

Real-time image-content-based beamline control for smart 4D X-ray imaging

Matthias Vogelgesang,^{a,*} Tomas Farago,^{b,c} Thilo F. Morgeneyer,^d Lukas Helfen,^{b,e} Tomy dos Santos Rolo,^f Anton Myagotin^g and Tilo Baumbach^{b,c}

Received 10 March 2016

Accepted 21 June 2016

Edited by P. A. Pianetta, SLAC National Accelerator Laboratory, USA

Keywords: control; tomography; laminography.

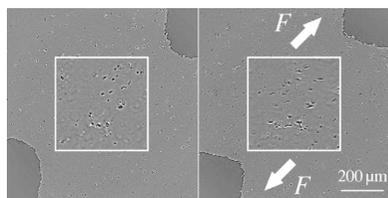
^aInstitute for Data Processing and Electronics, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany, ^bInstitute for Photon Science and Synchrotron Radiation, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany, ^cLaboratory for Applications of Synchrotron Radiation, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany, ^dMINES ParisTech, PSL Research University, MAT – Centre des Materiaux, CNRS UMR 7633, BP 87, 91003 Evry, France, ^eESRF – The European Synchrotron, 71 avenue des Martyrs, 38000 Grenoble, France, ^fInstitute for Accelerator Physics and Technology, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany, and ^gSt Petersburg State University of Civil Aviation, St Petersburg, Russia. *Correspondence e-mail: matthias.vogelgesang@kit.edu

Real-time processing of X-ray image data acquired at synchrotron radiation facilities allows for smart high-speed experiments. This includes workflows covering parameterized and image-based feedback-driven control up to the final storage of raw and processed data. Nevertheless, there is presently no system that supports an efficient construction of such experiment workflows in a scalable way. Thus, here an architecture based on a high-level control system that manages low-level data acquisition, data processing and device changes is described. This system is suitable for routine as well as prototypical experiments, and provides specialized building blocks to conduct four-dimensional *in situ*, *in vivo* and *operando* tomography and laminography.

1. Introduction

Synchrotron-radiation-based imaging diagnostics has become a reliable tool for systematic examination of chemical and biological samples in various research areas. Progress in photon flux density and advanced pixel array detectors with high spatial resolution, low noise and fast read-out, as well as fast and high-precision positioning manipulators, permit much reduced measuring time compared with solutions based on laboratory sources. Using these technologies, synchrotron radiation computed tomography (SRCT) experiments (Thompson *et al.*, 1984; Stock, 2008; Westneat *et al.*, 2008) can be extended to achieve data acquisition (DAQ) times of less than a second for a complete three-dimensional (3D) dataset (Jung *et al.*, 2012; Salvo *et al.*, 2012; Finegan *et al.*, 2015). Combined with optical flow analysis, this has enabled the development of *cine-tomography*, a technique that allows for the characterization of four-dimensional (4D) spatiotemporal structure evolution of technical and biological processes (dos Santos Rolo *et al.*, 2014).

Synchrotron radiation computed *laminography* (SRCL) has extended the applicability of 3D synchrotron imaging to thin plate-like objects (Helfen *et al.*, 2011) that otherwise prohibit homogeneous transmittance using conventional tomographic scans. It is especially suited to flat specimens which cannot be trimmed down (*e.g.* due to their uniqueness or the loss of their function) and thus do not fit into the detector's field of view (FOV) (Houssaye *et al.*, 2011; Reischig *et al.*, 2013). Moreover,



SRCL is suitable for *in situ* imaging studies of the impact of mechanical and thermal loads, e.g. to assess damage processes in carbon-fibre reinforced plastics (Xu *et al.*, 2010; Bull *et al.*, 2013), coatings (Maurel *et al.*, 2013) and alloy sheets (Morgeneyer *et al.*, 2013, 2014). Apart from *in situ* studies, SRCL enables *operando* experiments for failure analysis and lifetime prediction (Tian *et al.*, 2011).

Both SRCT and SRCL are used with a large variety of contrast modes such as phase contrast (Cloetens *et al.*, 1999; Weitkamp *et al.*, 2005; Harasse *et al.*, 2011; Altapova *et al.*, 2012), fluorescence (de Jonge & Vogt, 2010; Xu *et al.*, 2012a) and diffraction contrast (Ludwig *et al.*, 2008; Hänschke *et al.*, 2012) which have many applications in materials research (Yazzie *et al.*, 2012; Boden *et al.*, 2014), microsystem technology, cultural heritage, paleontology (Riedel *et al.*, 2012) and biology (Walker *et al.*, 2014; van de Kamp *et al.*, 2015; Greven *et al.*, 2015).

However, due to the complexity of the experimental setup, SRCT and in particular SRCL pose challenging problems for automation and software-controlled experiments. For a successful automated scan, the imaging and sample apparatus must be aligned and positioned properly, the sample be stabilized and the measurement setup controlled during the imaging process. Moreover, intelligent control of the imaging process with respect to the unpredictable spatiotemporal localization of the region of interest (ROI) and its evolution is particularly difficult to achieve because it requires information about the process under study. In the worst case the ROI may be completely missed or leave the FOV during the experiment. This problem becomes worse if the 4D resolution requirements of the studied details are not known *a priori* or change during the scan. Although two-dimensional (2D) radiographs already contain sufficient information for fast online feedback in many applications, in certain cases control decisions must be based on quality metrics that can only be derived from 3D or 4D image reconstruction.

Given that these problems are solved, conventional experiment acquisition schemes can be replaced by proactive workflows enabling us to:

- (a) Acquire sequences of tomo- or laminographic snapshots at different user-controlled loading stages of materials and devices.
- (b) Record continuous 3D and 4D film sequences at well defined stages of biological or technical processes.
- (c) Drive an experiment based on metrics derived from 3D and 4D reconstructions.
- (d) Achieve high sample-throughput with automatic quality assurance.

Apart from the progress made in sample throughput (Mader *et al.*, 2011) and user-friendly analysis tools (Gürsoy *et al.*, 2014), a necessity for data acquisition driven by 3D and 4D image metrics are fast image reconstruction, metric evaluation and feedback to the experimental set-up. Although modern GPUs reconstruct volumes in near *real-time* (Chilingaryan *et al.*, 2011; Myagotin *et al.*, 2013), integrating data processing with a decision-making processes and hardware feedback is still missing.

