmiche S, Labarbe R, Sarrut D, Sharp GC. The reconstruction toolkit (RTK), an open-source cone-beam CT reconstruction toolkit based on the insight toolkit (ITK). *J Phys: Conf* 2014;(489):1742–6588. <http://dx.doi.org/10.1088/1742-6596/489/1/012079>.
- [12] Thibaut M, Stute S, Didier B, Julien B, Carlier T, Comtat C, Filipovic M, Lamare F, Visvikis D. Castor: a generic data organization and processing code framework for multi-modal and multi-dimensional tomographic reconstruction. *Phys Med Biol* 2018;63(18). <http://dx.doi.org/10.1088/1361-6560/aadac1>.
- [13] Pedemonte S, Bousse A, Erlandsson K, Modat M, Arridge S, Hutton BF, Ourselin S. GPU Accelerated Rotation-Based Emission Tomography Reconstruction, *NSS/MIC* 2010. p. 2657–61. <http://dx.doi.org/10.1109/NSSMIC.2010.5874272>.
- [14] Paleo P. Iterative methods in regularized tomographic reconstruction. Thesis. <https://www.theses.fr/2017GREAT070.pdf>. [Accessed 15 September 2019].
- [15] Coban SB, et al. Flex-ray Lab: A highly flexible scanner for explorative tomography. 2019 [in preparation].
- [16] van Liere R, Batenburg KJ, Kostenko A, Wang CL, Garachon I. Imaging ancient Chinese ivory puzzle balls: deducing the make process. In: Sixth international workshop on image processing for art investigation. 2018, p. 48–50.
- [17] Coban SB, Hendriksen AA, Daniël MP, Palenstijn WJ, Batenburg JK. Data: Experimental cone-beam tomographic data for techniques to improve image resolution, Zenodo Data, v1; 2018. <http://dx.doi.org/10.5281/zenodo.2657644>.
- [18] Biguri A, Dosanjh M, Dosanjh M, Hancock S, Soleimani M. TIGRE: a MATLAB-GPU toolbox for CBCT image reconstruction. *Biomed Phys Eng Express* 2016;2:055010. <http://dx.doi.org/10.1088/2057-1976/2/5/055010>.
- [19] Masson-Berghoff A, Absent O'Flynn D. Invisible or revealed 'relics'? X-radiography and CT scanning of Egyptian bronze votive boxes from naukratis and elsewhere. *Brit Museum Stud Ancient Egypt Sudan* 2019;24:159–74.