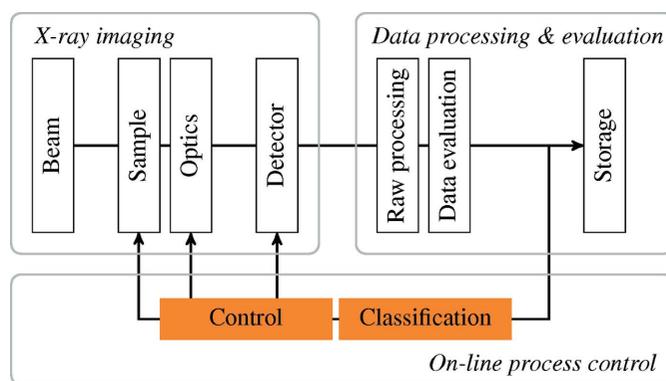


Figure 1

Schematic overview of the experimental workflow. The upper part is a regular X-ray imaging setup with subsequent data processing and storage. The lower part is a novel classification and control step enabling feedback loops.

In this paper we address the outlined experimental challenges by developing concepts, tools and methods for smart image recording. The abstract high-level architecture of our system that implements these ideas is depicted in Fig. 1. It allows the user to describe and execute computationally intensive experiment workflows in a flexible way. These workflows encompass self-alignment, interactive or automated control of 3D and 4D image quality, identification, tracking and repositioning of the 3D-ROI into the FOV and online control of the spatiotemporal resolution during the whole image recording process. As a proof of concept, we conducted two image-driven experiments. The first one automatically optimizes the temporal resolution based on the tomographic reconstruction of the currently scanned sample. The second experiment outlines an interactive laminography experiment that uses online reconstruction to assess the quality of the acquired data.

2. Experimental problems

From the list of general challenges, we outline specific problems that need to be solved in order to run successful automated scans.

For both SRCT and SRCL the specimen should be centered with respect to the focal point of the scanning geometry in order to ensure that the entire ROI is imaged. This can turn out to be a challenging problem when the specimen is laterally (*i.e.* perpendicular to the rotation axis) much larger than the ROI to be imaged. As seen from Fig. 2, the sample ROI must be positioned exactly at the intersection of the rotation axis and the beam, *i.e.* inside the volume depicted in dark grey. With specimens exhibiting low contrast or showing similar and overlapping features in the projections, deduction of the exact ROI position from projection images alone is sometimes impossible. An incorrect positioning causes the ROI to leave the FOV during a full rotation which results in incomplete data for further reconstruction.

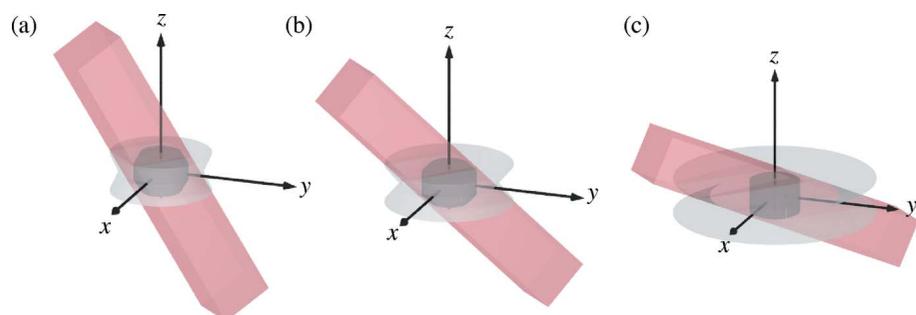


Figure 2
Coverage of the real space inside a flat specimen (positioned at the origin of the z axis) for imaging of a sample in a laminographic setup with beam angles of 30° (a), 45° (b) and 60° (c) with respect to the rotation axis (corresponding to z). The pink bar gives the beam direction for one particular projection direction. The regions sketched in grey are obtained by rotating this pink bar around the rotation axis. The light grey region is covered by at least one projection and the dark grey region is covered by the set of all projection directions. For the CT case with 90° angle, the dark grey zone corresponds to a cylinder.

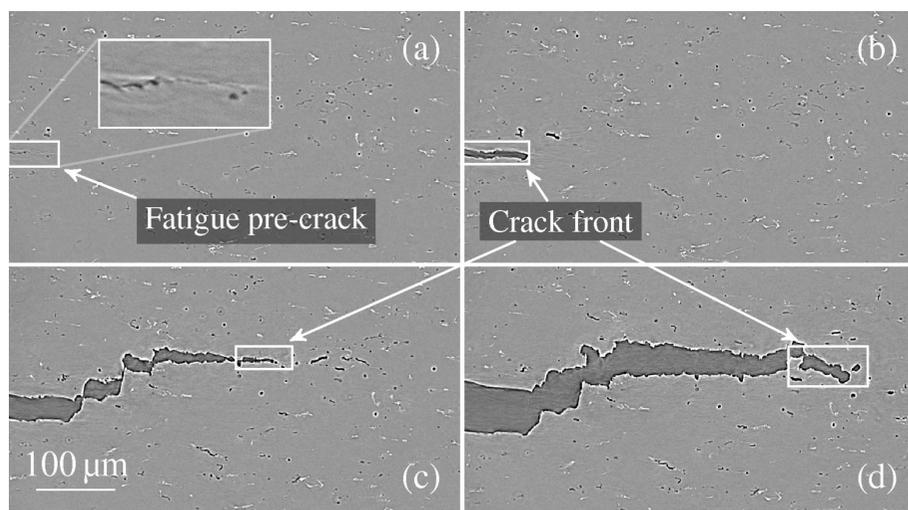


Figure 3
2D section of reconstructed 3D *in situ* laminography data where a hardly visible fatigue pre-crack (a) initiates the subsequent crack propagation which develops into macroscopic damage (b)–(d) under tensile loading of an AA6061 alloy fracture toughness specimen.

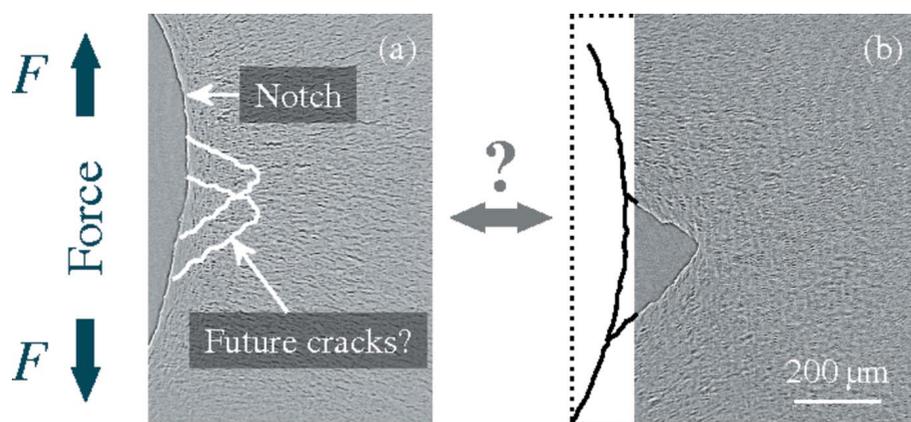


Figure 4
2D section of reconstructed 3D *in situ* laminography data showing damage in a polyamide 6 specimen. While the notch is visible in (a), it is lost when following the forming crack in (b).

We encountered this problem in a crack propagation experiment where a small fatigue-induced pre-crack initiates a larger crack that develops further under continuous load (Shen *et al.*, 2013). For subsequent analysis both the pre-crack as well as the crack front have to be in the FOV during the entire time, and thus the ROI needs to expand and shift. As seen in Fig. 3(a), the pre-crack is clearly visible in the 3D reconstruction; however, it could not be identified solely from the projection images because of its small size and lack of contrast. Without online reconstruction the ROI cannot be reliably identified and the experiment has to be carried out in a blind manner.

In a second example, shown in Fig. 4, the ROI shifted considerably from the initial position in Fig. 4(a) in which the machined notch serves as a reference position (Cheng *et al.*, 2016). Without 3D image control the reference position was lost in Fig. 4(b). Thus, from the data alone, the evolution of certain damage features can no longer be followed directly and requires complex post-processing.

Missing prior knowledge that could be acquired with online reconstruction can also introduce noise and disturbance in subsequent steps such as final high-quality reconstructions. For example, in the second experiment local stress relaxation caused unwanted deformation of the structure in the ROI that progressed during the scan and leads to improper reconstruction with doubled edge artifacts shown in Fig. 5(a). Another example shows a badly reconstructed slice, Fig. 5(b), of a dynamic foaming experiment shown next to a good one, Fig. 5(c). Here, the problem is caused by the sampling time of the volumes which was below the process speed. Because the temporal resolution of these dynamic processes is not known exactly before the experiment is conducted, it must be determined online.

3. System components

Due to the large variety in terms of experimental setup and parameter space, we need a flexible and modular

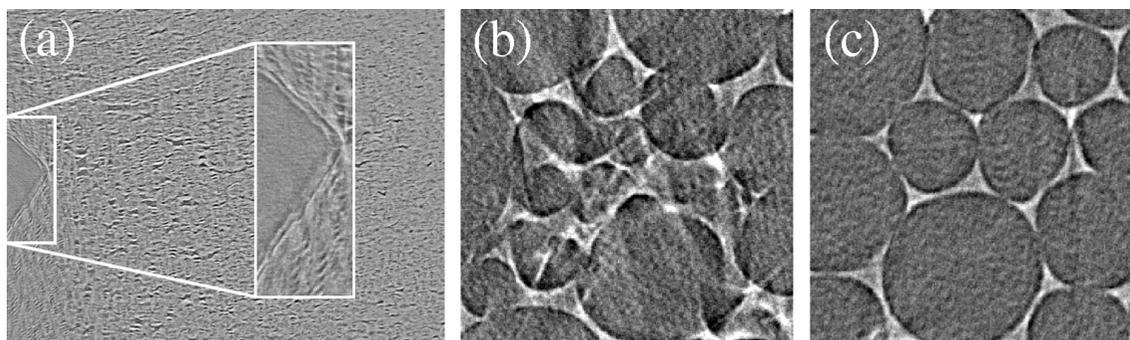


Figure 5

Unsatisfactory reconstructions due to sample movement caused by local stress relaxation (a) and insufficient sampling in time (b) and (c). Online reconstruction would have revealed issues in the left example early on and helps determining the best sampling time leading to fewer reconstruction artifacts as in (c).

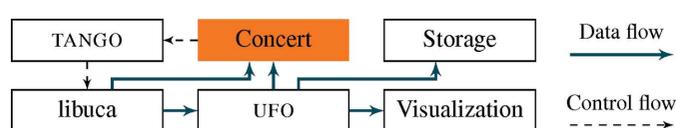


Figure 6

High-level system architecture with the *Concert* core component using TANGO and EPICS devices for slow motor control and the UFO framework for fast data processing.

```

UcaCamera *camera =
    uca_plugin_manager_get_camera("pco");
g_object_set(camera, "exposure-time", 1.2, NULL);

unsigned width, height;
g_object_get(camera, "roi-width", &width,
              "roi-height", &height, NULL);

char *buffer = malloc(width * height * 2);
uca_camera_start_recording (camera);
uca_camera_grab (camera, buffer);
  
```

system to solve the problems stated in the previous section. As shown in Fig. 6, our system separates data acquisition, data processing and experiment control into distinct components. The central control system component *Concert* (<https://github.com/ufo-kit/concert>) forwards the acquired data to the processing component and uses the result to further drive control decisions. In order to respect latency and bandwidth requirements on the one hand and provide easy access on the other, all low-level components are written in portable C that can communicate with the control system that is written in Python.

3.1. Data acquisition

The development of our data acquisition library *libuca* (<https://github.com/ufo-kit/libuca>) was driven by two central requirements: the need for a common application programming interface (API) that covers a variety of 2D pixel detectors and the lowest possible latencies with the highest possible throughput. The library is written in object-oriented C using the GObject API, with a base camera class providing a general interface for initialization, triggering and data readout. The data readout uses different buffering and synchronization modes to cover streaming and non-streaming cameras. The device-specific camera classes inherit from the base class and add properties to describe parameters that are not covered by the general API. The property metadata includes type, valid value range, textual description and an optional SI unit describing the value in physical terms. The following example shows how properties are accessed and a single frame is requested:

Depending on the specific use case, we have to consider different requirements concerning synchronous execution, latency and throughput. Camera implementations that are able to write *into* user memory allow for lowest possible latencies. To help sustain situations where the user cannot process the data in time, a software-side ring buffer can be filled asynchronously by an acquisition thread. No matter how data are acquired, the user can register callback functions to receive data asynchronously.

For high-speed *remote* data acquisition, we use a secondary InfiniBand data channel, that is independent of the TANGO control layer (Dritschler *et al.*, 2014). This approach provides transparent access to the camera data at a peak throughput of 31 Gb s^{-1} on a $4 \times$ quad data rate network.

3.2. Data processing

Synchrotron X-ray imaging experiments employ data processing tasks that range from simple image adjustments such as brightness and contrast corrections to computationally intensive algorithms like tomographic reconstruction and high-quality image denoizing. The majority of these problems can be described in terms of *parallel stream processing*. Parallel computer architectures such as multi-core CPUs and GPUs have become a commodity and now offer floating point performance of 5 TFLOP s^{-1} and a memory bandwidth of 330 GB s^{-1} which rivals the specifications of a supercomputer from the late 1990s at a fraction of the cost.

To process image streams on a heterogeneous computer system consisting of multi-core CPUs and GPUs we use the UFO data processing framework (Vogelgesang *et al.*, 2012). With this framework, algorithms are specified as pipelines

or graphs of simple atomic filters which process data flows. Depending on the specific task, massively parallel filter implementations use OpenCL kernels to execute code on accelerators such as GPUs or the Xeon Phi coprocessor from Intel[®]. At run-time, a scheduler distributes the work among available hardware resources such as CPUs, GPUs and remote network nodes to achieve almost linear scalability with respect to the number of processing units. The UFO framework is written in C and uses the GObject type system to provide an object-oriented interface to the subsystems similar to our data acquisition library. Using GObject introspection, the end user can access the high-performance computing system from a Python environment.

3.3. Reconstruction benchmarks

To establish workflows that solve the aforementioned problems such as following the scan or sequence of scans of a sample, we need to evaluate an updated subset of the reconstructed cross-section slices after every n th projection. Such real-time 3D imaging scenarios are only possible by processing the data stream on-the-fly using efficient reconstruction algorithms.

The filtered backprojection (FBP) algorithm is the best choice for two reasons: (i) it inherently provides a refined 3D image and (ii) the backprojection part requires only $\mathcal{O}(N^2)$ operations to process one X-ray projection of $N \times N$ pixels per slice. On the other hand the direct Fourier inversion method requires $\mathcal{O}(N^2 \log N)$ operations because the 2D inverse fast Fourier transform is needed every time a slice is updated.

We optimized the tomographic and laminographic FBP for current-generation GPUs to achieve the throughput required for online reconstruction. The tomographic FBP algorithm reconstructs individual slice pixels independently by using a large number of GPU compute units and individual slices independently using multiple GPUs, *i.e.* we employ fine- and coarse-grained parallelization to reduce the reconstruction time. The laminographic FBP algorithm is based on the work presented by Myagotin *et al.* (2013) and is parallelized in the same way as the tomographic case. However, the data are backprojected to a partial 3D volume instead of individual 2D slices.

For a realistic impression about latency and throughput behaviour we present benchmark data for tomographic and laminographic reconstruction and compare their effective performance. We swept across N which determines both the number of projections, a projection size of $N \times N$ pixels and a reconstruction size of $N \times N$ pixels per slice. For laminographic reconstruction, we used an axis tilt angle θ of 65° . We conducted the benchmarks on a system with seven NVIDIA[®] GeForce GTX TITAN GPUs that are based on NVIDIA's Kepler GK110 architecture. On this system we measured both the time to transfer data from the host to the GPU and to actually process the data. We excluded the time to transfer the resulting slices back to the host because subsequent metrics needed for control can be derived from the reconstructed

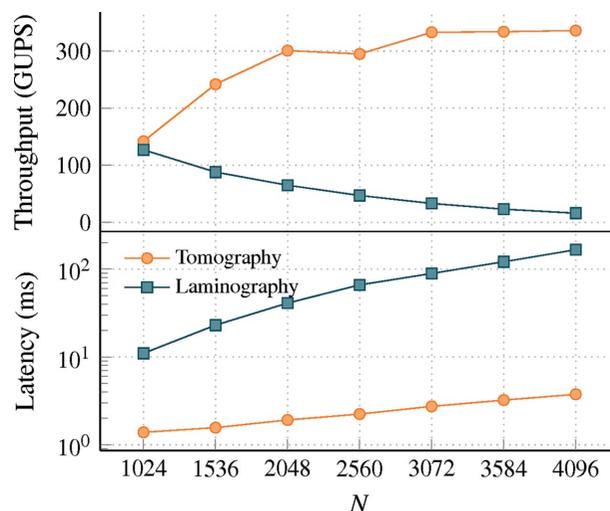


Figure 7 Data throughput (top) and latency (bottom, in logarithmic scale) measured for tomographic and laminographic filtered backprojection algorithms on a machine with seven GPUs.

volume in GPU memory. In most cases, such metrics are scalar values (*e.g.* if to repeat a scan, change a parameter to a new value, ...), which require only a few bytes compared with the input data and therefore justifies neglecting the time spent for downloading the scalar value. On this particular hardware architecture, processing 16 projections per kernel invocation yielded best performance with an execution throughput of about 90 Giga-updates per second (GUPS) (Myagotin *et al.*, 2013) on a GTX TITAN, which is similar to computed tomography (CT).

To compare both algorithms we measured the *throughput* and the *latency*. The throughput is quantified by dividing the number of additions for every voxel by the measured time and is given in GUPS. The latency is the time in milliseconds to copy a required projection region to the GPU and backproject it onto a slice. The throughput and latency results for $1024 \leq N \leq 4096$ are shown in Fig. 7.

Due to the geometry of a laminographic setup, the number of detector rows that need to be copied to the GPU depends on the axis tilt angle θ . To reconstruct the relevant area in a slice for a given θ , we have to backproject from a rectangular region of height $N \cos(\theta)$, whereas in CT one row from every projection is sufficient. That means we need 846 detector rows for 2000 projections and a tilt angle of $\theta = 65^\circ$ (*e.g.* the data of Fig. 3). As a consequence, laminographic backprojection cannot hide memory transfers completely as seen in the decreasing throughput. This behavior is pronounced for large N because more data must be copied yet the amount of work cannot be increased. This is caused by the memory limitation of 6 GB on a GTX TITAN.

Although our algorithm maximizes the compute-to-transfer-time ratio by copying only the required projection parts and backprojecting to 3D volumes instead of 2D slices, this does not remedy the problem completely. Due to the memory limitation, the tomographic backprojection outperforms the laminographic one both in throughput and latency.

3.4. Experiment control

To model and run different types of experiments conducted at an imaging beamline, a beamline scientist has to control hardware devices and describe process sequences using these devices. Moreover, for online and feedback-driven experiments, integration of the presented high-throughput data acquisition and high-performance computing facilities is required. To fulfil these different requirements, we designed and developed the *Concert* experiment control system (Vogelgesang *et al.*, 2013).

The entire system is written in Python and uses the widely deployed TANGO/PyTango infrastructure to access the majority of high-latency hardware devices such as motors. It is based on an asynchronous execution model to parallelize slow device access of motors and shutters, thus reducing overall experiment run-times. A typical example illustrates the benefits: during CT scans image sequences are acquired with closed and opened shutter as well as with and without a sample. Instead of sequencing the motor movements, we can move the sample into the FOV and start rotating the sample while the shutter is closed. Moreover, *Concert* forwards acquired data to the processing facilities and uses derived measures to control the process itself, thus closing the feedback loop.

4. Experiment workflows

In this section we will outline how the *Concert* experimental control system is used to define and run workflows and how the low-level facilities are integrated.

4.1. Implementation

Using automatically generated language bindings, our low-level data acquisition and computing libraries can be used within Python, thus allowing us to move time-critical tasks to external, accelerated C and GPU code. Instead of using these low-level libraries directly, we integrated them into a *coroutine*-based workflow system to enforce decoupling of independent subsystems and increase modularity. A coroutine is a function that can pause at defined program points and resume later, thus preserving internal program state (Moura & Ierusalimsky, 2009). Using coroutines, arbitrary workflows are defined by passing a target coroutine as an argument to a source coroutine; as an example, an online reconstruction pipeline can be written as:

```
acquire(sinograms(backproject(write())))
```

This scheme allows for flexible re-arrangement of the pipeline. For example, pre-viewing and writing the reconstructed slices during the reconstruction at the same time is achieved by *broadcasting* the received data to two different target coroutines:

```
acquire(sinograms(backproject(broadcast(write(),
                                   view()))))
```

Although the coroutine-based approach provides flexibility for defining workflows, coroutines are not reusable within the same process such as a long-running experiment session. To remedy this situation, we wrap coroutines in reusable functions and Acquisition classes as explained in §4.3.

4.2. Data management

Complex imaging experiments require structured storage of acquired data and metadata describing these data. We provide a generic Walker mechanism to write structured data decoupled from the storage format. To outline a hierarchical data structure, the user descends and ascends along a path and marks data for storage at the desired locations:

```
def write(walker):
    walker.descend('scan01')
    # acquire data in 'scan01/flats'
    acquire(walker.write('flats'))
    acquire(walker.write('radios'))
```

Depending on the requirements, data are either written as flat files in a newly created directory hierarchy or stored in a HDF5 file (Folk *et al.*, 1999) with HDF5 groups that reflect the hierarchical experiment structure:

```
write(Hdf5Walker('result.hdf5'))
write(DirectoryWalker(root='/tmp'))
```

4.3. Experiment composition

During an experiment users typically acquire normalization, sample and metadata. Each acquisition type is modeled by a separate Acquisition object allowing users to run experiments several times to investigate the outcome of parameter changes. An Experiment class comprises multiple acquisitions and executes them in a defined order. To react to changing requirements, acquisitions can be added, removed and re-arranged at run-time. This way we can create flexible experiment workflows that represent a single ‘run’ of an experiment. Addon objects can be attached to an experiment in order to process data on-the-fly. Current addon implementations cover tasks such as live preview, data storage and online reconstruction.

To avoid redundancies and allow users to work with the system without knowing the underlying architecture we have prepared experiment classes and data acquisition routines that can be combined in order to run standard synchrotron experiments like radiography, CT and others. For example, to conduct a CT experiment with online reconstruction, we create a Setup object which encompasses an experiment’s devices and their parameters as well as a TomographyScan object which rotates the sample with the proper angular velocity:

```
setup = Setup(camera, shutter, flat_motor,
             flat_position=0*q.mm,
             radio_position=10*q.mm)
scan = TomographyScan(setup, rot_motor,
                    intervals=500)
```

Then we feed both objects and a set of basic parameters to a Radiography object (subclassing Experiment) that connects the acquisition steps and ensures that dark and flat fields as well as the projection data are recorded properly.

```
ex = Radiography(setup, num_darks=10, num_flats=10,
                radio_producer=scan)
```

Last, we create a CTReconstruction addon to reconstruct slices from the incoming projections:

```
reco = CTReconstruction(ex, scan.intervals,
                        axis_position, slice_consumer=viewer)
ex.run()
```

Instead of using a single Tomography object to describe a tomographic experiment, we separate the data acquisition using the TomographyScan object from the sequencing of events represented by the Radiography object. This separation eases composition of experiments with different scan characteristics, e.g. step or continuous scans. Otherwise the user would have to create specific experiment classes to handle special scan modes.

5. Case study workflows

In this section we present two use cases that demonstrate the need for an experiment control framework that is capable of processing acquired data on-the-fly. In the first case we show how to improve time-resolved tomography by determining acquisition parameters during a scan. In the second case we show an improvement of data quality by continuous quality assessment of a laminography scan.

5.1. Online control of acquisition parameters for time-resolved measurements

To demonstrate automatic process parameter determination we conducted a CT experiment that investigates the change of structural properties of a liquid foam over time. Bubbles in this foam rupture and merge to form larger bubbles at unknown rates. With such a highly dynamic sample, a data acquisition that is too slow causes noticeable reconstruction artifacts due to inadequate sampling. As seen in Fig. 5(a), such

artifacts impede any quantitative measurements in subsequent post-reconstruction analyses. However, unnecessarily lowering exposure times to shorten acquisition decreases the signal-to-noise ratio and therefore degrades image quality which again complicates subsequent analysis. Moreover, superfluous frame rates are at the expense of the intermediate buffer capacity.

Hence, our workflow tried to find an optimum between foam stability and imaging quality in an automatic way. To determine an optimum quantitatively, we reconstructed slices online on-the-fly and evaluated the similarity between consecutive measurements. The experiment workflow was driven by a feedback loop following the concept shown in Fig. 1 closely. It consists of (i) continuous acquisition of two sets of tomographic projections that are initially acquired with a low exposure time (providing reasonable dynamic range) causing a slow acquisition speed, (ii) immediate reconstruction on GPUs, (iii) comparison of the reconstructed slices and (iv) a final step that may or may not adapt the acquisition speed. The comparison is based on the correlation coefficient

$$r = \frac{\sum_i(x_i - \bar{x})(y_i - \bar{y})}{[\sum_i(x_i - \bar{x})^2]^{1/2} [\sum_i(y_i - \bar{y})^2]^{1/2}}, \quad (1)$$

which was used to estimate the similarity between two slices. If r did not exceed a specified threshold, we doubled the acquisition speed and repeated the process, otherwise we stopped the process. Doubling the acquisition speed decreases the relative sample motion but also increases the noise level due to the lower exposure time. Our objective was therefore to find the slowest DAQ speed at which no noticeable 3D reconstruction artifacts occur.

Although the correlation of a projection at 0° and the flipped projection at 180° approximates general foam changes very well, a projection consists of absorption values in the beam propagation direction thus any information perpendicular to that direction is lost during the absorption. However, in this illustrative experiment we were interested in the forming and destruction of bubbles which might be obstructed by non-essential features in any direction as seen on the left in Fig. 8. Hence, we have to reconstruct slices from the data and

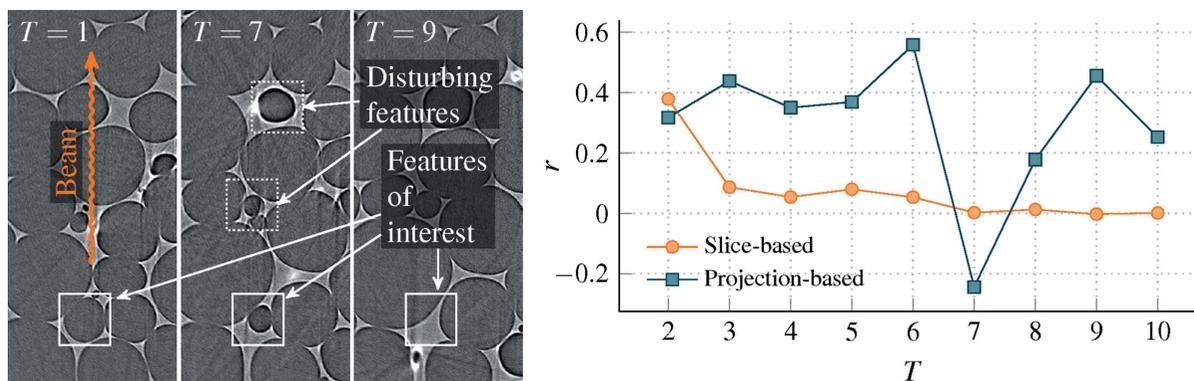


Figure 8 Tomographic slices T depicting the same region at different stages of the foaming process and the corresponding slice- and projection-based correlation coefficient r between the first and the n th tomogram.

correlate consecutive tomograms to obtain information about bubbles in pre-defined ROIs. As seen in the quantitative comparison, the slice-based correlation decays smoothly as the bubble in the ROI, depicted by the solid rectangle, disappears. The projection-based correlation, applied on the same projection row, disagrees with this situation because the disturbing events depicted by the dotted rectangles at $T = 7$ are superimposed on the ROI in the direction of the projection.

To test whether our hypothesis held and we could successfully determine the optimal acquisition speed in an automatic fashion, we set the correlation threshold that stopped the speed adjustment to an empirically determined value of 0.7. The process settled after three iterations acquiring data at 200, 400 and 800 frames s^{-1} with corresponding correlation coefficients between slice crops of 0.295, 0.397 and 0.744, *i.e.* we were able to converge towards stable, similar reconstructions. Thus, our slice-based feedback approach allowed us to capture bubble formation and rupture events in a specific ROI, which was not feasible with projection-based correlation due to the lack of the third dimension.

5.2. Interactive control of ROI positioning and data quality for *in situ* measurements

The aim of this case study is to follow *in situ* damage evolution in engineering materials as a function of an applied load force. SRCL allows us to measure the damage evolution (Kahziz *et al.*, 2015) in a relatively small region of interest in the material bulk while maintaining engineering-relevant mechanical boundary conditions. Using different large and flat specimens, damage mechanisms for different associated levels of stress triaxiality (ranging from shear to highly triaxial stress states) can be imaged in 3D and *in situ* at high resolution.

In the given example the material microstructure is resolved by a detector based on an optical microscope (Douissard *et al.*, 2012) which magnifies the image of a 8.7 LSO:Tb scintillator onto a scientific CMOS camera (pco.edge), yielding an effective pixel size of 0.65.

The main goal of the workflow was to acquire, store and reconstruct the data in an online fashion to assess the quality of the acquisition during the scan in order to avoid recording problematic or unusable data as outlined in §2. To achieve this, the control system broadcasts the incoming data stream simultaneously to the data storage and our compute framework for saving raw data and reconstructing slices for feedback. The reconstructed central slice is investigated right after the scan to judge whether the acquired data are not blurred and the scan successful.

Fig. 9 shows a properly aligned AA21xx alloy for a local shear loading case, without loading (*a*) and under shear load (*b*). Despite the presence of inherent laminographic artifacts (Xu *et al.*, 2012*b*), one can observe the elongation of the materials' pores into shear cracks under load. These cracks are aligned roughly along the principal shear force direction marked with arrows. The change in notch shape due to large plastic deformation can also be discerned.

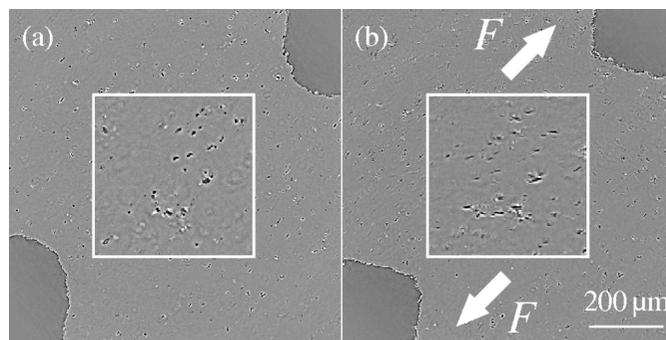


Figure 9
2D section of an *in situ* laminographic scan showing two well aligned notches at mid-thickness in a 1 mm-thick Al-alloy sample and a magnification of the smaller structures (*a*) in the undeformed state, and (*b*) *in situ* after local shear loading induced by force F .

6. Discussion

The two reconstruction-based workflows presented in §5.1 and §5.2 demonstrate the applicability of our high-throughput and low-latency experiment control system introduced in §3. With this system we are able to control the correct tomographic acquisition speed in a timely fashion as well as the ROI position and the image quality in an *in situ* laminography experiment. As a matter of fact, the fast GPU reconstruction not only helps to assess the acquisition quality in near real-time but is also used to automatically optimize reconstruction parameters before processing the data offline. The routine optimizes parameters such as the rotation center in the projections and the optimal inclination angles between rotation axis and beam and towards the detector coordinates.

Although we are able to analyze data online with low latencies and provide feedback to the experiment, the online GPU data processing is still currently two times slower than the offline version because of additional data transfers between the control system and the data processing stage. To solve this problem we will investigate ways to hide the memory transfer time better, *e.g.* using pinned memory techniques and GPUs with larger memory. Moreover, the laminographic backprojection algorithm is subject to further performance optimization in order to reduce memory transfer overheads.

As an outlook and with even faster data processing in mind, we will integrate specialized and robust metrics for event detection (so-called change detection), such as optical flow, image registration and pattern matching analysis. We will also work on a decision-making scheme which evaluates such metrics and provides feedback to the experiment without user interaction. These parts are necessary for further automation of high-speed dynamic experiments and to increase their robustness. A more robust approach will also help to reduce the present jitter which enables soft real-time operation.

7. Conclusion

In this paper, we have presented a flexible system composed of a high-performance computational backend, a fast, generic

data acquisition library and a high-level experiment control system to build flexible feedback workflows. We have evaluated the entire system at two independent experiments carried out at ANKA's IMAGE beamline and the ID19 beamline of the ESRF. The experiments have shown that tight integration of data acquisition, data processing and experiment control allows for fast optimization of experimental parameters and *in situ* quality assessments based on fast feedback from image analysis. In the particular examples, the system helped us to assess the interactions between notches and internal damage, *in situ* and in a non-destructive way, by using online laminography. Most importantly, we can instantly validate experimental parameters, the image acquisition quality as well as sample environments such as loading apparatus. This allows the user to concentrate on the scientific questions such as optimization of the load steps, fast and even automated spatiotemporal ROI identification which in turn improves efficiency of allocated beam time. In essence, our online analysis framework will greatly simplify the transition from traditional *in situ* experiments to interactive, real-time image-processing driven smart 4D imaging.

Acknowledgements

The presented work was funded by the German Federal Ministry of Education and Research (BMBF) as UFO-1 and UFO-2 under the grants 05K10CKB and 05K10VKE. We would like to thank the ESRF for beam time allocation used to conduct the laminographic experiment MA2183 for our case study. We thank the ANKA synchrotron source for provision of beam time for the tomography experiment. The support of the Agence Nationale de la Recherche (ANR-14-CE07-0034-02 grant for COMINSIDE project) is gratefully acknowledged.

References

- Altapova, V., Helfen, L., Myagotin, A., Hänschke, D., Moosmann, J., Gunneweg, J. & Baumbach, T. (2012). *Opt. Express*, **20**, 6496–6508.
- Boden, S., dos Santos Rolo, T., Baumbach, T. & Hampel, U. (2014). *Exp. Fluids*, **55**, 1768.
- Bull, D. J., Helfen, L., Sinclair, I., Spearing, S. M. & Baumbach, T. (2013). *Compos. Sci. Technol.* **75**, 55–61.
- Cheng, Y., Laiarinandrasana, L., Helfen, L., Proudhon, H., Klinkova, O., Baumbach, T. & Morgeneyer, T. F. (2016). *Macromolecular Chemistry and Physics*. New York: Wiley.
- Chilingaryan, S., Mirone, A., Hammersley, A., Ferrero, C., Helfen, L., Kopmann, A., dos Santos Rolo, T. & Vagović, P. (2011). *IEEE Trans. Nucl. Sci.* **58**, 1447–1455.
- Cloetens, P., Ludwig, W., Baruchel, J., Van Dyck, D., Van Landuyt, J., Guigay, J. & Schlenker, M. (1999). *Appl. Phys. Lett.* **75**, 2912.
- Douissard, P. A., Cecilia, A., Rochet, X., Chapel, X., Martin, T., van de Kamp, T., Helfen, L., Baumbach, T., Luquot, L., Xiao, X., Meinhardt, J. & Rack, A. (2012). *J. Instrum.* **7**, P09016.
- Dritschler, T., Chilingaryan, S., Farago, T., Kopmann, A. & Vogelgesang, M. (2014). *Proceedings of the 10th International Workshop on Personal Computers and Particle Accelerators (PCAPAC '14)*. In the press.
- Finegan, D. P., Scheel, M., Robinson, J. B., Tjaden, B., Hunt, I., Mason, T. J., Millichamp, J., Di Michiel, M., Offer, G. J., Hinds, G., Brett, D. J. L. & Shearing, P. R. (2015). *Nat. Commun.* **6**, 6924.
- Folk, M., Cheng, A. & Yates, K. (1999). *Proceedings of Supercomputing*, Vol. 99.
- Greven, H., van de Kamp, T., dos Santos Rolo, T., Baumbach, T. & Clemen, G. (2015). *Vertebr. Zool.* **65**, 81–99.
- Gürsoy, D., De Carlo, F., Xiao, X. & Jacobsen, C. (2014). *J. Synchrotron Rad.* **21**, 1188–1193.
- Hänschke, D., Helfen, L., Altapova, V., Danilewsky, A. & Baumbach, T. (2012). *Appl. Phys. Lett.* **101**, 244103.
- Harasse, S., Yashiro, W. & Momose, A. (2011). *Opt. Express*, **19**, 16560–16573.
- Helfen, L., Myagotin, A., Mikulík, P., Pernot, P., Voropaev, A., Elyyan, M., Di Michiel, M., Baruchel, J. & Baumbach, T. (2011). *Rev. Sci. Instrum.* **82**, 063702.
- Houssaye, A., Xu, F., Helfen, L., De Buffrénil, V., Baumbach, T. & Tafforeau, P. (2011). *J. Vertebr. Paleontol.* **31**, 2–7.
- Jonge, M. D. de & Vogt, S. (2010). *Curr. Opin. Struct. Biol.* **20**, 606–614.
- Jung, J., Lee, J., Kwon, N., Park, S., Chang, S., Kim, J., Pyo, J., Kohmura, Y., Nishino, Y., Yamamoto, M., Ishikawa, T. & Je, J. H. (2012). *Rev. Sci. Instrum.* **83**, 093704.
- Kahziz, M., Morgeneyer, T., Maziere, M., Helfen, L., Bouaziz, O. & Maire, E. (2015). *Exp. Mech.* **56**, 177–195.
- Kamp, T. van de, Cecilia, A., dos Santos Rolo, T., Vagović, P., Baumbach, T. & Riedel, A. (2015). *Arthropod Structure Devel.* **44**, 509–523.
- Ludwig, W., Schmidt, S., Lauridsen, E. M. & Poulsen, H. F. (2008). *J. Appl. Cryst.* **41**, 302–309.
- Mader, K., Marone, F., Hintermüller, C., Mikuljan, G., Isenegger, A. & Stampanoni, M. (2011). *J. Synchrotron Rad.* **18**, 117–124.
- Maurel, V., Helfen, L., Soullignac, R., Morgeneyer, T. F., Koster, A. & Rémy, L. (2013). *Oxid. Met.* **79**, 313–323.
- Morgeneyer, T. F., Helfen, L., Mubarak, H. & Hild, F. (2013). *Exp. Mech.* **53**, 543–556.
- Morgeneyer, T. F., Taillandier-Thomas, T., Helfen, L., Baumbach, T., Sinclair, I., Roux, S. & Hild, F. (2014). *Acta Mater.* **69**, 78–91.
- Moura, A. L. D. & Ierusalimsky, R. (2009). *ACM Trans. Program. Lang. Syst.* **31**, 1–31.
- Myagotin, A., Voropaev, A., Helfen, L., Hänschke, D. & Baumbach, T. (2013). *IEEE Trans. Image Processing*, **22**, 5348–5361.
- Reischig, P., Helfen, L., Wallert, A., Baumbach, T. & Dik, J. (2013). *Appl. Phys. A*, **111**, 983–995.
- Riedel, A., Dos Santos Rolo, T., Cecilia, A. & Van De Kamp, T. (2012). *Zool. J. Linn. Soc.* **165**, 773–794.
- Salvo, L., Lhuissier, P., Scheel, M., Terzi, S., Di Michiel, M., Boller, E., Taylor, J., Dahle, A. & Suéry, M. (2012). *Trans. Indian Inst. Met.* **65**, 623–626.
- Santos Rolo, T. dos, Ershov, A., van de Kamp, T. & Baumbach, T. (2014). *Proc. Natl Acad. Sci.* **111**, 3921–3926.
- Shen, Y., Morgeneyer, T. F., Garnier, J., Allais, L., Helfen, L. & Crépin, J. (2013). *Acta Mater.* **61**, 2571–2582.
- Stock, S. (2008). *Int. Mater. Rev.* **53**, 129–181.
- Thompson, A., Llacer, J., Campbell Finman, L., Hughes, E., Otis, J., Wilson, S. & Zeman, H. (1984). *Nucl. Instrum. Methods Phys. Res.* **222**, 319–323.
- Tian, T., Xu, F., Kyu Han, J., Choi, D., Cheng, Y., Helfen, L., Di Michiel, M., Baumbach, T. & Tu, K. N. (2011). *Appl. Phys. Lett.* **99**, 082114.
- Vogelgesang, M., Chilingaryan, S., dos Santos Rolo, T. & Kopmann, A. (2012). In *Proceedings of the 14th IEEE Conference on High Performance Computing and Communication and the 9th IEEE International Conference on Embedded Software and Systems (HPCC-ICESS) (HPCC '12)*, pp. 824–829. IEEE Computer Society.
- Vogelgesang, M., Farago, T., dos Santos Rolo, T., Kopmann, A. & Baumbach, T. (2013). In *Proceedings of the 14th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALPECS '13)*, 6–11 October 2013, San Francisco, CA, USA.

- Walker, S. M., Schwyn, D. A., Mokso, R., Wicklein, M., Müller, T., Doube, M., Stampanoni, M., Krapp, H. G. & Taylor, G. K. (2014). *PLoS Biol.* **12**, e1001823.
- Weitkamp, T., Diaz, A., David, C., Pfeiffer, F., Stampanoni, M., Cloetens, P. & Ziegler, E. (2005). *Opt. Express*, **13**, 6296–6304.
- Westneat, M. W., Socha, J. J. & Lee, W.-K. (2008). *Annu. Rev. Physiol.* **70**, 119–142.
- Xu, F., Helfen, L., Baumbach, T. & Suhonen, H. (2012b). *Opt. Express*, **20**, 794–806.
- Xu, F., Helfen, L., Moffat, A. J., Johnson, G., Sinclair, I. & Baumbach, T. (2010). *J. Synchrotron Rad.* **17**, 222–226.
- Xu, F., Helfen, L., Suhonen, H., Elgrabli, D., Bayat, S., Reischig, P., Baumbach, T. & Cloetens, P. (2012a). *PLoS One*, **7**, e50124.
- Yazzie, K., Williams, J., Phillips, N., De Carlo, F. & Chawla, N. (2012). *Mater. Charact.* **70**, 33–41.